

## PROGRAMIRANJE 1



**Milena Vujošević Janičić, Jovana Kovačević,  
Danijela Simić, Anđelka Zečević,  
Aleksandra Kocić**

# **PROGRAMIRANJE 1**

## **Zbirka zadataka**

**Beograd  
2017.**

Autori:

*dr Milena Vujošević Janičić*, 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

*Aleksandra Kocić*, 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 . . . . .	41
2.3	Petlje . . . . .	71
2.4	Rešenja . . . . .	101
2.5	Funkcije . . . . .	161
2.6	Rešenja . . . . .	174
<b>3</b>	<b>Predstavljanje podataka</b>	<b>209</b>
3.1	Nizovi . . . . .	209
3.2	Rešenja . . . . .	230
3.3	Pokazivači . . . . .	282
3.4	Rešenja . . . . .	286
3.5	Niske . . . . .	294
3.6	Rešenja . . . . .	306



# 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 decimalne. 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 decimalne.

## 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 decimalne. 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

---

```
1 int main()
2 {
3     /* Deklaracija promenljivih cija je vrednost neoznacena ceo broj. */
4     unsigned int cena, kolicina, iznos;
5     unsigned int kusur;
6
7     /* Ucitavaju se vrednosti cene, kolicine i iznosa. */
8     printf("Unesite cenu, kolicinu i iznos:\n");
9     scanf("%u%u%u", &cena, &kolicina, &iznos);
10
11     /* Izracunava se kusur. */
12     kusur = iznos - kolicina * cena;
13
14     /* Ispis vrednosti kusura. */
15     printf("Kusur je %u dinara.\n", kusur);
16
17     return 0;
18 }
19
20 }
```

### Rešenje 1.1.7

```
1 #include <stdio.h>
2
3 int main()
4 {
5
6     /* Deklaracija potrebnih promenljivih. */
7     unsigned int poletanje, poletanje_sat, poletanje_minut;
8     unsigned int sletanje, sletanje_sat, sletanje_minut;
9     unsigned int duzina, duzina_sat, duzina_minut;
10
11     /* Ucitavaju se sat i minut vremena poletanja. */
12     printf("Unesite vreme poletanja: ");
13     scanf("%u%u", &poletanje_sat, &poletanje_minut);
14
15     /* Ucitavaju se sat i minut vremena sletanja. */
16     printf("Unesite vreme sletanja: ");
17     scanf("%u%u", &sletanje_sat, &sletanje_minut);
18
19     /* Obe vrednosti se pretvaraju u sekunde,
20        kako bi se lakse izracunala razlika. */
21     poletanje = poletanje_sat * 3600 + poletanje_minut * 60;
22     sletanje = sletanje_sat * 3600 + sletanje_minut * 60;
23
24     /* Racunanje razlike u sekundama izmedju sletanja i poletanja. */
25     duzina = sletanje - poletanje;
26
27     /* Razlika u sekundama se pretvara u razliku u satima i minutima.
28        Razlika u satima se dobija celobrojn timer deljenjem broja sekundi
29        sa 3600.
30        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 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
9      /* Ucitavaju se brojevi n i m. */
10     printf("Unesite pozitivan ceo broj: ");
11     scanf("%u", &n);
12     printf("Unesite pozitivan dvocifreni broj: ");
13     scanf("%u", &m);
14
15     /* Levi deo rezultata su sve cifre levo od cifre stotina.
16     Na primer, ako je n=12345, levi deo rezultata je 12.
17     On se dobija deljenjem unetog broja sa 1000. */
18     levi = n / 1000;
19
20     /* Desni deo rezultata su sve cifre desno od cifre hiljada.
21     Za n=12345, desni deo rezultata je 345. */
22     desni = n % 1000;
23
24     /* Srednji deo rezultata je broj m.
25     U navedenom primeru, rezultat se dobija nadovezivanjem
26     brojeva 12, 67 i 345. Ovo se radi mnozenjem delova sa
27     odgovarajucim stepenom broja 10 i njihovim sabiranjem. */
28     novibroj = levi * 100000 + m * 1000 + desni;
29
30     /* Ispis rezultata. */
```

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

### Rešenje 1.1.16

```
1 #include <stdio.h>
3 int main()
4 {
5     /* Deklaracija potrebnih promenljivih. */
6     float in, cm;
7
8     /* Ucitava se realna vrednost koja predstavlja broj inca. */
9     printf("Unesite broj inca: ");
10    scanf("%f", &in);
11
12    /* Izracunava se rezultat (1 in = 2.54 cm) */
13    cm = in * 2.54;
14
15    /* Ispis rezultata (na dve decimale). */
16    printf("%.2f in = %.2f cm\n", in, cm);
17
18    return 0;
19 }
```

### 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

```
1 #include <stdio.h>
3 int main()
4 {
5     /* Deklaracija potrebnih promenljivih. */
6     float a11, a12, a21, a22;
7     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 decimalne. */
   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, površina;

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 površina pravougaonika. */
   površina = a * b;

18

20  /* Ispis rezultata na dve decimalne. */
   printf("Obim: %.2f\n", obim);
   printf("Površina: %.2f\n", površina);

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, površina;
```

```

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 pribilznoj 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 duzina stranice. */
10 printf("Unesite duzinu 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, povrsina;

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

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

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

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

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 /* 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,
15 deljenje je celobrojno.
Na primer, (1+1+3)/3 ima vrednost 1.*/

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 je 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 pozitivni 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 tri karaktera i ispisuje proizvod svih karaktera koji su cifre. Ukoliko među unetim karakterima nema cifara, program treba da ispiše odgovarajuću poruku. NAPOMENA: *Karakteri koji se unose su razmaknuti blanko znacima.*

### 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 2.1.9]

**Zadatak 2.1.10** Kasirka unosi šifru artikla koja se zadaje kao tri spojena karaktera koji mogu biti mala slova, velika slova ili cifre. U kasi, sve šifre su zapisane malim slovima i ciframa. Napisati program koji kasirkin unos 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 2.1.10]

**Zadatak 2.1.11** 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 četvorocifreni broj: 6835  
| Najveća cifra je: 8
```

### Primer 2

```
|| INTERAKCIJA SA PROGRAMOM:  
| Unesite četvorocifreni broj: 7777  
| Najveća cifra je: 7
```

### Primer 3

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

### Primer 4

```
|| INTERAKCIJA SA PROGRAMOM:  
| Unesite četvorocifreni broj: -2002  
| Najveća cifra je: 2
```

[Rešenje 2.1.11]

**Zadatak 2.1.12** Broj je Armstrongov ako je jednak zbiru kubova svojih cifara. Napisati program koji za dati pozitivan trocifreni broj proverava da li je Armstrongov. U slučaju neispravnog unosa, ispisati odgovarajuću poruku o grešci.

### Primer 1

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

### Primer 2

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

### Primer 3

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

### Primer 4

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

[Rešenje 2.1.12]

**Zadatak 2.1.13** 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 četvorocifreni broj: 8123  
|| Proizvod parnih cifara: 16
```

### Primer 2

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

### Primer 3

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

### Primer 4

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

[Rešenje 2.1.13]

**Zadatak 2.1.14** Napisati program koji učitava četvorocifreni broj i ispisuje broj koji se dobija kada se unetom broju razmene najmanja i najveća cifra. U slučaju da se najmanja ili najveća cifra pojavljuju na više pozicija, uzeti prvo pojavljivanje, gledajući sa desna na levo. U slučaju neispravnog unosa, ispisati odgovarajuću poruku o grešci.

### 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 cetvorocifreni broj: 247
|| Greska: niste uneli cetvorocifreni broj.
```

### Primer 4

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

[Rešenje 2.1.14]

**Zadatak 2.1.15** 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 cetvorocifreni broj: 1389
|| Cifre su uredjene neopadajuće.
```

### Primer 2

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

### Primer 3

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

### Primer 4

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

[Rešenje 2.1.15]

**Zadatak 2.1.16** Napisati program koji ispituje da li se tačke  $A(x_1, y_1)$  i  $B(x_2, y_2)$  nalaze u istom kvadrantu. Koordinate tačaka su realni brojevi jednostruke tačnosti.

### Primer 1

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite koordinate tacke A: 1.5 6
|| Unesite koordinate tacke B: 2.33 9.8
|| Tacke se nalaze u istom kvadrantu.
```

### Primer 2

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite koordinate tacke A: -3 6
|| Unesite koordinate tacke B: 0.33 -5
|| Tacke se ne nalaze u istom kvadrantu.
```

### Primer 3

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite koordinate tacke A: 0 -6
|| Unesite koordinate tacke B: -1 -99.66
|| Tacke se nalaze u istom kvadrantu.
```

### Primer 4

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite koordinate tacke A: 3 -6
|| Unesite koordinate tacke B: -0.33 0
|| Tacke se ne nalaze u istom kvadrantu.
```

[Rešenje 2.1.16]

**Zadatak 2.1.17** 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.

*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
Tacke se nalaze na istoj pravoj.
```

*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
Tacke se ne nalaze na istoj pravoj.
```

*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
Tacke se nalaze na istoj pravoj.
```

*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
Tacke se nalaze na istoj pravoj.
```

*Primer 5*

```
INTERAKCIJA SA PROGRAMOM:
Unesite koordinate tacke A: 1 2
Unesite koordinate tacke B: 1 2
Unesite koordinate tacke C: -56 1.3
Tacke se nalaze na istoj pravoj.
```

*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
Tacke se nalaze na istoj pravoj.
```

[Rešenje 2.1.17]

**Zadatak 2.1.18** Napisati program za rad sa intervalima. Za dva celobrojna intervala  $[a_1, b_1]$  i  $[a_2, b_2]$ , program treba da odredi:

- dužinu preseka datih intervala
- presečni interval datih intervala
- dužinu prave koju pokrivaju dati intervali
- 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 2.1.18]

**Zadatak 2.1.19** 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 2.1.19]

**Zadatak 2.1.20** 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 2.1.20]

**Zadatak 2.1.21** Data je funkcija  $f(x) = 2 \cdot \cos(x) - x^3$ . Napisati program koji za učitane vrednost realne promenljive  $x$  i vrednost celobrojne promenljive  $k$  koje može biti 1, 2 ili 3 izračunava vrednost funkcije  $F(x, k)$  koja se dobija tako što se funkcija  $f$  primeni  $k$ -puta ( $F(x, 1) = f(x)$ ,  $F(x, 2) = f(f(x))$ ,  $F(x, 3) = f(f(f(x)))$ ) 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 [2.1.21](#)]

**Zadatak 2.1.22** 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 [2.1.22](#)]

**Zadatak 2.1.23** 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 [2.1.23](#)]

**Zadatak 2.1.24** 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 2.1.24]

**Zadatak 2.1.25** Napisati program koji za uneti datum u formatu *dan.mesec*. ispisuje godišnje doba kojem pripadaju. NAPOMENA: *Pretpostaviti da je unos ispravan.*

### 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 2.1.25]

**Zadatak 2.1.26** 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 2.1.26]

**Zadatak 2.1.27** Napisati program koji za uneti datum u formatu *dan.me-sec.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 2.1.27]

**Zadatak 2.1.28** 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 2.1.28]

**Zadatak 2.1.29** 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 2.1.29]

\* **Zadatak 2.1.30** Polje šahovske table se definiše parom celih brojeva  $(x, y)$ ,  $1 \leq x, y \leq 8$ , gde je  $x$  redni broj reda, a  $y$  redni broj kolone. Napisati program koji za unete parove  $(k, l)$  i  $(m, n)$  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 šahovske 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 2.1.30]

## 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 }
```



```

21  /* II nacin: koriscenjem funkcije fabs cija se deklaracija nalazi
    u zaglavlju math.h: apsolutno_x=fabs(x); */
23  return 0;
}

```

### Rešenje 2.1.3

```

1  #include <stdio.h>
3  int main()
4  {
5      /* Deklaracija potrebnih promenljivih. */
6      int x;
7      float recipročno_x;
9      /* Ucitavanje vrednosti broja x. */
10     printf("Unesite jedan ceo broj:");
11     scanf("%d", &x);
13     /* Vrsi se provera ispravnosti ulaznih podataka. Napomena: za
        razliku od izlaza iz programa sa kodom 0 (return 0;) koji
14     služi kao indikator da se program završio uspešno, izlaz iz
15     programa sa izlaznim kodom koji se razlikuje od nule služi
16     kao indikator da je pri izvršavanju programa došlo do neke
17     greske. */
18     if (x == 0) {
19         printf("Greska: nedozvoljeno je deljenje nulom.\n");
20         return -1;
21     }
22
23     /* Racuna se recipročna vrednost. */
24     recipročno_x = 1.0 / x;
25
26     /* Ispis rezultata. */
27     printf("Recipročna vrednost: %.4f\n", recipročno_x);
28
29     return 0;
30 }

```

### Rešenje 2.1.4

```

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

```

## 2 Kontrola toka

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

11

13  /* Pocetna vrednost sume se postavlja na 0. */
   suma = 0;

15  /* Na sumu se dodaju vrednosti onih brojeva cija je vrednost
   pozitivna. Uvecavanje je moguće uraditi na dva nacina:
17     I nacin: suma = suma + vrednost;
     II nacin: suma += vrednost; */
19  if (a > 0)
       suma = suma + a;

21
23  if (b > 0)
       suma += b;

25  if (c > 0)
       suma += c;

27
29  /* Ispis rezultata. */
   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;
     int najjeftiniji;
     int cena_bez_popusta, cena_sa_popustom;

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

13

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

19

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

23     if (b < najjeftiniji)
```

```

    najjeftiniji = b;
25
    if (c < najjeftiniji)
27        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. */
    if (sati > 24 || sati < 0 || minuti > 59 || minuti < 0) {
17         printf("Greska: neispravan unos vremena.\n");
        return -1;
19     }

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

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

```

## 2 Kontrola toka

---

```
35  /* Ispis rezultata. */
    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()
{
5     /* Deklaracija potrebne promenljive. */
    unsigned int x;

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

11

13     /* Proverava se da li je godina prestupna ili ne i ispisuje se
        odgovarajuca poruka. Godina je prestupna ukoliko vazi jedan od
        narednih uslova:
        1. da je deljiva sa 4, a nije sa 100
        2. da je deljiva sa 400. */
15     if ((x % 4 == 0 && x % 100 != 0) || x % 400 == 0)
17         printf("Godina je prestupna.\n");
    else
19         printf("Godina nije prestupna.\n");

21
23     return 0;
}
```

### Rešenje 2.1.8

```
1  #include <stdio.h>

2
3  int main()
4  {
5     /* Deklaracija karakterske promenljive. */
6     char c;

7     /* Ucitava se jedan karakter. */
8     printf("Unesite karakter: ");
10    scanf("%c", &c);

12    /* Ispis karaktera i vrednosti njegovog ASCII koda. */
    printf("Uneti karakter: %c\n", c);
}
```

```

14 printf("ASCII kod: %d\n", c);

16 /* Karakteri koji odgovaraju velikim slovima su u ASCII tablici
   smesteni sekvencijalno. Na primer, ASCII kod karaktera 'A' je
18 65, 'B' je 66, ..., 'Z' je 90. Isto vazi i za mala slova: 'a'
   je 97, 'b' je 98, ..., 'z' je 122.

20
   Oдавде, ako se vrsi provera da li je neki karakter veliko
22 slovo, dovoljno je proveriti da li se njegov ASCII kod nalazi
   izmedju ASCII kodova slova 'A' i slova 'Z'.

24
   Dodatno, moze se primetiti da je razlika izmedju ASCII koda
26 svakog malog i odgovarajuceg velikog slova konstanta koja ima
   vrednost 'a'-'A', sto je isto sto i 'b'-'B', itd. Zbog toga,
28 ako je potrebno od velikog slova dobiti malo, onda je
   dovoljno ASCII kodu velikog slova dodati pomenutu konstantu.
30 Za mala slova, vazi obrnuto - da bi se dobilo veliko slovo,
   ova konstanta se oduzima. */

32
   if (c >= 'A' && c <= 'Z') {
34     printf("Odgovarajuce malo slovo: %c\n", c + ('a' - 'A'));
     printf("ASCII kod: %d\n", c + ('a' - 'A'));
36   }

   if (c >= 'a' && c <= 'z') {
38     printf("Odgovarajuce veliko slovo: %c\n", c - ('a' - 'A'));
     printf("ASCII kod: %d\n", c - ('a' - 'A'));
40   }

42   return 0;
44 }

```

### Rešenje 2.1.9

```

1  #include <stdio.h>

3  int main()
   {
5     /* Deklaracija potrebnih promenljivih. */
     unsigned int broj_cifara = 0;
7     unsigned int proizvod_cifara = 1;

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

```

```
19  c1 = getchar();
    getchar();
    c2 = getchar();
21  getchar();
    c3 = getchar();
23
    /* II nacin ucitavanja ulaza: koriscenjem funkcije scanf()
25     Blanko znaci se navode kao deo ocekivanog formata ulaza.
        char c1, c2, c3;
27     scanf("%c %c %c", &c1, &c2, &c3); */

29  /* Pogresan nacin ucitavanja ulaza:
        scanf("%c%c%c", &c1, &c2, &c3);
31     U ovom slucaju ce u c1 biti upisan prvi karakter, u c2
        blanko i u c3 drugi karakter. */
33
35  /* Karakteri koji predstavljaju cifre su u ASCII tablici takodje
        smesteni sekvencijalno. Na primer, '0' ima ASCII kod 48, '1'
        49, ..., '9' ima ASCII kod 57.

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

41
43     Dodatno, ako je potrebno izracunati dekadnu vrednost karaktera
        koji je cifra, dovoljno je od ASCII koda tog karaktera,
        oduzeti ASCII kod karaktera '0'. Na primer, '4'-'0' = 52 - 48
        = 4. */
45
47  /* Racuna se proizvod onih karaktera koji su cifre. */
    if (c1 >= '0' && c1 <= '9') {
        proizvod_cifara *= (c1 - '0');
49     broj_cifara++;
    }

51
53     if (c2 >= '0' && c2 <= '9') {
        proizvod_cifara *= (c2 - '0');
        broj_cifara++;
55     }

57     if (c3 >= '0' && c3 <= '9') {
        proizvod_cifara *= (c3 - '0');
59     broj_cifara++;
    }

61
63  /* Ispis rezultata. */
    if (broj_cifara == 0)
        printf("Medju unetim karakterima nema cifara.\n");
65     else
        printf("Proizvod cifara: %u\n", proizvod_cifara);
67
69     return 0;
}
```

## Rešenje 2.1.10

```
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 2.1.11

```
1  #include <stdio.h>
   #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 vrednost broja n. */
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     /* Ispis rezultata */
43     printf("Najveca cifra je: %d\n", najveca_cifra);
44
45     return 0;
46 }
```



## 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;
9
10    /* Ucitava se vrednost broja n. */
11    printf("Unesite pozitivan trocifreni broj: ");
12    scanf("%d", &n);
13
14    /* Vrsi se provera ispravnosti ulaznih podataka. */
15    if (n < 100 || n > 999) {
16        printf("Greska: niste uneli pozitivan trocifreni broj.\n");
17        return -1;
18    }
19
20    /* Izdvajaju se cifre broja n. */
21    jedinica = n % 10;
22    desetica = (n / 10) % 10;
23    stotina = n / 100;
24
25    /* Ispis rezultata. */
26    if (n == jedinica * jedinica * jedinica +
27        desetica * desetica * desetica + stotina * stotina * stotina)
28        printf("Broj je Armstrongov.\n");
29    else
30        printf("Broj nije Armstrongov.\n");
31
32    return 0;
33 }
```

## 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, hiljada;
9     int broj_parnih, proizvod_parnih;
10
11    /* Ucitava se vrednost broja n. */
12    printf("Unesite cetvorocifreni broj: ");
```

```
13  scanf("%d", &n);

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

19  /* Vrsi se provera ispravnosti ulaznih podataka. */
21  if (n < 1000 || n > 9999) {
23      printf("Greska: niste uneli cetvorocifreni broj.\n");
25      return -1;
27  }

29  /* Izdvajaju se cifre broja n. */
31  jedinica = n % 10;
33  desetica = (n / 10) % 10;
35  stotina = (n / 100) % 10;
37  hiljada = n / 1000;

39  /* Inicijalizacija brojaca i rezultata. */
41  broj_parnih = 0;
43  proizvod_parnih = 1;

45  /* Za svaku cifru se vrsi provera da li je parna i ukoliko jeste
    tekuci rezultat se mnozi sa tekucom cifrom. */
47  if (jedinica % 2 == 0) {
49      proizvod_parnih = proizvod_parnih * jedinica;
51      broj_parnih++;
53  }

55  if (desetica % 2 == 0) {
57      proizvod_parnih = proizvod_parnih * desetica;
59      broj_parnih++;
61  }

63  if (stotina % 2 == 0) {
65      proizvod_parnih = proizvod_parnih * stotina;
67      broj_parnih++;
69  }

71  if (hiljada % 2 == 0) {
73      proizvod_parnih = proizvod_parnih * hiljada;
75      broj_parnih++;
77  }

79  /* Ispis rezultata. */
81  if (broj_parnih == 0) {
83      printf("Nema parnih cifara.\n");
85  } else {
87      printf("Proizvod parnih cifara: %d\n", proizvod_parnih);
89  }

91  return 0;
```

65 }

## Rešenje 2.1.14

```

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 vrednost broja 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
   /* Po algoritmu za trazenje najvece/najmanje cifre (koji je
      prikazan u zadatku 2.1.11) racunaju se najveca i najmanja
      cifra broja n, 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, cifre desetica
      10, itd... */
33
   najveca = jedinica;
35  stepen_najvece = 1;
37
   if (desetica > najveca) {
       najveca = desetica;
       stepen_najvece = 10;
41   }
43
   if (stotina > najveca) {
       najveca = stotina;
45   }
47

```

```

    stepen_najvece = 100;
49 }

51 if (hiljada > najveca) {
    najveca = hiljada;
53     stepen_najvece = 1000;
    }

55 /* Racunanje najmanje cifre. */
57 najmanja = jedinica;
    stepen_najmanje = 1;

59 if (desetica < najmanja) {
61     najmanja = desetica;
    stepen_najmanje = 10;
63 }

65 if (stotina < najmanja) {
    najmanja = stotina;
67     stepen_najmanje = 100;
    }

69 if (hiljada < najmanja) {
71     najmanja = hiljada;
    stepen_najmanje = 1000;
73 }

75 /* Ideja: U broju 4179, najmanja cifra je 1 i njen stepen je 100,
    a najveca cifra je 9 i njen stepen je 1. Zamena mesta se vrši
77     tako što se oduzme 9 i doda 1, a zatim oduzme 100 i doda 900. */
    rezultat = n_abs - najveca * stepen_najvece
79                 + najmanja * stepen_najvece
                - najmanja * stepen_najmanje
81                 + najveca * stepen_najmanje;

83 /* Ako je pocetni broj bio negativan i rezultat treba da bude
    negativan. */
85 if(n < 0)
    rezultat = -rezultat;

87 /* Ispis rezultata. */
89 printf("Rezultat: %d\n", rezultat);

91 return 0;
}
```

### Rešenje 2.1.15

```

1 #include <stdio.h>
  #include <stdlib.h>
3
```

```

5 int main()
6 {
7     /* Deklaracija potrebnih promenljivih. */
8     int n;
9     char jedinica, desetica, stotina, hiljada;
10
11     /* Ucitava se vrednost broja n. */
12     printf("Unesite cetvorocifreni broj: ");
13     scanf("%d", &n);
14
15     /* Da bi program radio ispravno i za negativne vrednosti, uzima
16        se apsolutna vrednost broja n. */
17     n = abs(n);
18
19     /* Vrsi se provera ispravnosti ulaznih podataka. */
20     if (n < 1000 || n > 9999) {
21         printf("Greska: niste uneli cetvorocifreni broj.\n");
22         return -1;
23     }
24
25     /* Izdvajaju se cifre broja n. */
26     jedinica = n % 10;
27     desetica = (n / 10) % 10;
28     stotina = (n / 100) % 10;
29     hiljada = n / 1000;
30
31     /* Ispis rezultata. */
32     if (hiljada <= stotina && stotina <= desetica
33         && desetica <= jedinica)
34         printf("Cifre su uredjene neopadajuce. \n");
35     else if (hiljada >= stotina && stotina >= desetica
36         && desetica >= jedinica)
37         printf("Cifre su uredjene nerastuce. \n");
38     else
39         printf("Cifre nisu uredjene.\n");
40
41     return 0;
42 }

```

### Rešenje 2.1.16

```

1 #include<stdio.h>
2
3 int main()
4 {
5     /* Deklaracija potrebnih promenljivih. */
6     float xa, ya, xb, yb;
7
8     /* Ucitavaju se koordinate tacaka A i B. */
9     printf("Unesite koordinate tacke A: ");
10    scanf("%f%f", &xa, &ya);

```

```
12 printf("Unesite koordinate tacke B: ");
   scanf("%f%f", &xb, &yb);

14

16 /* Proverava se da li su obe tacke u istom kvadrantu. */
   if ((xa >= 0 && ya >= 0 && xb >= 0 && yb >= 0) ||
       (xa <= 0 && ya >= 0 && xb <= 0 && yb >= 0) ||
18       (xa >= 0 && ya <= 0 && xb >= 0 && yb <= 0) ||
       (xa <= 0 && ya <= 0 && xb <= 0 && yb <= 0)) {
20     printf("Tacke se nalaze u istom kvadrantu.\n");
   } else {
22     printf("Tacke se ne nalaze u istom kvadrantu.\n");
   }

24   return 0;
26 }
```

### Rešenje 2.1.17

```
1 #include<stdio.h>

3 int main()
{
5   /* Deklaracija potrebnih promenljivih. */
   float xa, ya, xb, yb, xc, yc;
7   float k, n;

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

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

15  printf("Unesite koordinate tacke C: ");
17  scanf("%f%f", &xc, &yc);

19  /* Ako su bilo koje dve tacke jednake, onda se sigurno sve tri
     nalaze na jednoj pravoj. */
21  if ((xa == xb && ya == yb) ||
       (xa == xc && ya == yc) || (xb == xc && yb == yc)) {
23    printf("Tacke se nalaze na istoj pravoj.\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 xb jednako xa, ova prava je paralelna sa y
     osom i k ima vrednost beskonacno, a n ima vrednost 0, tj.
31  jednačina prave je x = xa (sto je isto sto i x = xb). Da bi se
     izbeglo deljenje nulom (xb-xa), ovaj slucaj se posebno
```

```

33     obradjuje. */
34     if (xb != xa) {
35         k = (yb - ya) / (xb - xa);
36         n = ya - k * xa;
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 (yc == k * xc + n)
40             printf("Tacke se nalaze na istoj pravoj.\n");
41         else
42             printf("Tacke se ne nalaze na istoj pravoj.\n");
43     } else {
44         /* Proverava se da li se i tacka C nalazi na pravoj x = xb. */
45         if (xc == xb)
46             printf("Tacke se nalaze na istoj pravoj.\n");
47         else
48             printf("Tacke se ne nalaze na istoj pravoj.\n");
49     }

51     /* II nacin: Tacke su kolinearne ako je:
52        |xa ya 1 |
53        |xb yb 1 | = 0
54        |xc yc 1 |
55        odnosno, ako je:
56        xa*yb + ya*xc + xb*yc - ya*xb - xa*yc - yb*xc = 0

57        if(xa*yb + ya*xc + xb*yc - ya*xb - xa*yc - yb*xc == 0)
58            printf("Tacke se nalaze na istoj pravoj. \n");
59        else
60            printf("Tacke se ne nalaze na istoj pravoj. \n"); */

61     return 0;
62 }

```

### Rešenje 2.1.18

```

#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) {
        /* I slucaj: intervali se seku i [a1,b1] je pre [a2,b2]. */
16        printf("Duzina preseka: %d\n", b1 - a2);
    }
}

```

```

18     printf("Presecni interval: [%d, %d]\n", a2, b1);
    printf("Duzina koju pokrivaju: %d\n", b2 - a1);
    printf("Najmanji interval: [%d, %d]\n", a1, b2);
20 } else if (a2 <= a1 && b2 >= a1) {
    /* II slucaj: intervali se seku i [a2,b2] je pre [a1,b1]. */
22     printf("Duzina preseka:: %d\n", b2 - a1);
    printf("Presecni interval: [%d, %d]\n", a1, b2);
24     printf("Duzina koju pokrivaju: %d\n", b1 - a2);
    printf("Najmanji interval: [%d, %d]\n", a2, b1);
26 } else if (a1 >= a2 && b1 <= b2) {
    /* III slucaj: interval [a1,b1] se nalazi unutar [a2,b2]. */
28     printf("Duzina preseka:: %d\n", b1 - a1);
    printf("Presecni interval: [%d, %d]\n", a1, b1);
30     printf("Duzina koju pokrivaju: %d\n", b2 - a2);
    printf("Najmanji interval: [%d, %d]\n", a2, b2);
32 } else if (a2 >= a1 && b2 <= b1) {
    /* IV slucaj: interval [a2,b2] se nalazi unutar [a1,b1]. */
34     printf("Duzina preseka:: %d\n", b2 - a2);
    printf("Presecni interval: [%d, %d]\n", a2, b2);
36     printf("Duzina koju pokrivaju: %d\n", b1 - a1);
    printf("Najmanji interval: [%d, %d]\n", a1, b1);
38 } else {
    /* V slucaj: intervali su disjunktni. */
40     printf("Duzina preseka:: 0\n");
    printf("Presecni interval: prazan\n");
42     printf("Duzina koju pokrivaju: %d\n", b1 - a1 + b2 - a2);
    if (a1 < a2)
44         printf("Najmanji interval: [%d, %d]\n", a1, b2);
    else
46         printf("Najmanji interval: [%d, %d]\n", a2, b1);
    }
48
50     return 0;
}

```

### Rešenje 2.1.19

```

1 #include <stdio.h>
   #include <math.h>
3
4 int main()
5 {
6     /* Deklaracija potrebnih promenljivih. */
7     float a, b, c;
8     float D;
9
10    /* Ucitavaju se koeficijenti kvadratne jednacine. */
11    printf("Unesite koeficijente A, B i C:");
    scanf("%f%f%f", &a, &b, &c);
13
14    /* Racunaju se resenja jednacine u zavisnosti od vrednosti

```



```

15     koeficijenta 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
34         /* U zavisnosti od vrednosti diskriminante, ispisuje se
35            rezultat. */
36         if (D < 0) {
37             printf("Jednacina nema realnih resenja\n");
38         } else if (D > 0) {
39             printf("Jednacina ima dva realna resenja %.2f i %.2f\n",
40                 (-b + sqrt(D)) / (2 * a), (-b - sqrt(D)) / (2 * a));
41         } else {
42             printf("Jednacina ima jedinstveno realno resenje %.2f\n",
43                 -b / (2 * a));
44         }
45     }
46
47     return 0;
48 }

```

### Rešenje 2.1.20

```

1  #include <stdio.h>
2
3  int main()
4  {
5      /* Deklaracija potrebnih promenljivih. */
6      int k, broj;
7
8      /* Ucitava se trazena pozicija. */
9      printf("Unesite k: ");
10     scanf("%d", &k);
11
12     /* Vrsi se provera ispravnosti ulaznih podataka. */
13     if (k < 1 || k > 189) {
14         printf("Greska: neispravan unos pozicije.\n");
15     }
16 }

```

## 2 Kontrola toka

```
15     return -1;
16 }
17
18 /* Racuna se rezultat. */
19 if (k < 10) {
20     /* I slucaj: trazi se jednocifreni broj. */
21     printf("Na %d-toj poziciji je broj %d.\n", k, k);
22 } else {
23     /* II slucaj: trazi se dvocifreni broj. */
24
25     /* Ideja: izracunati broj na koji pokazuje pozicija k. Zatim,
26        ako je k parno, uzeti cifru desetica tog broja, a ako je k
27        neparno, uzeti cifru jedinica tog broja.
28
29        Na primer, za k=14 i k=15, broj koji se nalazi na ovim
30        pozicijama je 12, pa u slucaju da je k=14, treba ispisati 1,
31        a u slucaju da je k=15, treba ispisati 2. */
32
33     /* Odredjivanje odgovarajuceg broja: Kada bi niz izgledao
34        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
36        vrednost detet jer je prvi broj u nizu desетка.) Na primer:
37        k=1, broj =  $9 + 1/2 + 1 = 9 + 0 + 1 = 10$  k=2, broj =  $9 + 2/2$ 
38        = 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,
40        potrebno je i njih uzeti u obzir - odatle: broj =  $9 +$ 
41         $(k-9)/2 + 1$  za neparne vrednosti k, odnosno broj =  $9 +$ 
42         $(k-9)/2$  za parne vrednosti k. */
43     if (k % 2 != 0) {
44         broj = 9 + (k - 9) / 2;
45         printf("Na %d-toj poziciji je broj %d.\n", k, broj % 10);
46     } else {
47         broj = 9 + (k - 9) / 2 + 1;
48         printf("Na %d-toj poziciji je broj %d.\n", k, broj / 10);
49     }
50 }
51
52 return 0;
53 }
```

### Rešenje 2.1.21

```
1 #include <stdio.h>
2 #include <math.h>
3
4 int main()
5 {
6     /* Deklaracija potrebnih promenljivih. */
7     float x, Fx;
8     int k;
9 }
```

```

11  /* Ucitavaju se vrednosti x i k. */
    printf("Unesite redom x i k: ");
    scanf("%f %d", &x, &k);
13
15  /* Vrsi se provera ispravnosti ulaznih podataka. */
    if (k < 1 || k > 3) {
        printf("Greska: nedozvoljena vrednost za k.\n");
        return 0;
17    }
19
21  /* U zavisnosti od vrednosti k, data funkcija ce se izracunati
    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
29  /* Ispis rezultata. Napomena: ispis realnih brojeva sa %g
    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
    0.9999 kao 0.9999. */
31    printf("F(%g,%d)=%.2f\n", x, k, Fx);
33
    return 0;
35 }

```

### Rešenje 2.1.22

```

1  #include <stdio.h>
3
4  int main()
5  {
6      /* Deklaracija potrebnih promenljivih. */
7      int dan;
8
9      /* Ucitava se redni broj dana u nedelji. */
10     printf("Unesite broj: ");
11     scanf("%d", &dan);
12
13     /*I nacin: koriscenjem if-else naredbe.
14     if(dan == 1)
15         printf("ponedeljak\n");
16     else if(dan == 2)
17         printf("utorak\n");
18     else if(dan == 3)
19         printf("sreda\n");
20     else if(dan == 4)
21         printf("cetvrtak\n");
22     else if(dan == 5)
23         printf("petak\n");

```

## 2 Kontrola toka

---

```
23     else if(dan == 6)
24         printf("subota\n");
25     else if(dan == 7)
26         printf("nedelja\n");
27     else
28         printf("Greska: neispravan unos dana.\n"); */
29
30     /* II nacin: koriscenjem switch naredbe.*/
31     switch (dan) {
32     case 1:
33         /* Ako dan ima vrednost 1, ispisuje se ponedeljak. */
34         printf("ponedeljak\n");
35
36         /* Ako se naredba break ne navede, izvorsice se i sledeca
37            naredba, tj. ispis ce biti "ponedeljak utorak". */
38         break;
39     case 2:
40         /* Postupak se ponavlja i za ostale dane. */
41         printf("utorak\n");
42         break;
43     case 3:
44         printf("sreda\n");
45         break;
46     case 4:
47         printf("cetvrtak\n");
48         break;
49     case 5:
50         printf("petak\n");
51         break;
52     case 6:
53         printf("subota\n");
54         break;
55     case 7:
56         printf("nedelja\n");
57         break;
58     default:
59         /* Ako vrednost promenljive dan nije ni jedna od vrednosti
60            izmedju 1 i 7, onda je uneta vrednost neispravna. */
61         printf("Greska: neispravan unos dana.\n");
62     }
63
64     return 0;
65 }
```

### Rešenje 2.1.23

```
1  #include <stdio.h>
2
3  int main()
4  {
5      /* Deklaracija potrebnih promenljivih. */
```

```
7   char c;

/* Ucitava se jedan karakter. */
9   printf("Unesite jedan karakter:");
   scanf("%c", &c);

11  /* 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':
16     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   return 0;
32 }
33 }
```

### Rešenje 2.1.24

```
1  #include <stdio.h>

3  int main()
4  {
5     /* Deklaracija potrebnih promenljivih. */
6     char op;
7     int x, y;

9     /* Ucitava se izraz. */
10    printf("Unesite izraz: ");
11    scanf("%d %c %d", &x, &op, &y);

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    }
```

```
21  case '*':
    printf("Rezultat je: %d\n", x * y);
23  break;
    case '/':
25      if (y == 0)
        printf("Greska: deljenje nulom.\n");
27      else
        printf("Rezultat je: %d\n", x / y);
29      break;
    case '%':
31      printf("Rezultat je: %d\n", x % y);
        break;
33  default:
        printf("Greska: nepoznat operator.\n");
35  }

37  return 0;
}
```

### Rešenje 2.1.25

```
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
18                 zima. */
19             printf("zima\n");
20             break;
21         case 3:
22             /* Ako je mesec mart, onda se godisnje doba odredjuje u
23                 zavisnosti od dana u mesecu. */
24             if (dan < 21)
25                 printf("zima\n");
26             else
27                 printf("prolece\n");
28             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:
    /* Ako je mesec jun, onda se godisnje doba odredjuje u
37     zavisnosti od dana u mesecu. */
        if (dan < 21)
39             printf("prolece\n");
        else
41             printf("leto\n");
        break;
43 case 7:
44 case 8:
    /* Ako je mesec jul ili avgust, onda je sigurno u pitanju
45     leto. */
        printf("leto\n");
47         break;
49 case 9:
    /* Ako je mesec septembar, onda se godisnje doba odredjuje u
51     zavisnosti od dana u mesecu. */
        if (dan < 23)
53             printf("leto\n");
        else
55             printf("jesen\n");
        break;
57 case 10:
58 case 11:
    /* Ako je mesec oktobar ili novembar, onda je sigurno u pitanju
59     jesen. */
        printf("jesen\n");
61         break;
63 case 12:
    /* Ako je mesec decembar, onda se godisnje doba odredjuje u
65     zavisnosti od dana u mesecu. */
        if (dan < 22)
67             printf("jesen\n");
        else
69             printf("zima\n");
71     }
    return 0;
73 }

```

### Rešenje 2.1.26

```

#include <stdio.h>
2
int main()
4 {
    /* Deklaracija potrebnih promenljivih. */

```

```
6   int godina;
   int mesec;
8   int prestupna;

10  /* Ucitava se vrednost godine. */
   printf("Unesite godinu: ");
12  scanf("%d", &godina);

14  /* Vrsi se provera ispravnosti ulaznih podataka. */
   if (godina < 0) {
16     printf("Greska: neispravan unos godine.\n");
     return -1;
18  }

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

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

30  /* U zavisnosti od vrednosti meseca, ispisuje se odgovarajuci
     rezultat. */
32  switch (mesec) {
   case 1:
34     printf("Januar, 31 dan\n");
     break;
36     case 2:
         if (prestupna)
38             printf("Februar, 29 dana\n");
         else
40             printf("Februar, 28 dana\n");
         break;
42     case 3:
         printf("Mart, 31 dan\n");
         break;
44     case 4:
         printf("April, 30 dana\n");
         break;
46     case 5:
         printf("Maj, 31 dan\n");
         break;
50     case 6:
         printf("Jun, 30 dana\n");
         break;
52     case 7:
         printf("Jul, 31 dan\n");
         break;
54     case 8:
         break;
```



```

58     printf("Avgust, 31 dan\n");
        break;
60 case 9:
    printf("Septembar, 30 dana\n");
62     break;
    case 10:
64         printf("Oktobar, 31 dan\n");
            break;
66 case 11:
    printf("Novembar, 30 dana\n");
68     break;
    case 12:
70         printf("Decembar, 31 dan\n");
            break;
72 default:
    printf("Greska: neispravan unos meseca.\n");
74     return -1;
}

76 return 0;
78 }

```

### Rešenje 2.1.27

```

1  #include <stdio.h>
3  int main()
4  {
5      /* Deklaracija potrebnih promenljivih. */
        int dan, mesec, godina, dozvoljeni_broj_dana;
7
8      /* Ucitava se datum. */
9      printf("Unesite datum: ");
        scanf("%d.%d.%d", &dan, &mesec, &godina);
11
12     /* Vrsi se provera korektnosti vrednosti unete godine. */
13     if (godina < 0) {
        printf("Datum nije korektan.\n");
15         return 0;
    }
17
18     /* Vrsi se provera korektnosti vrednosti unetog meseca. */
19     if (mesec < 1 || mesec > 12) {
        printf("Datum nije korektan.\n");
21         return 0;
    }
23
24     /* Vrsi se provera korektnosti vrednosti unetog dana. */
25     switch (mesec) {
        case 1:
27         case 3:

```

## 2 Kontrola toka

```

29  case 5:
30  case 7:
31  case 8:
32  case 10:
33  case 12:
34      /* Dozvoljeni broj dana za januar, mart, maj, jul, avgust,
35         oktobar i decembar je 31 */
36      dozvoljeni_broj_dana = 31;
37      break;
38  case 2:
39      /* Dozvoljeni broj dana za februar je 28 ili 29 u zavisnosti od
40         toga da li je godina prestupna ili ne. */
41      if ((godina % 4 == 0 && godina % 100 != 0) || godina % 400 == 0)
42          dozvoljeni_broj_dana = 29;
43      else
44          dozvoljeni_broj_dana = 28;
45      break;
46  case 4:
47  case 6:
48  case 9:
49  case 11:
50      /* Dozvoljeni broj dana za april, jun, septembar i novembar je
51         30. */
52      dozvoljeni_broj_dana = 30;
53      break;
54  }
55
56  if (dan < 0 || dan > dozvoljeni_broj_dana) {
57      printf("Datum nije korektan.\n");
58      return 0;
59  }
60
61  /* Kako su sve provere korektnosti prosle, datum se smatra
62     korektnim. */
63  printf("Datum je korektan.\n");
64
65  return 0;
66 }
```

### Rešenje 2.1.28

```

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 datum. */
10     printf("Unesite datum: ");
```

```
11  scanf("%d.%d.%d.", &dan, &mesec, &godina);
13
14  /* Racunaju se dan, mesec i godina prethodnog dana. */
15  prethodni_dan = dan - 1;
16  prethodni_mesec = mesec;
17  prethodni_godina = godina;
18
19  /* Ako je potrebno, vrse se korekcije. */
20  if (prethodni_dan == 0) {
21      prethodni_mesec = mesec - 1;
22      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 2.1.29

Rešenje je analogno rešenju zadatka 2.1.28.

### Rešenje 2.1.30

```
1  #include<stdio.h>
2  #include<stdlib.h>
3
4  int main()
5  {
6      /* Deklaracija potrebnih promenljivih. */
7      int k, l, m, n;
8
9      /* Ucitavaju se vrednosti pozicija na tabli. */
10     printf("Unesite (k,l): ");
11     scanf("%d%d", &k, &l);
12
13     printf("Unesite (m,n): ");
14     scanf("%d%d", &m, &n);
15
16     /* Vrsi se provera ispravnosti ulaznih podataka. */
17     if (k < 1 || k > 8 || l < 1 || l > 8 ||
18         m < 1 || m > 8 || n < 1 || n > 8) {
19         printf("Greska: neispravna pozicija.\n");
20         return -1;
21     }
22
23     if(k == m && l == n){
24         printf("Greska: pozicije moraju biti razlicite.\n");
25         return -1;
26     }
27
28     /* Proverava se da li su (k,l) i (m,n) iste boje. Polja su iste
29        boje ako su: 1) oba reda parna i obe kolone parne ILI 2) oba
30        reda neparna i obe kolone neparne. */
31     if (((k % 2 == m % 2) && (l % 2 == n % 2))
32         || ((k % 2 != m % 2) && (l % 2 != n % 2)))
33         printf("Polja su iste boje.\n");
34     else
35         printf("Polja su razlicite boje.\n");
36
37     /* Proverava se da li kraljica sa (k,l) napada polje (m,n).
38        Kraljica napada polje u sledecim situacijama:
39        1) Ako se nalaze u istom redu (k==m)
40        2) Ako se nalaze u istoj koloni (l==n)
41        3) Ako se nalaze na istoj dijagonali. Dijagonala moze biti:
42           a) paralelna glavnoj dijagonali (abs(k-l) == abs(m-n))
43           b) paralelna sporednoj dijagonali (k+l == m+n) */
44     if ((k == m) || (l == n) || (abs(k - l) == abs(m - n))
45         || (k + l == m + n)){
46         printf("Kraljica sa (%d, %d) ugrozava (%d, %d).\n",
47             k, l, m, n);
48     }
49     else {
50         printf("Kraljica sa (%d, %d) ne ugrozava (%d, %d).\n",
```

```

52         k, l, m, n);
53     }
54     /* Proverava se da li konj sa (k, l) napada polje (m, n). Postoji
55        8 mogucih vrednosti za polja koja konj napada. Vrsi se
56        provera da li je (m,n) jednako nekom od tih polja. */
57     if ((abs(k-m) == 2 && abs(n-l) == 1) || (abs(n-l) == 2 && abs(m-k)
58         == 1))
59         printf("Konj sa (%d, %d) ugrozava (%d, %d).\n",
60             k, l, m, n);
61     else
62         printf("Konj sa (%d, %d) ne ugrozava (%d, %d).\n",
63             k, l, m, n);
64     return 0;
65 }

```

## 2.3 Petlje

**Zadatak 2.3.1** Napisati program koji pet puta ispisuje tekst *Mi volimo da programiramo.*

*Primer 1*

```

INTERAKCIJA SA PROGRAMOM:
Mi volimo da programiramo.
Mi volimo da programiramo.
Mi volimo da programiramo.
Mi volimo da programiramo.
Mi volimo da programiramo.

```

[Rešenje 2.3.1]

**Zadatak 2.3.2** Napisati program koji učitava pozitivan ceo broj  $n$  i  $n$  puta ispisuje tekst *Mi volimo da programiramo.* U slučaju neispravnog unosa, ispisati odgovarajuću poruku o grešci.

*Primer 1*

```

INTERAKCIJA SA PROGRAMOM:
Unesite broj n: 6
Mi volimo da programiramo.
Mi volimo da programiramo.
Mi volimo da programiramo.
Mi volimo da programiramo.
Mi volimo da programiramo.
Mi volimo da programiramo.

```

*Primer 2*

```

INTERAKCIJA SA PROGRAMOM:
Unesite broj n: 0
Greska: pogresan unos broja n.

```

### Primer 3

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesite broj n: -5  
|| Greska: pogresan unos broja n.
```

### Primer 4

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesite broj n: 1  
|| Mi volimo da programiramo.
```

[Rešenje 2.3.2]

**Zadatak 2.3.3** Napisati program koji učitava nenegativan ceo broj  $n$  a potom ispisuje sve cele brojeve od 0 do  $n$ . U slučaju neispravnog unosa, ispisati odgovarajuću poruku o grešci.

### Primer 1

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesite broj n: 4  
|| 0 1 2 3 4
```

### Primer 2

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesite broj n: -10  
|| Greska: pogresan unos broja n.
```

[Rešenje 2.3.3]

**Zadatak 2.3.4** Napisati program koji učitava dva cela broja  $n$  i  $m$ , ( $n \leq m$ ) i ispisuje sve cele brojeve iz intervala  $[n, m]$ .

- (a) Koristiti `while` petlju.
- (b) Koristiti `for` petlju.
- (c) Koristiti `do-while` petlju.

U slučaju neispravnog unosa, ispisati odgovarajuću poruku o grešci.

### Primer 1

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesite granice intervala: -2 4  
|| -2 -1 0 1 2 3 4
```

### Primer 2

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesite granice intervala: 10 6  
|| Greska: pogresan unos granica.
```

[Rešenje 2.3.4]

**Zadatak 2.3.5** Napisati program koji učitava nenegativan ceo broj  $n$  i izračunava njegov faktoriyel. U slučaju neispravnog unosa, ispisati odgovarajuću poruku o grešci.

*Primer 1*

```

|| INTERAKCIJA SA PROGRAMOM:
|| Unesite broj n: 18
|| 18! = 6402373705728000

```

*Primer 2*

```

|| INTERAKCIJA SA PROGRAMOM:
|| Unesite broj n: 8
|| 8! = 40320

```

*Primer 3*

```

|| INTERAKCIJA SA PROGRAMOM:
|| Unesite broj n: 40
|| Pri racunanju 40! ce doći do prekoracenja.

```

*Primer 4*

```

|| INTERAKCIJA SA PROGRAMOM:
|| Unesite broj n: -5
|| Greska: neispravan unos.

```

[Rešenje 2.3.5]

**Zadatak 2.3.6** Napisati program koji učitava realan broj  $x$  i ceo nenegativan broj  $n$  i izračunava  $n$ -ti stepen broja  $x$ , tj.  $x^n$ . U slučaju neispravnog unosa, ispisati odgovarajuću poruku o grešci.

*Primer 1*

```

|| INTERAKCIJA SA PROGRAMOM:
|| Unesite redom brojeve x i n: 4 3
|| Rezultat: 64.00000

```

*Primer 2*

```

|| INTERAKCIJA SA PROGRAMOM:
|| Unesite redom brojeve x i n: 5.8 5
|| Rezultat: 6563.56768

```

*Primer 3*

```

|| INTERAKCIJA SA PROGRAMOM:
|| Unesite redom brojeve x i n: 11.43 -6
|| Greska: neispravan unos broja n.

```

*Primer 4*

```

|| INTERAKCIJA SA PROGRAMOM:
|| Unesite redom brojeve x i n: 11.43 0
|| Rezultat: 1.00000

```

[Rešenje 2.3.6]

**Zadatak 2.3.7** Napisati program koji učitava realan broj  $x$  i ceo broj  $n$  i izračunava  $n$ -ti stepen broja  $x$ .

*Primer 1*

```

|| INTERAKCIJA SA PROGRAMOM:
|| Unesite redom brojeve x i n: 2 -3
|| Rezultat: 0.125

```

*Primer 2*

```

|| INTERAKCIJA SA PROGRAMOM:
|| Unesite redom brojeve x i n: -3 2
|| Rezultat: 9.000

```

[Rešenje 2.3.7]

**Zadatak 2.3.8** Pravi delioci celog broja su svi delioci sem jedinice i samog tog broja. Napisati program za uneti pozitivan ceo broj  $n$  ispisuje sve njegove prave delioce. U slučaju neispravnog unosa, ispisati odgovarajuću poruku o grešci.

## 2 Kontrola toka

---

### Primer 1

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesite broj n: 100  
|| 2 4 5 10 20 25 50
```

### Primer 2

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesite broj n: -6  
|| Greska: neispravan unos.
```

[Rešenje 2.3.8]

**Zadatak 2.3.9** Napisati program koji za uneti ceo broj ispisuje broj dobijen uklanjanjem svih nula sa desne strane unetog broja.

### Primer 1

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesite broj: 12000  
|| 12
```

### Primer 2

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesite broj: 0  
|| 0
```

### Primer 3

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesite broj: -1400  
|| -14
```

### Primer 4

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesite broj: 147  
|| 147
```

[Rešenje 2.3.9]

**Zadatak 2.3.10** Napisati program koji učitava ceo broj i ispisuje njegove cifre u obrnutom poretku.

### Primer 1

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesite ceo broj: 6789  
|| 9 8 7 6
```

### Primer 2

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesite ceo broj: -892345  
|| 5 4 3 2 9 8
```

[Rešenje 2.3.10]

**Zadatak 2.3.11** Napisati program koji za uneti pozitivan ceo broj ispisuje da li je on deljiv sumom svojih cifara. U slučaju neispravnog unosa, ispisati odgovarajuću poruku o grešci.

### Primer 1

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesite broj: 12  
|| Broj 12 je deljiv sa 3.
```

### Primer 2

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesite broj: 2564  
|| Broj 2564 nije deljiv sa 17.
```



*Primer 3*

```

|| INTERAKCIJA SA PROGRAMOM:
|| Unesite broj: -4
|| Greska: neispravan ulaz.

```

*Primer 4*

```

|| INTERAKCIJA SA PROGRAMOM:
|| Unesite broj: 0
|| Greska: neispravan ulaz.

```

[Rešenje 2.3.11]

**Zadatak 2.3.12** Knjigovođa vodi evidenciju o transakcijama jedne firme i treba da napiše izveštaj o godišnjem poslovanju te firme. Firma je tokom godine imala  $t$  transakcija. Transakcije su predstavljene celim brojevima i u slučaju da je vrednost transakcije pozitivna, ta transakcija označava prihod firme, a u slučaju da je negativna rashod. Napisati program koji učitava nenegativan ceo broj  $t$  i podatke o  $t$  transakcija i zatim izračunava i ispisuje ukupan prihod, ukupan rashod i zaradu, odnosno gubitak koji je firma ostvarila tokom godine. U slučaju neispravnog unosa, ispisati odgovarajuću poruku o grešci.

*Primer 1*

```

|| INTERAKCIJA SA PROGRAMOM:
|| Unesite broj t: 7
|| Unesite transakcije:
|| 8 -50 45 2007 -67 -123 14
|| Prihod: 2074
|| Rashod: -240
|| Zarada: 1834

```

*Primer 2*

```

|| INTERAKCIJA SA PROGRAMOM:
|| Unesite broj t: 5
|| Unesite transakcije:
|| -5 -20 -4 -200 -8
|| Prihod: 0
|| Rashod: -237
|| Gubitak: 237

```

*Primer 3*

```

|| INTERAKCIJA SA PROGRAMOM:
|| Unesite broj t: -6
|| Greska: neispravan unos.

```

*Primer 4*

```

|| INTERAKCIJA SA PROGRAMOM:
|| Unesite broj n: 0
|| Nema evidentiranih transakcija.

```

*Primer 5*

```

|| INTERAKCIJA SA PROGRAMOM:
|| Unesite broj t: 2
|| Unesite transakcije:
|| 120 -120
|| Prihod: 120
|| Rashod: -120
|| Zarada: 0

```

[Rešenje 2.3.12]

**Zadatak 2.3.13** Napisati program koji učitava pozitivan ceo broj  $n$ , a potom i  $n$  celih brojeva. Izračunati i ispisati zbir onih brojeva koji su istovremeno neparni i negativni. U slučaju neispravnog unosa, ispisati odgovarajuću poruku o grešci.

### Primer 1

```
INTERAKCIJA SA PROGRAMOM:
Unesite broj n: 5
Unesite n brojeva:
1 -5 -6 3 -11
Zbir neparnih i negativnih: -16
```

### Primer 2

```
INTERAKCIJA SA PROGRAMOM:
Unesite broj n: -4
Greska: neispravan unos.
```

### Primer 3

```
INTERAKCIJA SA PROGRAMOM:
Unesite broj n: 4
Unesite n brojeva:
5 8 13 17
Zbir neparnih i negativnih: 0
```

### Primer 4

```
INTERAKCIJA SA PROGRAMOM:
Unesite broj n: 0
Greska: neispravan unos.
```

[Rešenje 2.3.13]

**Zadatak 2.3.14** Napisati program koji učitava pozitivan ceo broj  $n$ , a potom  $n$  celih brojeva i računa i ispisuje sumu brojeva koji su deljivi sa 5, a nisu deljivi sa 7. U slučaju neispravnog unosa, ispisati odgovarajuću poruku o grešci.

### Primer 1

```
INTERAKCIJA SA PROGRAMOM:
Unesite broj n: 5
Unesite n brojeva: :2 35 5 -175 -20
Suma je -15.
```

### Primer 2

```
INTERAKCIJA SA PROGRAMOM:
Unesite broj n: -3
Greska: neispravan unos.
```

### Primer 3

```
INTERAKCIJA SA PROGRAMOM:
Unesite broj n: 10
Unesite n brojeva:
-5 6 175 -20 -25 -8 42 245 1 6
Suma je -50.
```

### Primer 4

```
INTERAKCIJA SA PROGRAMOM:
Unesite broj n: 6
Unesite brojeve:
2205 -1904 2 7 -540 5
Suma je -535.
```

[Rešenje 2.3.14]

**Zadatak 2.3.15** Napisati program koji učitava cele brojeve sve dok se ne unese nula i ispisuje proizvod onih unetih brojeva koji su pozitivni.

### Primer 1

```
INTERAKCIJA SA PROGRAMOM:
Unesite brojeve:
-87 12 -108 -13 56 0
Proizvod pozitivnih brojeva je 672.
```

### Primer 2

```
INTERAKCIJA SA PROGRAMOM:
Unesite brojeve: 0
Nije unet nijedan broj.
```

### Primer 3

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite brojeve:
|| -5 -200 -43 0
|| Medju unetim brojevima nema pozitivnih.
```

### Primer 4

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite brojeve: 1 0
|| Proizvod pozitivnih brojeva je 1.
```

[Rešenje 2.3.15]

**Zadatak 2.3.16** Napisati program koji za uneti ceo broj proverava i ispisuje da li se cifra 5 nalazi u njegovom zapisu.

### Primer 1

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite broj: 1857
|| Broj 1857 sadrzi cifru 5.
```

### Primer 2

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite broj: 84
|| Broj 84 ne sadrzi cifru 5.
```

### Primer 3

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite broj: -2515
|| Broj -2515 sadrzi cifru 5.
```

[Rešenje 2.3.16]

**Zadatak 2.3.17** Napisati program koji učitava cele brojeve sve do unosa broja nula, a zatim izračunava i ispisuje aritmetičku sredinu unetih brojeva na četiri decimale.

### Primer 1

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite brojeve: 8 5 6 3 0
|| Aritmeticka sredina: 5.5000
```

### Primer 2

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite brojeve: 0
|| Nisu uneti brojevi.
```

### Primer 3

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite brojeve: 762 -12 800 2010 -356 899 -101 0
|| Aritmeticka sredina: 571.7143
```

[Rešenje 2.3.17]

**Zadatak 2.3.18** U prodavnici se nalaze artikali čije su cene pozitivni realni brojevi. Napisati program koji učitava cene artikala sve do unosa broja nula i izračunava i ispisuje prosečnu vrednost cena u radnji. U slučaju neispravnog unosa, ispisati odgovarajuću poruku o grešci.

### Primer 1

```
INTERAKCIJA SA PROGRAMOM:
Unesite cene: 8 5.2 6.11 3 0
Prosečna cena: 5.5775
```

### Primer 2

```
INTERAKCIJA SA PROGRAMOM:
Unesite cene: 6.32 -9
Greska: neispravan unos cene.
```

### Primer 3

```
INTERAKCIJA SA PROGRAMOM:
Unesite cene: 0
Nisu unete cene.
```

[Rešenje 2.3.18]

**Zadatak 2.3.19** Napisati program koji učitava pozitivan ceo broj  $n$ , a potom  $n$  realnih brojeva, a zatim određuje i ispisuje koliko puta je prilikom unosa došlo do promene znaka. U slučaju neispravnog unosa, ispisati odgovarajuću poruku o grešci.

### Primer 1

```
INTERAKCIJA SA PROGRAMOM:
Unesite broj n: 9
Unesite brojeve:
7.82 4.3 -1.2 56.8 -3.4 -72.1 8.9 11.2 -11.2
Broj promena je 5.
```

### Primer 2

```
INTERAKCIJA SA PROGRAMOM:
Unesite broj n: 5
Unesite brojeve:
-23.8 -11.2 0 5.6 7.2
Broj promena je 1.
```

### Primer 3

```
INTERAKCIJA SA PROGRAMOM:
Unesite broj n: -6
Greska: neispravan unos.
```

### Primer 4

```
INTERAKCIJA SA PROGRAMOM:
Unesite broj n: 0
Greska: neispravan unos.
```

[Rešenje 2.3.19]

**Zadatak 2.3.20** U prodavnici se nalazi  $n$  artikala čije su cene pozitivni realni brojevi. Napisati program koji učitava  $n$ , a potom i cenu svakog od  $n$  artikala i određuje i ispisuje najmanju cenu. U slučaju neispravnog unosa, ispisati odgovarajuću poruku o grešci.

### Primer 1

```
INTERAKCIJA SA PROGRAMOM:
Unesite broj artikla: 6
Unesite cene artikala:
12 3.4 90 100.53 53.2 12.8
Najmanja cena: 3.400000
```

### Primer 2

```
INTERAKCIJA SA PROGRAMOM:
Unesite broj artikla: 3
Unesite cene artikala:
4 -8 92
Greska: neispravan unos cene.
```

*Primer 3*

```

|| INTERAKCIJA SA PROGRAMOM:
|| Unesite broj artikla: -9
|| Greska: neispravan unos.

```

[Rešenje 2.3.20]

**Zadatak 2.3.21** Nikola želi da obradi bazu i da joj kupi jedan poklon u radnji. On na raspolaganju ima  $m$  dinara. U radnji se nalazi  $n$  artikala i zanima ga koliko ima artikala u radnji čija cena je manja ili jednaka  $m$ . Napisati program koji pomaže Nikoli da brzo odredi broj artikala. Program učitava realan nenegativan broj  $m$ , ceo nenegativan broj  $n$  i  $n$  pozitivnih realnih brojeva. Ispisati koliko artikala ima cenu čija je vrednost manja ili jednaka  $m$ . **NAPOMENA:** *Pretpostaviti da je unos ispravan.*

*Primer 1*

```

|| INTERAKCIJA SA PROGRAMOM:
|| Nikolin budzet: 12.37
|| Unesite broj artikala: 5
|| Unesite cene artikala: 11 54.13 6 13 8
|| Ukupno artikala: 3

```

*Primer 2*

```

|| INTERAKCIJA SA PROGRAMOM:
|| Nikolin budzet: 2
|| Unesite broj artikala: 4
|| Unesite cene artikala: 1 11 4.32 3
|| Ukupno artikala: 1

```

[Rešenje 2.3.21]

**Zadatak 2.3.22** Napisati program koji učitava ceo nenegativan broj  $n$ ,  $n$  celih brojeva i zatim izračunava i ispisuje tražene vrednosti. U slučaju neispravnog unosa, ispisati odgovarajuću poruku o grešci.

- (a) Broj sa najvećom cifrom desetica. Ukoliko ima više takvih, ispisati prvi.

*Primer 1*

```

|| INTERAKCIJA SA PROGRAMOM:
|| Unesite broj n: 5
|| Unesite brojeve:
|| 18 365 25 1 78
|| Broj sa najvećom cifrom desetica: 78.

```

*Primer 2*

```

|| INTERAKCIJA SA PROGRAMOM:
|| Unesite broj n: 8
|| Unesite brojeve:
|| 14 1576 -1267 -89 109 122 306 918
|| Broj sa najvećom cifrom desetica: -89.

```

*Primer 3*

```

|| INTERAKCIJA SA PROGRAMOM:
|| Unesite broj n: 4
|| Unesite brojeve:
|| 100 200 300 400
|| Broj sa najvećom cifrom desetica: 100.

```

*Primer 4*

```

|| INTERAKCIJA SA PROGRAMOM:
|| Unesite broj n: -12
|| Greska: neispravan unos.

```

## 2 Kontrola toka

---

- (b) Broj sa najvećim brojem cifara. Ukoliko ima više takvih, ispisati prvi.

### Primer 1

```
INTERAKCIJA SA PROGRAMOM:
Unesite broj n: 5
Unesite n brojeva: 18 -365 251 1 78
Najviše cifara ima broj -365.
```

### Primer 2

```
INTERAKCIJA SA PROGRAMOM:
Unesite broj n: 7
Unesite n brojeva:
3 892 18 21 639 742 85
Najviše cifara ima broj 892.
```

### Primer 3

```
INTERAKCIJA SA PROGRAMOM:
Unesite broj n: 0
Nisu uneti brojevi.
```

### Primer 4

```
INTERAKCIJA SA PROGRAMOM:
Unesite broj n: -7
Greska: neispravan unos.
```

### Primer 5

```
INTERAKCIJA SA PROGRAMOM:
Unesite broj n: 5
Unesite n brojeva: 0 1 2 -3 4
Najviše cifara ima broj 0.
```

### Primer 6

```
INTERAKCIJA SA PROGRAMOM:
Unesite broj n: 5
Unesite n brojeva: -5 4 -3 2 1
Najviše cifara ima broj -5.
```

- (c) Broj sa najvećom vodećom cifrom. Vodeća cifra je cifra najveće težine u zapisu broja. Ukoliko ima više takvih, ispisati prvi.

### Primer 1

```
INTERAKCIJA SA PROGRAMOM:
Unesite broj n: 5
Unesite n brojeva: 8 964 -32 511 27
Broj sa najvećom vodećom cifrom je 964.
```

### Primer 2

```
INTERAKCIJA SA PROGRAMOM:
Unesite broj n: 3
Unesite n brojeva: 0 0 0
Broj sa najvećom vodećom cifrom je 0.
```

### Primer 3

```
INTERAKCIJA SA PROGRAMOM:
Unesite broj n: 3
Unesite n brojeva: 41 669 -8
Broj sa najvećom vodećom cifrom je -8.
```

### Primer 4

```
INTERAKCIJA SA PROGRAMOM:
Unesite broj n: 0
Nisu uneti brojevi.
```

[Rešenje 2.3.22]

**Zadatak 2.3.23** Vršena su merenja nadmorskih visina na određenom delu teritorije i naučnike zanima razlika između najveće i najmanje nadmorske visine. Napisati program koji učitava cele brojeve sve do unosa 0 koji označavaju nadmorske visine i ispisuje razliku najveće i najmanje nadmorske visine.

*Primer 1*

```

|| INTERAKCIJA SA PROGRAMOM:
|| Unesite brojeve: 8 6 5 2 11 7 0
|| Razlika: 9

```

*Primer 2*

```

|| INTERAKCIJA SA PROGRAMOM:
|| Unesite brojeve: 8 -1 8 6 0
|| Razlika: 9

```

*Primer 3*

```

|| INTERAKCIJA SA PROGRAMOM:
|| Unesite brojeve: 0
|| Nisu unete nadmorske visine.

```

*Primer 4*

```

|| INTERAKCIJA SA PROGRAMOM:
|| Unesite brojeve: -500 0
|| Razlika: 0

```

*Primer 5*

```

|| INTERAKCIJA SA PROGRAMOM:
|| Unesite brojeve: -500 -300 -5000 0
|| Razlika: 4700

```

[Rešenje 2.3.23]

**Zadatak 2.3.24** Napisati program koji učitava ceo broj  $n$  ( $n > 1$ ), nenegativan ceo broj  $d$ , a zatim i  $n$  celih brojeva i izračunava i ispisuje koliko ima parova uzastopnih brojeva među unetim brojevima koji se nalaze na rastojanju  $d$ . Rastojanje između brojeva je definisano sa  $d(x, y) = |y - x|$ . U slučaju neispravnog unosa, ispisati odgovarajuću poruku o grešci.

*Primer 1*

```

|| INTERAKCIJA SA PROGRAMOM:
|| Unesite brojeve n i d: 5 2
|| Unesite n brojeva: 2 3 5 1 -1
|| Broj parova: 2

```

*Primer 2*

```

|| INTERAKCIJA SA PROGRAMOM:
|| Unesite brojeve n i d: 10 5
|| Unesite n brojeva:
|| -3 6 11 -20 -25 -8 42 37 1 6
|| Broj parova: 4

```

*Primer 3*

```

|| INTERAKCIJA SA PROGRAMOM:
|| Unesite brojeve n i d: 5 0
|| Unesite n brojeva: 1 1 1 1 1
|| Broj parova: 4

```

*Primer 4*

```

|| INTERAKCIJA SA PROGRAMOM:
|| Unesite brojeve n i d: 1 3
|| Greska: neispravan unos.

```

[Rešenje 2.3.24]

**Zadatak 2.3.25** Napisati program koji uneti pozitivan ceo broj transformiše tako što svaku parnu cifru u zapisu broja uveća za jedan. Ispisati dobijeni broj. U slučaju neispravnog unosa, ispisati odgovarajuću poruku o grešci.

### Primer 1

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesite broj: 2417  
|| Rezultat: 3517
```

### Primer 2

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesite broj: 138  
|| Rezultat: 139
```

### Primer 3

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesite broj: 59  
|| Rezultat: 59
```

[Rešenje 2.3.25]

**Zadatak 2.3.26** Napisati program koji učitava jedan ceo broj i zatim formira i ispisuje broj koji se dobija izbacivanjem svake druge cifre polaznog broja, idući sa desna na levo.

### Primer 1

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesite broj: 21854  
|| Rezultat: 284
```

### Primer 2

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesite broj: 18  
|| Rezultat: 8
```

### Primer 3

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesite broj: 1  
|| Rezultat: 1
```

### Primer 4

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesite broj: -67123  
|| Rezultat: -613
```

[Rešenje 2.3.26]

\* **Zadatak 2.3.27** Napisati program koji na osnovu unetog pozitivnog celog broja formira i ispisuje broj koji se dobija izbacivanjem cifara koje su u polaznom broju jednake zbiru svojih suseda. U slučaju neispravnog unosa, ispisati odgovarajuću poruku o grešci.

### Primer 1

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesite broj: 28631  
|| 2631
```

### Primer 2

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesite broj: 440  
|| 40
```

### Primer 3

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesite broj: -5  
|| Greska: neispravan unos.
```

[Rešenje 2.3.27]

\* **Zadatak 2.3.28** Broj je *palindrom* ukoliko se isto čita i sa leve i sa desne strane. Napisati program koji učitava pozitivan ceo broj i proverava da li je učitani broj palindrom. U slučaju neispravnog unosa, ispisati odgovarajuću poruku o grešci.



*Primer 1*

```

INTERAKCIJA SA PROGRAMOM:
Unesite broj: 25452
Broj je palindrom.

```

*Primer 2*

```

INTERAKCIJA SA PROGRAMOM:
Unesite broj: 895
Broj nije palindrom.

```

*Primer 3*

```

INTERAKCIJA SA PROGRAMOM:
Unesite broj: 5
Broj je palindrom.

```

[Rešenje 2.3.28]

**Zadatak 2.3.29** Fibonačijev niz počinje ciframa 0 i 1, a svaki član se dobija kao zbir prethodna dva. Napisati program koji učitava nenegativan ceo broj  $n$  i određuje i ispisuje  $n$ -ti član Fibonačijevog niza. Niz se indeksira počevši od nule. U slučaju neispravnog unosa, ispisati odgovarajuću poruku o grešci.

*Primer 1*

```

INTERAKCIJA SA PROGRAMOM:
Unesite broj n: 10
F[10] = 55

```

*Primer 2*

```

INTERAKCIJA SA PROGRAMOM:
Unesite broj n: -100
Greska: neispravan unos.

```

*Primer 3*

```

INTERAKCIJA SA PROGRAMOM:
Unesite broj n: 40
F[40] = 102334155

```

*Primer 4*

```

INTERAKCIJA SA PROGRAMOM:
Unesite broj n: 20
F[20] = 6765

```

[Rešenje 2.3.29]

**Zadatak 2.3.30** Niz prirodnih brojeva formira se prema sledećem pravilu:

$$a_{n+1} = \begin{cases} \frac{a_n}{2} & \text{ako je } a_n \text{ parno} \\ \frac{3 \cdot a_n + 1}{2} & \text{ako je } a_n \text{ neparno} \end{cases}$$

Napisati program koji za uneti početni član niza  $a_0$  (pozitivan ceo broj) štampa niz brojeva sve do onog člana niza koji je jednak 1. U slučaju neispravnog unosa, ispisati odgovarajuću poruku o grešci.

*Primer 1*

```

INTERAKCIJA SA PROGRAMOM:
Unesite prvi clan: 56
56 28 14 7 11 17 26 13 20 10
5 8 4 2 1

```

*Primer 2*

```

INTERAKCIJA SA PROGRAMOM:
Unesite prvi clan: -48
Greska: neispravan unos.

```

### Primer 3

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite prvi clan: 67
|| 67 101 152 76 38 19 29 44 22 11
|| 17 26 13 20 10 5 8 4 2 1
```

### Primer 4

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite prvi clan: 33
|| 33 50 25 38 19 29 44 22
|| 11 17 26 13 20 10 5 8 4 2 1
```

[Rešenje 2.3.30]

**\* Zadatak 2.3.31** Papir  $A_0$  ima površinu  $1m^2$  i odnos stranica  $1 : \sqrt{2}$ . Papir  $A_1$  dobija se podelom papira  $A_0$  po dužoj ivici. Papir  $A_2$  dobija se podelom  $A_1$  papira po dužoj ivici itd. Napisati program koji za uneti nenegativan broj  $k$  ispisuje dimenzije papira  $A_k$  u milimetrima. Rezultat ispisati kao celobrojne vrednosti. U slučaju neispravnog unosa, ispisati odgovarajuću poruku o grešci.

### Primer 1

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite format papira: 4
|| 210 297
```

### Primer 2

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite format papira: 0
|| 840 1189
```

### Primer 3

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite format papira: -7
|| Greska: neispravan unos.
```

### Primer 4

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite format papira: 9
|| 37 52
```

[Rešenje 2.3.31]

**Zadatak 2.3.32** Napisati program koji učitava karaktere dok se ne unese karakter tačka, i ako je karakter malo slovo ispisuje odgovarajuće veliko, ako je karakter veliko slovo ispisuje odgovarajuće malo, a u suprotnom ispisuje isti karakter kao i uneti.

### Primer 1

```
|| INTERAKCIJA SA PROGRAMOM:
|| Danas je Veoma Lep DAN.
|| dANAS JE vEOMA lEP dan
```

### Primer 2

```
|| INTERAKCIJA SA PROGRAMOM:
|| PROGRAMIRANJE 1 je zanimljivo!.
|| programiranje 1 JE ZANIMLJIVO!
```

[Rešenje 2.3.32]

**Zadatak 2.3.33** Napisati program koji učitava karaktere sve do kraja ulaza, a potom ispisuje broj velikih slova, broj malih slova, broj cifara, broj belina i zbir unetih cifara.

*Primer 1*

```

INTERAKCIJA SA PROGRAMOM:
Tekst sa brojevima: 124, -8900, 23...
velika: 1, mala: 15
cifre: 9, beline: 5
suma cifara: 29

```

*Primer 2*

```

INTERAKCIJA SA PROGRAMOM:
NEMA cifara!
velika: 4, mala: 6
cifre: 0, beline: 1
suma cifara: 0

```

[Rešenje 2.3.33]

**Zadatak 2.3.34** Program učitava pozitivan ceo broj  $n$ , a potom i  $n$  karaktera. Za svaki od samoglasnika ispisati koliko puta se pojavio među unetim karakterima. Ne praviti razliku između malih i velikih slova. U slučaju neispravnog unosa, ispisati odgovarajuću poruku o grešci.

*Primer 1*

```

INTERAKCIJA SA PROGRAMOM:
Unesite broj n: 5
Unesite n karaktera: uAbao
Samoglasnik a: 2
Samoglasnik e: 0
Samoglasnik i: 0
Samoglasnik o: 1
Samoglasnik u: 1

```

*Primer 2*

```

INTERAKCIJA SA PROGRAMOM:
Unesite broj n: 7
Unesite n karaktera: jk+EEae
Samoglasnik a: 1
Samoglasnik e: 3
Samoglasnik i: 0
Samoglasnik o: 0
Samoglasnik u: 0

```

*Primer 3*

```

INTERAKCIJA SA PROGRAMOM:
Unesite broj n: 5
Unesite n karaktera: UuUuU
Samoglasnik a: 0
Samoglasnik e: 0
Samoglasnik i: 0
Samoglasnik o: 0
Samoglasnik u: 5

```

*Primer 4*

```

INTERAKCIJA SA PROGRAMOM:
Unesite broj n: -7
Greska: neispravan unos.

```

[Rešenje 2.3.34]

**Zadatak 2.3.35** Program učitava pozitivan ceo broj  $n$ , a zatim i  $n$  karaktera. Napisati program koji proverava da li se od unetih karaktera može napisati reč *Zima*. U slučaju neispravnog unosa, ispisati odgovarajuću poruku o grešci.

### Primer 1

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite broj n: 4
|| Unestite 1. karakter: +
|| Unestite 2. karakter: o
|| Unestite 3. karakter: Z
|| Unestite 4. karakter: j
|| Ne moze se napisati rec Zima.
```

### Primer 2

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite broj n: 10
|| Unestite 1. karakter: i
|| Unestite 2. karakter: 9
|| Unestite 3. karakter: 0
|| Unestite 4. karakter: p
|| Unestite 5. karakter: a
|| Unestite 6. karakter: Z
|| Unestite 7. karakter: o
|| Unestite 8. karakter: m
|| Unestite 9. karakter: M
|| Unestite 10. karakter: -
|| Moze se napisati rec Zima.
```

### Primer 3

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite broj n: 0
|| Greska: neispravan unos.
```

[Rešenje 2.3.35]

**Zadatak 2.3.36** Napisati program koji učitava pozitivan ceo broj  $n$  i ispisuje vrednost sume kubova brojeva od 1 do  $n$ , odnosno  $s = 1 + 2^3 + 3^3 + \dots + n^3$ . U slučaju neispravnog unosa, ispisati odgovarajuću poruku o grešci.

### Primer 1

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite broj n: 14
|| Suma kubova: 11025
```

### Primer 2

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite broj n: 25
|| Suma kubova: 105625
```

### Primer 3

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite broj n: 0
|| Greska: neispravan unos.
```

### Primer 4

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite broj n: -5
|| Greska: neispravan unos.
```

[Rešenje 2.3.36]

**Zadatak 2.3.37** Napisati program koji učitava pozitivan ceo broj  $n$  i ispisuje sumu kubova,  $s = 1 + 2^3 + 3^3 + \dots + k^3$ , za svaku vrednost  $k = 1, \dots, n$ . U slučaju neispravnog unosa, ispisati odgovarajuću poruku o grešci.

*Primer 1*

```

|| INTERAKCIJA SA PROGRAMOM:
|| Unesite broj n: 5
|| k=1, suma=1
|| k=2, suma=9
|| k=3, suma=36
|| k=4, suma=100
|| k=5, suma=225

```

*Primer 2*

```

|| INTERAKCIJA SA PROGRAMOM:
|| Unesite broj n: 8
|| k=1, suma=1
|| k=2, suma=9
|| k=3, suma=36
|| k=4, suma=100
|| k=5, suma=225
|| k=6, suma=441
|| k=7, suma=784
|| k=8, suma=1296

```

*Primer 3*

```

|| INTERAKCIJA SA PROGRAMOM:
|| Unesite broj n: 0
|| Greska: neispravan unos.

```

*Primer 4*

```

|| INTERAKCIJA SA PROGRAMOM:
|| Unesite broj n: -5
|| Greska: neispravan unos.

```

[Rešenje 2.3.37]

**Zadatak 2.3.38** Napisati program koji učitava realan broj  $x$  i pozitivan ceo broj  $n$  i izračunava i ispisuje sumu  $S = x + 2 \cdot x^2 + 3 \cdot x^3 + \dots + n \cdot x^n$ . U slučaju neispravnog unosa, ispisati odgovarajuću poruku o grešci.

*Primer 1*

```

|| INTERAKCIJA SA PROGRAMOM:
|| Unesite redom brojeve x i n: 2 3
|| S=34.000000

```

*Primer 2*

```

|| INTERAKCIJA SA PROGRAMOM:
|| Unesite redom brojeve x i n: 1.5 5
|| S=74.343750

```

*Primer 3*

```

|| INTERAKCIJA SA PROGRAMOM:
|| Unesite redom brojeve x i n: 5.5 0
|| Greska: neispravan unos.

```

*Primer 4*

```

|| INTERAKCIJA SA PROGRAMOM:
|| Unesite redom brojeve x i n: -0.5 -5
|| Greska: neispravan unos.

```

[Rešenje 2.3.38]

**Zadatak 2.3.39** Napisati program koji učitava realan broj  $x$  i pozitivan ceo broj  $n$  i izračunava i ispisuje sumu  $S = 1 + \frac{1}{x} + \frac{1}{x^2} + \dots + \frac{1}{x^n}$ . U slučaju neispravnog unosa, ispisati odgovarajuću poruku o grešci.

*Primer 1*

```

|| INTERAKCIJA SA PROGRAMOM:
|| Unesite redom brojeve x i n: 2 4
|| S=1.937500

```

*Primer 2*

```

|| INTERAKCIJA SA PROGRAMOM:
|| Unesite redom brojeve x i n: 1.8 6
|| S=2.213249

```

### Primer 3

```
INTERAKCIJA SA PROGRAMOM:
Unesite redom brojeve x i n: 5.5 0
Greska: neispravan unos.
```

### Primer 4

```
INTERAKCIJA SA PROGRAMOM:
Unesite redom brojeve x i n: -0.5 -5
Greska: neispravan unos.
```

[Rešenje 2.3.39]

**\* Zadatak 2.3.40** Napisati program koji učitava realne brojeve  $x$  i  $eps$  i sa zadatom tačnošću  $eps$  izračunava i ispisuje sumu  $S = 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \dots$ . Izračunati sumu u odnosu na tačnost  $eps$  znači uporediti poslednji član sume sa  $eps$  i ukoliko je taj poslednji član manji od  $eps$  prekinuti dalja izračunavanja. UPUTSTVO: Prilikom računanja sume koristiti prethodni izračunati član sume u računanju sledećeg člana sume. Naime, ako je izračunat član sume  $\frac{x^n}{n!}$  na osnovu njega se lako može dobiti član  $\frac{x^{n+1}}{(n+1)!}$ . Nikako ne računati stepen i faktorijel odvojeno zbog neefikasnosti takvog rešenja i zbog mogućnosti prekoračenja.

### Primer 1

```
INTERAKCIJA SA PROGRAMOM:
Unesite x: 2
Unesite tacnost eps: 0.001
S=7.388713
```

### Primer 2

```
INTERAKCIJA SA PROGRAMOM:
Unesite x: 3
Unesite tacnost eps: 0.01
S=20.079666
```

[Rešenje 2.3.40]

**\* Zadatak 2.3.41** Napisati program koji učitava realne brojeve  $x$  i  $eps$  i sa zadatom tačnošću  $eps$  izračunava i ispisuje sumu  $S = 1 - x + \frac{x^2}{2!} - \frac{x^3}{3!} + \frac{x^4}{4!} - \frac{x^5}{5!} + \dots$ . NAPOMENA: Voditi računa o efikasnosti rešenja i o mogućnosti prekoračenja.

### Primer 1

```
INTERAKCIJA SA PROGRAMOM:
Unesite x: 3
Unesite tacnost eps: 0.000001
S=0.049787
```

### Primer 2

```
INTERAKCIJA SA PROGRAMOM:
Unesite x: 3.14
Unesite tacnost eps: 0.01
S=0.049072
```

[Rešenje 2.3.41]

**Zadatak 2.3.42** Napisati program koji učitava realan broj  $x$  i pozitivan ceo broj  $n$  i izračunava proizvod  $P = (1 + \cos(x)) \cdot (1 + \cos(x^2)) \cdot \dots \cdot (1 + \cos(x^n))$ . U

slučaju neispravnog unosa, ispisati odgovarajuću poruku o grešci. NAPOMENA: *Voditi računa o efikasnosti rešenja.*

*Primer 1*

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite redom brojeve x i n: 3.4 5
|| P = 0.026817
```

*Primer 2*

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite redom brojeve x i n: 12 8
|| P = 2.640565
```

*Primer 3*

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite redom brojeve x i n: 12 0
|| Greska: neispravan unos.
```

*Primer 4*

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite redom brojeve x i n: 12 -6
|| Greska: neispravan unos.
```

[Rešenje 2.3.42]

\* **Zadatak 2.3.43** Napisati program koji učitava pozitivan ceo broj  $n$  i ispisuje vrednost razlomka

$$\frac{1}{1 + \frac{1}{2 + \frac{1}{3 + \frac{1}{4 + \frac{1}{\dots + \frac{1}{(n-1) + \frac{1}{n}}}}}}}}.$$

U slučaju neispravnog unosa, ispisati odgovarajuću poruku o grešci.

*Primer 1*

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite broj n: 4
|| R = 0.697674
```

*Primer 2*

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite broj n: 20
|| R = 0.697775
```

*Primer 3*

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite broj n: 0
|| Greska: neispravan unos.
```

[Rešenje 2.3.43]

\* **Zadatak 2.3.44** Napisati program koji učitava realan broj  $x$  i pozitivan ceo broj  $n$  i računa sumu

$$1 - \frac{x^2}{2!} + \frac{x^4}{4!} - \dots + (-1)^n \frac{x^{2n}}{(2n)!}.$$

U slučaju neispravnog unosa, ispisati odgovarajuću poruku o grešci. NAPOMENA: *Voditi računa o efikasnosti rešenja i o mogućnosti prekoračenja.*

### Primer 1

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite x i n: 5.6 8
|| S=0.779792
```

### Primer 2

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite x i n: 14.32 11
|| S=-6714.066406
```

### Primer 3

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite x i n: 2 -6
|| Greska: neispravan unos.
```

[Rešenje 2.3.44]

\* **Zadatak 2.3.45** Napisati program koji učitava pozitivan ceo broj  $n$  i koji računa proizvod

$$S = (1 + \frac{1}{2!})(1 + \frac{1}{3!}) \dots (1 + \frac{1}{n!}).$$

U slučaju neispravnog unosa, ispisati odgovarajuću poruku o grešci. NAPOMENA: Voditi računa o efikasnosti rešenja i o mogućnosti prekoračenja.

### Primer 1

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite broj n: 5
|| P = 1.838108
```

### Primer 2

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite broj n: 7
|| P = 1.841026
```

### Primer 3

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite broj n: 0
|| Greska: neispravan unos.
```

### Primer 4

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite broj n: 10
|| P = 1.841077
```

[Rešenje 2.3.45]

\* **Zadatak 2.3.46** Napisati program koji učitava neparan ceo broj  $n$  ( $n \geq 5$ ) i izračunava i ispisuje sumu

$$S = 1 \cdot 3 \cdot 5 - 1 \cdot 3 \cdot 5 \cdot 7 + 1 \cdot 3 \cdot 5 \cdot 7 \cdot 9 - 1 \cdot 3 \cdot 5 \cdot 7 \cdot 9 \cdot 11 + \dots (-1)^{\frac{n-1}{2}+1} \cdot 1 \cdot 3 \cdot \dots \cdot n.$$

U slučaju greške pri unosu podataka ispisati odgovarajuću poruku. NAPOMENA: Voditi računa o efikasnosti rešenja i o mogućnosti prekoračenja.

### Primer 1

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite broj n: 9
|| 855
```

### Primer 2

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite broj n: 11
|| -9540
```

### Primer 3

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite broj n: 20
|| Greska: neispravan unos.
```

### Primer 4

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite broj n: -3
|| Greska: neispravan unos.
```



[Rešenje 2.3.46]

**Zadatak 2.3.47** Napisati program koji učitava realne brojeve  $x$  i  $a$  i pozitivan ceo broj  $n$  i zatim izračunava i ispisuje vrednost izraza

$$\underbrace{((\dots((x+a)^2+a)^2+a)^2+\dots a)^2}_n.$$

U slučaju neispravnog unosa, ispisati odgovarajuću poruku o grešci.

*Primer 1*

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite brojeve x i a: 3.2 0.2
|| Unesite broj n: 5
|| Izraz = 135380494030332048.000000
```

*Primer 2*

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite brojeve x i a: 2 1
|| Unesite broj n: 3
|| Izraz = 10201.000000
```

*Primer 3*

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite brojeve x i a: 2.6 0.3
|| Unesite broj n: 3
|| Izraz = 5800.970129
```

*Primer 4*

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite brojeve x i a: 5.4 7
|| Unesite broj n: -2
|| Greska: neispravan unos.
```

[Rešenje 2.3.47]

**Zadatak 2.3.48** Napisati programe koji za unetu pozitivnu celobrojnu vrednost  $n$  ispisuju tražene tablice. NAPOMENA: *Pretpostaviti da je unos ispravan.*

- (a) Napisati program koji za unetu vrednost  $n$  ispisuje tablicu množenja.

*Primer 1*

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite broj n: 1
|| 1
```

*Primer 2*

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite broj n: 2
|| 1 2
|| 2 4
```

*Primer 3*

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite broj n: 3
|| 1 2 3
|| 2 4 6
|| 3 6 9
```

*Primer 4*

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite broj n: 4
|| 1 2 3 4
|| 2 4 6 8
|| 3 6 9 12
|| 4 8 12 16
```

- (b) Napisati program koji za unetu  $n$  ispisuje sve brojeve od 1 do  $n^2$  pri čemu se ispisuje po  $n$  brojeva u jednoj vrsti.

### Primer 1

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite broj n: 3
|| 1 2 3
|| 4 5 6
|| 7 8 9
```

### Primer 2

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite broj n: 4
|| 1 2 3 4
|| 5 6 7 8
|| 9 10 11 12
|| 13 14 15 16
```

- (c) Napisati program koji za uneto  $n$  ispisuje tablicu brojeva tako da su u prvoj vrsti svi brojevi od 1 do  $n$ , a svaka naredna vrsta dobija se rotiranjem prethodne vrste za jedno mesto u levo.

### Primer 1

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite broj n: 3
|| 1 2 3
|| 2 3 1
|| 3 1 2
```

### Primer 2

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite broj n: 4
|| 1 2 3 4
|| 2 3 4 1
|| 3 4 1 2
|| 4 1 2 3
```

- (d) Napisati program koji za uneto  $n$  iscrtava pravougli „trougao” sačinjen od „koordinata” svojih tačaka. „Koordinata” tačke je oblika  $(i, j)$  pri čemu  $i, j = 0, \dots, n$ . Prav ugao se nalazi u gornjem levom uglu slike i njegova koordinata je  $(0, 0)$ . Koordinata  $i$  se uvećava po vrsti, a koordinata  $j$  po koloni, pa je zato koordinata tačke koja je ispod tačke  $(0, 0)$  jednaka  $(1, 0)$ , a koordinata tačke koja je desno od tačke  $(0, 0)$  jednaka  $(0, 1)$ .

### Primer 1

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite broj n: 1
|| (0,0)
```

### Primer 2

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite broj n: 2
|| (0,0) (0,1)
|| (1,0)
```

### Primer 3

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite broj n: 3
|| (0,0) (0,1) (0,2)
|| (1,0) (1,1)
|| (2,0)
```

### Primer 4

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite broj n: 4
|| (0,0) (0,1) (0,2) (0,3)
|| (1,0) (1,1) (1,2)
|| (2,0) (2,1)
|| (3,0)
```

[Rešenje 2.3.48]

**Zadatak 2.3.49** Napisati program koji za uneti pozitivan ceo broj  $n$  zvezdicama iscrtava odgovarajuću sliku. NAPOMENA: *Pretpostaviti da je unos ispravan.*

- (a) Slika predstavlja kvadrat stranice  $n$  sastavljen od zvezdica.

Primer 1

```
INTERAKCIJA SA PROGRAMOM:
Unesite broj n: 3
***
***
***
```

Primer 2

```
INTERAKCIJA SA PROGRAMOM:
Unesite broj n: 4
****
****
****
****
```

- (b) Slika predstavlja rub kvadrata dimenzije  $n$ .

Primer 1

```
INTERAKCIJA SA PROGRAMOM:
Unesite broj n: 5
*****
*   *
*   *
*   *
*   *
*****
```

Primer 2

```
INTERAKCIJA SA PROGRAMOM:
Unesite broj n: 2
**
**
```

- (c) Slika predstavlja rub kvadrata dimenzije  $n$  koji i na glavnoj dijagonali ima zvezdice.

Primer 1

```
INTERAKCIJA SA PROGRAMOM:
Unesite broj n: 5
*****
** *
* * *
* **
*****
```

Primer 1

```
INTERAKCIJA SA PROGRAMOM:
Unesite broj n: 4
****
** *
* **
****
```

[Rešenje 2.3.49]

\* **Zadatak 2.3.50** Napisati program koji za uneti pozitivan ceo broj  $n$  zvezdicama iscrtava slovo  $X$  dimenzije  $n$ . NAPOMENA: *Pretpostaviti da je unos ispravan.*

### Primer 1

```
INTERAKCIJA SA PROGRAMOM:
Unesite broj n: 5
* *
* *
*
* *
* *
```

### Primer 2

```
INTERAKCIJA SA PROGRAMOM:
Unesite broj n: 3
* *
*
* *
```

[Rešenje 2.3.50]

**\* Zadatak 2.3.51** Napisati program koji za uneti neparan pozitivan broj  $n$  korišćenjem znaka + iscrtava veliko + dimenzije  $n$ . NAPOMENA: *Pretpostaviti da je unos ispravan.*

### Primer 1

```
INTERAKCIJA SA PROGRAMOM:
Unesite broj n: 5
+
+
+++++
+
+
```

### Primer 2

```
INTERAKCIJA SA PROGRAMOM:
Unesite broj n: 3
+
+++
+
```

[Rešenje 2.3.51]

**Zadatak 2.3.52** Napisati program koji učitava pozitivan ceo broj  $n$ , a potom iscrtava odgovarajuću sliku. NAPOMENA: *Pretpostaviti da je unos ispravan.*

- (a) Slika predstavlja pravougli trougao sastavljen od zvezdica. Kateta trougla je dužine  $n$ , a prav ugao se nalazi u gornjem levom uglu slike.

### Primer 1

```
INTERAKCIJA SA PROGRAMOM:
Unesite broj n: 3
***
**
*
```

### Primer 1

```
INTERAKCIJA SA PROGRAMOM:
Unesite broj n: 4
****
***
**
*
```

- (b) Slika predstavlja pravougli trougao sastavljen od zvezdica. Kateta trougla je dužine  $n$ , a prav ugao se nalazi u donjem levom uglu slike.

### Primer 1

```
INTERAKCIJA SA PROGRAMOM:
Unesite broj n: 3
*
**
***
```

### Primer 2

```
INTERAKCIJA SA PROGRAMOM:
Unesite broj n: 4
*
**
***
****
```

- (c) Slika predstavlja pravougli trougao sastavljen od zvezdica. Kateta trougla je dužine  $n$ , a prav ugao se nalazi u gornjem desnom uglu slike.

### Primer 1

```
INTERAKCIJA SA PROGRAMOM:
Unesite broj n: 3
***
**
*
```

### Primer 1

```
INTERAKCIJA SA PROGRAMOM:
Unesite broj n: 4
****
***
**
*
```

- (d) Slika predstavlja pravougli trougao sastavljen od zvezdica. Kateta trougla je dužine  $n$ , a prav ugao se nalazi u donjem desnom uglu slike.

### Primer 1

```
INTERAKCIJA SA PROGRAMOM:
Unesite broj n: 3
*
**
***
```

### Primer 2

```
INTERAKCIJA SA PROGRAMOM:
Unesite broj n: 4
*
**
***
****
```

- (e) Slika predstavlja trougao sastavljen od zvezdica. Trougao se dobija spajanjem dva pravougla trougla čija kateta je dužine  $n$ , pri čemu je prav ugao prvog trougla u njegovom donjem levom uglu, dok je prav ugao drugog trougla u njegovom gornjem levom uglu, a spajanje se vrši po horizontalnoj kateti.

### Primer 1

```
INTERAKCIJA SA PROGRAMOM:
Unesite broj n: 3
*
**
***
**
*
```

### Primer 2

```
INTERAKCIJA SA PROGRAMOM:
Unesite broj n: 4
*
**
***
****
***
**
*
```

- (f) Slika predstavlja rub jednakokrakog pravouglog trougla čije su katete dužine  $n$ . Program učitava karakter  $c$  i taj karakter koristi za iscrtavanje ruba trougla.

*Primer 1*

```
INTERAKCIJA SA PROGRAMOM:
Unesite broj n: 4
Unesite karakter c: *
*
**
* *
****
```

*Primer 2*

```
INTERAKCIJA SA PROGRAMOM:
Unesite broj n: 5
Unesite karakter c: +
+
++
+ +
+ +
++++
```

[Rešenje 2.3.52]

**Zadatak 2.3.53** Napisati program koji učitava pozitivan ceo broj  $n$ , a potom iscrtava odgovarajuću sliku. NAPOMENA: *Pretpostaviti da je unos ispravan.*

- (a) Slika predstavlja jednakostranični trougao stranice  $n$  koji je sastavljen od zvezdica.

*Primer 1*

```
INTERAKCIJA SA PROGRAMOM:
Unesite broj n: 3
*
**
***
```

*Primer 2*

```
INTERAKCIJA SA PROGRAMOM:
Unesite broj n: 4
*
**
***
****
```

- (b) Slika predstavlja jednakostranični trougao stranice  $n$  koji je sastavljen od zvezdica pri čemu je vrh trougla na dnu slike.

*Primer 1*

```
INTERAKCIJA SA PROGRAMOM:
Unesite broj n: 3
****
***
**
*
```

*Primer 2*

```
INTERAKCIJA SA PROGRAMOM:
Unesite broj n: 4
*****
****
***
**
*
```

- (c) Slika predstavlja trougao koji se dobija spajanjem dva jednakostranična trougla stranice  $n$  koji su sastavljeni od zvezdica.

Primer 1

```
INTERAKCIJA SA PROGRAMOM:
Unesite broj n: 3
*
***
*****
***
*
```

Primer 2

```
INTERAKCIJA SA PROGRAMOM:
Unesite broj n: 5
*
***
*****
*****
*****
*****
***
*
```

- (d) Slika predstavlja rub jednakostraničnog trougla čija stranica je dužine  $n$ .

Primer 1

```
INTERAKCIJA SA PROGRAMOM:
Unesite broj n: 3
*
* *
* * *
```

Primer 1

```
INTERAKCIJA SA PROGRAMOM:
Unesite broj n: 5
*
* *
* * *
*   *
* * * *
* * * * *
```

- (e) Slika se dobija spajanjem dva jednakostranična trougla čija stranica je dužine  $n$ . Iscrtavati samo rub trouglova.

Primer 1

```
INTERAKCIJA SA PROGRAMOM:
Unesite broj n: 3
*
* *
* * *
* *
*
```

Primer 2

```
INTERAKCIJA SA PROGRAMOM:
Unesite broj n: 5
*
* *
* * *
*   *
* * * *
* * * * *
*   *
* *
* *
*
```

\* **Zadatak 2.3.54** Napisati program koji za uneti pozitivan ceo broj  $n$  iscrtava strelice dimenzije  $n$ . NAPOMENA: *Pretpostaviti da je unos ispravan.*

### Primer 1

```
INTERAKCIJA SA PROGRAMOM:
Unesite broj n: 3
*
 *
***
 *
 *
```

### Primer 2

```
INTERAKCIJA SA PROGRAMOM:
Unesite broj n: 5
*
 *
 *
 *
*****
 *
 *
 *
 *
```

[Rešenje 2.3.54]

\* **Zadatak 2.3.55** Napisati program koji učitava pozitivan ceo broj  $n$ , i iscrta sliku koja se dobija na sledeći način: u prvom redu je jedna zvezdica, u drugom redu su dve zvezdice razdvojene razmakom, treći red je sastavljen od zvezdica i iste je dužine kao i drugi red, četvrti red se sastoji od tri zvezdice razdvojene razmakom, a peti red je sastavljen od zvezdica i iste je dužine kao i četvrti red itd. Ukupna visina slike je  $n$ . NAPOMENA: *Pretpostaviti da je unos ispravan.*

### Primer 1

```
INTERAKCIJA SA PROGRAMOM:
Unesite broj n: 7
*
* *
***
* * *
*****
* * * *
*****
```

[Rešenje 2.3.55]

\* **Zadatak 2.3.56** Napisati program koji učitava pozitivne cele brojeve  $m$  i  $n$  i iscrta jedan do drugog  $n$  kvadrata čija je svaka strana sastavljena od  $m$  zvezdica razdvojenih prazninom. NAPOMENA: *Pretpostaviti da je unos ispravan.*



## Primer 1

```

INTERAKCIJA SA PROGRAMOM:
Unesite brojeve n i m: 5 3
*****
*       *       *       *
*       *       *       *
*       *       *       *
*       *       *       *
*****

```

## Primer 2

```

INTERAKCIJA SA PROGRAMOM:
Unesite brojeve n i m: 4 4
*****
*       *       *       *
*       *       *       *
*       *       *       *
*       *       *       *
*****

```

[Rešenje 2.3.56]

\* **Zadatak 2.3.57** Napisati program koji učitava pozitivan ceo broj  $n$  i štampa romb sastavljen od minusa u pravougaoniku sastavljenom od zvezdica. NAPOMENA: *Pretpostaviti da je unos ispravan.*

## Primer 1

```

INTERAKCIJA SA PROGRAMOM:
Unesite broj n: 6
*****
****--****
****--****
***-----***
**-----**
*-----*
*-----*
***-----***
****--****
****--****
*****

```

## Primer 2

```

INTERAKCIJA SA PROGRAMOM:
Unesite broj n: 2
****
*--*
****

```

[Rešenje 2.3.57]

**Zadatak 2.3.58** Napisati program koji učitava ceo broj  $n$  ( $n \geq 2$ ) i koji iscrtava sliku kuće sa krovom: kuća je kvadrat stranice  $n$ , a krov jednakostranični trougao stranice  $n$ . Pretpostaviti da je unos korektan.

## Primer 1

```

INTERAKCIJA SA PROGRAMOM:
Unesite broj n: 4
  *
 * *
* * *
* * * *
 * *
  *
* * *

```

## Primer 1

```

INTERAKCIJA SA PROGRAMOM:
Unesite broj n: 3
  *
 * *
* * *
 * *
  *

```

[Rešenje 2.3.58]

\* **Zadatak 2.3.59** Napisati program koji učitava pozitivan ceo broj  $n$  i ispisuje brojeve od 1 do  $n$ , zatim od 2 do  $n - 1$ , 3 do  $n - 2$ , itd. Ispis se završava kada nije moguće ispisati ni jedan broj. NAPOMENA: *Pretpostaviti da je unos ispravan.*

*Primer 1*

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesite broj n: 5  
|| 1 2 3 4 5 2 3 4 3
```

*Primer 2*

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesite broj n: 6  
|| 1 2 3 4 5 6 2 3 4 5 3 4
```

*Primer 3*

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesite broj n: 7  
|| 1 2 3 4 5 6 7 2 3 4 5 6 3 4 5 4
```

*Primer 4*

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesite broj n: 3  
|| 1 2 3 2
```

[Rešenje 2.3.59]

\* **Zadatak 2.3.60** Napisati program koji učitava pozitivan ceo broj  $n$  i ispisuje sve brojeve od 1 do  $n$ , zatim svaki drugi broj od 1 do  $n$ , zatim svaki treći broj od 1 do  $n$  itd., završavajući sa svakim  $n$ -tim (tj. samo sa 1). NAPOMENA: *Pretpostaviti da je unos ispravan.*

*Primer 1*

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesite broj n: 3  
|| 1 2 3  
|| 1 3  
|| 1
```

*Primer 2*

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesite broj n: 7  
|| 1 2 3 4 5 6 7  
|| 1 3 5 7  
|| 1 4 7  
|| 1 5  
|| 1 6  
|| 1 7  
|| 1
```

*Primer 3*

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesite broj n: 1  
|| 1
```

[Rešenje 2.3.60]

## 2.4 Rešenja

### Rešenje 2.3.1

```
1 #include <stdio.h>
2
3 int main()
4 {
5     /* Deklaracija potrebnih promenljivih. */
6     int i;
7
8     /* Promenljiva i kontrolise koliko puta ce se petlja izvesti i
9        naziva se brojac petlje. Njena pocetna vrednost se postavlja na
10        0 jer se u pocetku petlja nije ni jednom izvela. */
11     i = 0;
12
13     /* Petlja ce se izvesti za i=0,1,2,3,4. Kada i dostigne vrednost
14        5 uslov i < 5 nece biti ispunjen i prelazi se na prvu sledecu
15        naredbu nakon tela petlje. */
16     while (i < 5) {
17
18         /* Ispis poruke. */
19         printf("Mi volimo da programiramo.\n");
20
21         /* Uvecavanje brojaca za 1. */
22         i++;
23     }
24
25     return 0;
26 }
```

### Rešenje 2.3.2

```
1 #include<stdio.h>
2
3 int main()
4 {
5     /* Deklaracija potrebnih promenljivih. */
6     int i, n;
7
8     /* Ucitava se vrednost broja n. */
9     printf("Unesite broj n: ");
10    scanf("%d", &n);
11
12    /* Vrsi se provera ispravnosti ulaza. */
13    if (n <= 0) {
14        printf("Greska: pogresan unos broja n.\n");
15        return -1;
16    }
17 }
```

## 2 Kontrola toka

---

```
17  /* Inicijalizacija brojaca. */
19  i = 0;

21  /* Trazena poruka se ispisuje n puta. */
22  while (i < n) {
23      printf("Mi volimo da programiramo.\n");
24      i++;
25  }

27  return 0;
}
```

### Rešenje 2.3.3

```
1  #include <stdio.h>

3  int main()
4  {
5      /* Deklaracija potrebnih promenljivih. */
6      int i, n;

7      /* Ucitava se vrednost broja n. */
8      printf("Unesite broj n: ");
9      scanf("%d", &n);

11     /* Vrsi se provera ispravnosti ulaza. */
12     if (n < 0) {
13         printf("Greska: pogresan unos broja n.\n");
14         return -1;
15     }

17     /* Inicijalizacija brojaca. */
18     i = 0;

21     /* Posto je potrebno ispisati sve brojeve [0,n], telo petlje
22     se izvrsava za svako i <= n. */
23     while (i <= n) {

25         /* Ispisuje se trenutna vrednost brojaca. */
26         printf("%d\n", i);

27         /* Prelazi se na sledeci broj. */
28         i++;
29     }

31     return 0;
33 }
```

## Rešenje 2.3.4

```
1  #include <stdio.h>
3  int main()
4  {
5      /* Deklaracija potrebnih promenljivih. */
6      int n, m, i;
7
8      /* Ucitavaju se vrednosti granica intervala. */
9      printf("Unesite granice intervala: ");
10     scanf("%d%d", &n, &m);
11
12     /* Vrsi se provera ispravnosti ulaznih podataka. */
13     if (m < n) {
14         printf("Greska: pogresan unos granica.\n");
15         return -1;
16     }
17
18     /* a) I nacin: koriscenjem while petlje. */
19     /* Inicijalizacija brojaca na levu granicu intervala. */
20     i = n;
21
22     /* Ispisuju se sve vrednosti brojaca izmedju leve i desne
23     granice intervala, ukljucujuci i same granice. */
24     while (i <= m) {
25         printf("%d ", i);
26         i++;
27     }
28
29     /* b) II nacin: koriscenjem for petlje.
30
31     Naredba i=n se izvorsava jednom, pre prve iteracije.
32     Uslov petlje i<=m se proverava pre svake iteracije.
33     Naredba i++ se izvorsava nakon svake iteracije.
34
35     for (i = n; i <= m; i++){
36         printf("%d ", i);
37     } */
38
39     /* c) III nacin: koriscenjem do while petlje.
40
41     Uslov petlje se proverava na kraju svake iteracije.
42     Zbog toga se do while petlja izvorsava bar jednom, cak i u
43     slucaju da uslov petlje nikada nije ispunjen. U ovom slucaju
44     je to ispravno jer je poznato da ce interval imati bar
45     jedan element. U opstem slucaju to ne mora da vaziti.
46
47     i = n;
48     do {
49         printf("%d ", i);
50         i++;
```

```
51 }
    while (i <= m); */
53
    printf("\n");
55
    return 0;
57 }
```

### Rešenje 2.3.5

```
1  #include<stdio.h>
3  int main()
4  {
5      /* Deklaracija potrebnih promenljivih. */
6      int n, tekuca_vrednost;
7
8      /* Za cuvanje vrednosti faktoriijela se koristi tip unsigned long
9         jer izracunata vrednost moze da bude jako veliki broj. */
10     unsigned long faktoriijel;
11
12     /* Ucitava se vrednost broja n. */
13     printf("Unesite broj n: ");
14     scanf("%d", &n);
15
16     /* Vrsi se provera ispravnosti ulaza. */
17     if (n < 0) {
18         printf("Greska: neispravan unos..\n");
19         return -1;
20     }
21
22     if (n >= 22) {
23         printf("Pri racunanju %d! ce doci do prekoracenja.\n", n);
24         return -1;
25     }
26
27     /* Tekuca vrednost uzima vrednosti n, n-1, n-2, ..., 2.
28        Na pocetku se inicijalizuje na n, a zatim se u svakoj
29        iteraciji umanjuje za 1. */
30     tekuca_vrednost = n;
31
32     /* Inicijalizacija vrednosti faktoriijela. */
33     faktoriijel = 1;
34
35     /* Racuna se vrednost faktoriijela tako sto se trenutni rezultat
36        u svakoj iteraciji mnozi sa promenljivom cija vrednost krece
37        od n, a zatim se u svakoj iteraciji umanjuje za 1. */
38     while (tekuca_vrednost > 1) {
39         faktoriijel = faktoriijel * tekuca_vrednost;
40         tekuca_vrednost--;
41     }
```

```

43  /* Ispis rezultata. */
    printf("%d! = %lu\n", n, faktorijel);
45
    return 0;
47 }

```

### Rešenje 2.3.6

```

1  #include <stdio.h>

3  int main()
{
5  /* Deklaracije potrebnih promenljivih. */
    int n, i;
7  float x, rezultat;

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

13  /* Vrsi se provera ispravnosti ulaza. */
    if (n < 0) {
15      printf("Greska: neispravan unos broja n.\n");
        return -1;
17  }

19  /* Inicijalizacija rezultata. */
    rezultat = 1;

21  /* Vrednost n-tog stepena broja x se dobija tako sto se tekuca
23     vrednost rezultata n puta pomnozi sa brojem x.
        (rezultat = x * x * ... * x) = x^n */
25  for (i=0; i<n; i++)
        rezultat = rezultat * x;

27  /* Ispis rezultata. */
29  printf("Rezultat: : %.5f\n", rezultat);

31  return 0;
}

```

### Rešenje 2.3.7

```

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

3
    int main()
5  {

```

```
/* Deklaracije potrebnih promenljivih. */
7 int n, i, znak;
float x, rezultat;

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

13 /* Pamti se znak stepena i uzima se apsolutna vrednost stepena. */
15 znak = 1;
if(n < 0){
17     znak = -1;
    n = abs(n);
19 }
/* Inicijalizacija rezultata. */
21 rezultat = 1;

23 /* Racuna se vrednost x^n. */
for (i=0; i<n; i++)
25     rezultat = rezultat * x;

27 /* Ako je stepen bio negativan, rezultat je 1/x^n. */
if (znak == -1)
29     printf("Rezultat: %.3f\n", 1 / rezultat);
else
31     printf("Rezultat: %.3f\n", rezultat);

33 return 0;
}
```

### Rešenje 2.3.8

```
1 #include<stdio.h>

3 int main()
{
5     /* Deklaracija potrebnih promenljivih. */
    int n, i;

7     /* Ucitava se vrednost broja n. */
    printf("Unesite broj n: ");
9     scanf("%d", &n);

11     /* Vrsi se provera ispravnosti ulaza. */
    if (n <= 0) {
13         printf("Greska: neispravan unos.\n");
        return -1;
15     }

17     /* I nacin: Za svaki broj iz intervala [2, n-1] se proverava da
19     li deli broj n (tj. da li je ostatak pri deljenju sa n jednak
```



```

nuli). Ako je uslov ispunjen, taj broj se ispisuje.
21 for (i = 2; i < n; i++) {
    if (n % i == 0)
23     printf("%d ", i);
}
25 printf("\n");
*/

27 /* II nacin (brzi): Provera se ne vrši za sve brojeve iz
29 intervala [2, n-1], vec za brojeve iz intervala
    [2, sqrt(n)], tj. za sve brojeve k za koje vazi da je
31 k*k <= n. */
for (i = 2; i*i <= n; i++) {
33     /* Ako i deli n, treba razlikovati dva slucaja. */
    if (n % i == 0){
35         if (i == n / i) {
            /* I slucaj: kada je i koren broja, npr. 4 za 16,
37             ispisuje se samo broj i. */
            printf("%d ", i);
39         }
        else {
41             /* II slucaj: u suprotnom, ispisuje se taj broj i
                broj n / i, npr. 2 za 16, ispisuju se i 2 i 8. */
43             printf("%d %d ", i, n / i);
        }
45     }
}
47 printf("\n");

49 return 0;
}

```

### Rešenje 2.3.9

```

1  #include <stdio.h>

3  int main()
{
5     /* Deklaracija broja n. */
    int n;

7     /* Ucitava se vrednost broja n. */
    printf("Unesite broj: ");
    scanf("%d", &n);

11     /* Slucaj kada broj n ima vrednost nula se posebno obradjuje.
        Kada ovo ne bi bilo navedeno, petlja u nastavku bi se
        u ovom slucaju izvorsavala beskonacno. */
13     if (n == 0) {
        printf("0\n");
15         return 0;
17     }
}

```

```
19 }
20
21 /* Dok god je poslednja cifra broja n nula, broj n se deli sa
22    10 i na taj nacin se iz broja uklanja poslednja cifra. */
23 while (n % 10 == 0)
24     n = n / 10;
25
26 /* Ispis rezultata. */
27 printf("%d\n", n);
28
29 return 0;
30 }
```

### Rešenje 2.3.10

```
1  #include<stdio.h>
2  #include<stdlib.h>
3
4  int main()
5  {
6      /* Deklaracija potrebnih promenljivih. */
7      int x;
8
9      /* Ucitava se vrednost broja x. */
10     printf("Unesi ceo broj:");
11     scanf("%d", &x);
12
13     /* Uzima se apsolutna vrednost broja da bi izdvojene cifre bile
14        pozitivni brojevi. Na primer, 123%10 je 3, a -123%10 je -3. */
15     x = abs(x);
16
17     /* Slucaj kada je uneti broj 0 se posebno obradjuje. */
18     if(x == 0)
19     {
20         printf("0\n");
21         return 0;
22     }
23
24     /* U petlji se obradjuje cifra po cifra broja, dok god ima
25        neobradjenih cifara u broju. */
26     while (x != 0) {
27         /* Ispisuje se poslednja cifra broja x. */
28         printf("%d ", x % 10);
29
30         /* Uklanja se poslednja cifra broja x. */
31         x /= 10;
32     }
33     printf("\n");
34
35     return 0;
36 }
```

## Rešenje 2.3.11

```
1  #include <stdio.h>
3  int main()
4  {
5      /* Deklaracija potrebnih promenljivih. */
6      int n, suma, pom_n;
7
8      /* Ucitava se vrednost broja n. */
9      printf("Unesite broj: ");
10     scanf("%d", &n);
11
12     /* Vrsi se provera ispravnosti ulaza. */
13     if (n <= 0) {
14         printf("Greska: neispravan unos.\n");
15         return -1;
16     }
17
18     /* Pravi se kopija originalnog broja, da bi originalna vrednost
19        n ostala nepromenjena. */
20     pom_n = n;
21
22     /*Inicijalizacija sume cifara. */
23     suma = 0;
24
25     /* Racuna se suma cifara. */
26     while (pom_n != 0) {
27         /* Na sumu se dodaje poslednja cifra broja. */
28         suma += pom_n % 10;
29         /* Sa broja se skida poslednja cifra. */
30         pom_n /= 10;
31     }
32
33     /* Ispis rezultata. */
34     if (n % suma == 0)
35         printf("Broj %d je deljiv sa %d.\n", n, suma);
36     else
37         printf("Broj %d nije deljiv sa %d.\n", n, suma);
38
39     return 0;
40 }
```

## Rešenje 2.3.12

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

```
3  int main()
5  {
    /* Deklaracija potrebnih promenljivih. */
7   int t, x, i;
   int ukupan_prihod, ukupan_rashod, ukupan_rashod_abs;

9   /* Ucitava se vrednost broja t. */
11  printf("Unesite broj t:");
   scanf("%d", &t);

13  /* Vrsi se provera ispravnosti ulaza. */
15  if (t < 0) {
       printf("Greska: neispravan unos.\n");
17     return -1;
   }
19  else if( t == 0) {
       printf("Nema evidentiranih transakcija.");
21     return 0;
   }

23  /* Inicijalizacija suma. */
25  ukupan_prihod = 0;
   ukupan_rashod = 0;

27  /* Ucitavanje transakcija i izracunavanje suma. */
29  printf("Unesite transakcije: ");
   i = 0;
31  while (i < t) {
       /* Ucitava se jedna transakcija. */
33     scanf("%d", &x);

35     /* Dodaje se na odgovarajucu sumu. */
       if (x < 0)
37         ukupan_rashod += x;
       else
39         ukupan_prihod += x;

41     /* Uvecava se brojac. */
       i++;
43 }

45 /* Ispis rezultata. */
47 printf("Prihod: %d\n", ukupan_prihod);
   printf("Rashod: %d\n", ukupan_rashod);

49 ukupan_rashod_abs = abs(ukupan_rashod);
   if(ukupan_prihod >= ukupan_rashod_abs)
51     printf("Zarada: %d\n", ukupan_prihod - ukupan_rashod_abs);
   else
53     printf("Gubitak: %d\n", ukupan_rashod_abs - ukupan_prihod);
```

```
55 return 0;
}
```

### Rešenje 2.3.13

```
1  #include <stdio.h>
3  int main()
4  {
5      /* Deklaracija potrebnih promenljivih. */
6      int n, x, i;
7      int zbir = 0;
8
9      /* Ucitava se vrednost broja n. */
10     printf("Unesite broj n: ");
11     scanf("%d", &n);
12
13     /* Vrsi se provera ispravnosti ulaza. */
14     if (n <= 0) {
15         printf("Greska: neispravan unos.\n");
16         return -1;
17     }
18
19     /* Ucitava se n brojeva i izracunava se trazeni zbir. */
20     printf("Unesite n brojeva: ");
21     i = 0;
22     while (i < n) {
23         /* Ucitava se jedan broj. */
24         scanf("%d", &x);
25
26         /* Ako je ucitani broj negativan i neparan,
27            dodaje se na zbir. */
28         if (x < 0 && x % 2 != 0)
29             zbir = zbir + x;
30
31         /* Uvecava se brojac. */
32         i++;
33     }
34
35     /* Ispis rezultata. */
36     printf("Zbir neparnih i negativnih: %d\n", zbir);
37
38     return 0;
39 }
```

### Rešenje 2.3.14

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

```
3 int main()
4 {
5     /* Deklaracija potrebnih promenljivih. */
6     int n, broj;
7     int suma = 0;
8     int i;
9
10    /* Ucitava se vrednost broja n. */
11    printf("Unesite broj n: ");
12    scanf("%d", &n);
13
14    /* Vrsi se provera ispravnosti ulaza. */
15    if (n <= 0) {
16        printf("Greska: neispravan unos.\n");
17        return -1;
18    }
19
20    /* Ucitava se n brojeva i izracunava se trazena suma. */
21    printf("Unesite brojeve: ");
22    for (i = 0; i < n; i++) {
23        scanf("%d", &broj);
24
25        if (broj % 5 == 0 && broj % 7 != 0)
26            suma += broj;
27    }
28
29    /* Ispis rezultata. */
30    printf("Suma je %d.\n", suma);
31
32    return 0;
33 }
```

### Rešenje 2.3.15

```
1 #include <stdio.h>
2 int main()
3 {
4     /* Deklaracija potrebnih promenljivih. */
5     int x, proizvod;
6
7     /* Indikator koji oznacava da li je korisnik uneo bar jedan
8        broj. */
9     int unet_bar_jedan = 0;
10
11    /* Indikator koji oznacava da li je korisnik uneo bar jedan
12       pozitivan broj. */
13    int unet_pozitivan = 0;
14
15    /* Inicijalizacija proizvoda. */
16    proizvod = 1;
17 }
```

```

19 printf("Unesite brojeve:");
21 while (1) {
23     /* Ucitava se jedan broj. */
24     scanf("%d", &x);
25
26     /* Ako je uneta nula, petlja se prekida naredbom break. */
27     if (x == 0)
28         break;
29
30     /* Ako petlja nije prekinuta, znaci da je unet bar jedan broj.
31        Iz tog razloga se vrednost indikatora za unete brojeve
32        postavlja na 1. */
33     unet_bar_jedan = 1;
34
35     /* Proverava se da li je broj x pozitivan. */
36     if(x > 0){
37         /* Ako jeste, znaci da je unet bar jedan pozitivan broj i iz
38            tog razloga se vrednost odgovarajuceg indikatora postavlja
39            na 1. */
40         unet_pozitivan = 1;
41
42         /* Azurira se vrednost proizvoda pozitivnih brojeva. */
43         proizvod = proizvod * x;
44     }
45 }
46
47 /* Ispis rezultata. */
48 if (unet_bar_jedan == 0)
49     printf("Nije unet nijedan broj.\n");
50 else if (unet_pozitivan == 0)
51     printf("Medju unetim brojevima nema pozitivnih.\n");
52 else
53     printf("Proizvod pozitivnih brojeva je %d.\n", proizvod);
54
55 return 0;
56 }

```

### Rešenje 2.3.16

```

1  #include <stdio.h>
2  #include <stdlib.h>
3
4  int main()
5  {
6      /* Deklaracije potrebnih promenljivih. */
7      int n, cifra, n_original;
8      int pronadjena_petica = 0;
9

```

## 2 Kontrola toka

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

13
    /* Pamti se originalna vrednost unetog broja. */
15    n_original = n;

17    /* Uzima se apsolutna vrednost unetog broja. */
    if (n < 0)
19        n = abs(n);

21    /* Petlja se izvrsava dok god ima cifara u broju. */
    while (n > 0) {
23        /* Izdvaja se poslednja cifra broja. */
        cifra = n % 10;

25
        /* Proverava se da li je ona jednaka broju 5 */
27        if (cifra == 5) {
            /* Ako jeste, vrednost odgovarajuceg indikatora se postavlja
29                na 1 i petlja se prekida. */
            pronadjena_petica = 1;
            break;
31        }

33
        /* Ako petlja nije prekinuta, iz broja se uklanja poslednja
35        cifra i postupak se ponavlja dok god ima neobradjenih
        cifara. */
        n = n / 10;
37    }

39
    /* Ispis rezultata.
41    Napomena: Koristi se unapred zapamcena promenljiva n_original
        jer je promenljiva n izmenjena u petlji. */
43    if (pronadjena_petica == 0)
        printf("Broj %d sadrzi cifru 5.\n", n_original);
45    else
        printf("Broj %d ne sadrzi cifru 5.\n", n_original);

47    return 0;
49 }
```

### Rešenje 2.3.17

```
1  #include <stdio.h>

3  int main()
    {
5      /* Deklaracije i inicijalizacije potrebnih promenljivih. */
        int x;
7        int broj_brojeva = 0;
        int suma = 0;
```



```

9      /* Brojevi se ucitavaju u petlji sve do unosa broja 0. */
11     printf("Unesite brojeve: ");
12     while (1) {
13         scanf("%d", &x);

14
15         if (x == 0)
16             break;

17         /* Procitani broj se dodaje na sumu. */
18         suma += x;

19         /* Uvecava se broj ucitanih brojeva. */
20         broj_brojeva++;
21     }

22
23     /* Ispis rezultata.
24      Napomena: primetiti da su i suma i broj_brojeva celi brojevi
25      i da je neophodno bar jednu od te dve vrednosti pretvoriti
26      u realan broj kako deljenje ne bi bilo celobrojno. */
27     if (broj_brojeva == 0)
28         printf("Nisu uneti brojevi.\n");
29     else
30         printf("Aritmeticka sredina: %.4f\n",
31                (double) suma / broj_brojeva);

32
33     return 0;
34 }

```

### Rešenje 2.3.18

```

1  #include <stdio.h>

3  int main()
4  {
5      /* Deklaracije potrebnih promenljivih. */
6      float cena, suma = 0;
7      int broj_artikla = 0;

8
9      /* Cene se ucitavaju sve do unosa broja 0. */
10     printf("Unesite cenu: ");
11     while (1) {
12         scanf("%f", &cena);

13
14         if (cena == 0)
15             break;

16         /* Vrsi se provera ispravnosti ulaza. */
17         if (cena < 0) {
18             printf("Greska: neispravan unos cene.\n");
19             return -1;

```

## 2 Kontrola toka

---

```
21     }

23     /* Suma se uvecava za vrednost unete cene. */
    suma += cena;

25     /* Broj unetih artikala se uvecava za 1. */
27     broj_artikla++;
    }

29     /* Ispis rezultata. */
31     if (broj_artikla == 0)
        printf("Nisu unete cene.\n");
33     else
        printf("Prosecna cena: %.4f\n", suma / broj_artikla);

35     return 0;
37 }
```

### Rešenje 2.3.19

```
1  #include <stdio.h>

3  int main()
{
5     /* Deklaracija potrebnih promenljivih. */
    int n, i, broj_promena = 0;
7     double prethodni, trenutni;

9     /* Ucitava se vrednost broja n. */
    printf("Unesite broj n ");
11    scanf("%d", &n);

13    /* Vrsi se provera ispravnosti ulaza. */
    if (n <= 0) {
15        printf("Greska: neispravan unos.\n");
        return -1;
17    }

19    printf("Unesite brojeve: ");
    /* Provera promene znaka se vrsi za svaka dva susedna uneta
21    broja. Prvi broj se ucitava pre petlje i smesta se u
    promenljivu prethodni. Zatim se u petlji ucitava drugi i
23    njihov znak se poredi. Postupak se ponavlja za sve parove,
    tako sto se uvek na kraju petlje poslednji ucitani broj
25    postavi da bude prethodni za sledecu iteraciju. */
    scanf("%lf", &prethodni);

27

29    /* Kako je vec jedan broj unet, brojac se postavlja na 1, a ne
    na 0. */
    for (i = 1; i < n; i++) {
31
```

```

33      /* Ucitava se broj. */
      scanf("%lf", &trenutni);

35      /* Proverava se da li je doslo do promene znaka izmedju
         prethodnog i trenutnog. Oni su razlicitog znaka ako vazi:
37         1. da im je proizvod negativan ILI
         2. da im je proizvod nula, a jedan od njih je negativan. */
39      if (prethodni * trenutni < 0)
          broj_promena++;
41      else if (prethodni * trenutni == 0 &&
               (prethodni < 0 || trenutni < 0))
43          broj_promena++;

45      /* Trenutni broj postaje prethodni za sledecu iteraciju. */
      prethodni = trenutni;
47  }

49      /* Ispis rezultata. */
      printf("Broj promena je %d.\n", broj_promena);
51
53      return 0;
}

```

### Rešenje 2.3.20

```

1  #include <stdio.h>

3  int main()
  {
5      /* Deklaracije potrebnih promenljivih. */
      int n, i;
7      float cena, min_cena;

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

13     /* Vrsi se provera ispravnosti ulaza. */
      if (n <= 0) {
15         printf("Greska: neispravan unos.\n");
         return -1;
17     }

19     printf("Unesite cene artikala:");

21     /* Minimalna cena se inicijalizuje na cenu prvog artikla. Zbog
        toga se cena prvog artikla ucitava pre petlje. */
23     scanf("%f", &cena);
      if (cena <= 0) {
25         printf("Greska: neispravan unos cene.\n");
         return -1;

```

## 2 Kontrola toka

---

```
27     }
    min_cena = cena;
29
    /* Ucitava se i preostalih n-1 cena i racuna se najmanja. */
31    for(i=1; i<n; i++){
        scanf("%f", &cena);
33
        if (cena <= 0) {
35            printf("Greska: neispravan unos cene.\n");
            return -1;
37        }
39        if (cena < min_cena)
            min_cena = cena;
41        i++;
    }
43
    /* Ispis rezultata. */
45    printf("Najmanja cena: %f\n", min_cena);
47
    return 0;
}
```

### Rešenje 2.3.21

```
1  #include <stdio.h>
3  int main()
{
5     /* Deklaracije potrebnih promenljivih. */
    float cena, m;
7     unsigned int n, i, broj_artikala = 0;
9
    /* Ucitava se vrednost broja m. */
    printf("Nikolin budzet: ");
11    scanf("%f", &m);
13
    /* Ucitava se broj artikala. */
    printf("Unesite broj artikala: ");
15    scanf("%u", &n);
17
    /* Unose se cene artikala i racuna se rezultat. */
    printf("Unesite cene artikala: ");
19
    for(i=0; i<n; i++){
21        /* Ucitava se cena artikla. */
        scanf("%f", &cena);
23
        /* Provera se da li Nikola moze da kupi trenutni artikal. */
25        if (cena <= m)
            broj_artikala++;
    }
}
```

```

27 }

29 /* Ispis rezultata. */
printf("Ukupno artikala: %d\n", broj_artikala);

31 return 0;

33 }

```

### Rešenje 2.3.22

Rešenje (a)

```

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

3

5  int main()
6  {
   /* Deklaracije potrebnih promenljivih. */
7  int n, i, x, rezultat;
   int x_desetica, najveca_desetica;

9

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

13

   /* Vrsi se provera ispravnosti ulaza. */
15 if (n < 0) {
       printf("Greska: neispravan unos.\n");
17     return -1;
   }

19

   /* Ako nema unetih brojeva, ispisuje se odgovarajuca poruka. */
21 if (n == 0) {
       printf("Nisu uneti brojevi.\n");
23     return 0;
   }

25

   printf("Unesite brojeve: ");

27

   /* Prvi broj se ucitava pre petlje, zbog ispravne
29   inicijalizacije. */
   scanf("%d", &x);
31 /* Promenljiva najveca_desetica se postavlja na cifru desetica
   ucitanog broja. Napomena: pri racunanju se uzima apsolutna
33   vrednost broja jer je npr. (-123/10)= -12 i -12 % 10 = -2,
   a cifra desetica treba da bude 2. */
35 najveca_desetica = (abs(x) / 10) % 10;
   /* Kako je na kraju potrebno ispisati broj cija je cifra desetica
37   najveca, trenutna vrednost rezultata se postavlja na vrednost
   ucitanog broja. */

```

## 2 Kontrola toka

```
39  rezultat = x;

41  /* Ucitava se i preostalih n-1 brojeva i ako se naidje na broj
    cija je cifra desetica veca od trenutno najvece, azuriraju
43  se vrednosti najvece desetice i rezultata. */
44  for (i = 1; i < n; i++) {
45      scanf("%d", &x);

47      x_desetica = (abs(x) / 10) % 10;

49      if (x_desetica > najveca_desetica) {
        najveca_desetica = x_desetica;
51      rezultat = x;
52      }
53  }

55  /*II nacin: Inicijalizacija najvece desetice na neku vrednost
    koja je sigurno manja od svih vrednosti koje cifra desetica
57  moze da uzme (dakle, bilo sta sto je manje od 0 jer cifra
    desetica moze imati vrednosti izmedju 0 i 9).
59  Zatim se u petlji izracunava rezultat, analogno prvom nacinu.

61  najveca_desetica = -1;
    for(i=0; i<n; i++)
63  {
        scanf("%d", &x);

65        x_desetica = (abs(x) / 10) % 10;

67        if (x_desetica > najveca_desetica) {
            najveca_desetica = x_desetica;
69            rezultat = x;
71        }
72    }
73    */

75    /* Ispis rezultata. */
    printf("Broj sa najvecom cifrom desetice: %d\n", rezultat);
77
78    return 0;
79 }
```

### Rešenje (b)

```
1  #include <stdio.h>
2  #include <stdlib.h>
3
4  int main()
5  {
6      /* Deklaracije potrebnih promenljivih. */
7      int n, i;
```

```
9   int x, x_kopija, broj_cifara;
10  int najveci_broj_cifara, rezultat;

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

14  /* Vrsi se provera ispravnosti ulaza. */
15  if (n < 0) {
16      printf("Greska: neispravan unos.\n");
17      return -1;
18  }

19  /* Ako nema unetih brojeva, ispisuje se odgovarajuca poruka. */
20  if (n == 0) {
21      printf("Nisu uneti brojevi.\n");
22      return 0;
23  }

24  /* Maksimalan broj cifara se postavlja na 0 jer svaki broj ima
25     vise od 0 cifara. */
26  najveci_broj_cifara = 0;

27  printf("Unesite n brojeva: ");
28  for (i = 0; i < n; i++) {
29      scanf("%d", &x);

30      /* Racuna se broj cifara unetog broja x. */
31      x_kopija = abs(x);
32      broj_cifara = 0;
33      do {
34          broj_cifara++;
35          x_kopija = x_kopija / 10;
36      } while (x_kopija != 0);

37      /* Ako je broj cifara unetog broja veci od najveceg broja
38         cifara, azuriraju se vrednosti najveceg broja cifara i
39         tekuceg rezultata. */
40      if (broj_cifara > najveci_broj_cifara) {
41          najveci_broj_cifara = broj_cifara;
42          rezultat = x;
43      }
44  }

45  /* Ispis rezultata. */
46  printf("Najvise cifara ima broj %d.\n", rezultat);

47  return 0;
48 }
```

Rešenje (c)

```
1  #include <stdio.h>
2  #include <stdlib.h>
3
4  int main()
5  {
6      /* Deklaracije potrebnih promenljivih. */
7      int n, i;
8      int x, x_kopija, vodeca_cifra;
9      int najveca_vodeca_cifra, rezultat;
10
11     /* Ucitava se vrednost broja n. */
12     printf("Unesite broj n: ");
13     scanf("%d", &n);
14
15     /* Vrsi se provera ispravnosti ulaza. */
16     if (n < 0) {
17         printf("Greska: neispravan unos.\n");
18         return -1;
19     }
20
21     /* Ako nema unetih brojeva, ispisuje se odgovarajuca poruka. */
22     if (n == 0) {
23         printf("Nisu uneti brojevi.\n");
24         return 0;
25     }
26
27     /* Inicijalizacija najvece vodece cifre na -1. */
28     najveca_vodeca_cifra = -1;
29
30     printf("Unesite n brojeva: ");
31     for (i = 0; i < n; i++) {
32         scanf("%d", &x);
33
34         /* Racuna se vodeca cifra ucitanog broja x. */
35         x_kopija = abs(x);
36         while (x_kopija > 10) {
37             x_kopija = x_kopija / 10;
38         }
39         vodeca_cifra = x_kopija;
40
41         /* Ako je izdvojena cifra veca od najvece vodece cifre,
42            azuriraju se vrednosti najvece vodece cifre i rezultata. */
43         if (vodeca_cifra > najveca_vodeca_cifra) {
44             najveca_vodeca_cifra = vodeca_cifra;
45             rezultat = x;
46         }
47     }
48
49     /* Ispis rezultata. */
50     printf("%d\n", rezultat);
```



```
52 return 0;
}
```

### Rešenje 2.3.23

```
1  #include <stdio.h>
3  int main()
4  {
5      /* Deklaracija potrebnih promenljivih. */
6      int x;
7      int najmanja, najveca;
9      printf("Unesite brojeve: ");
10     /* Prvi broj se ucitava izvan petlje zbog inicijalizacije
11        najvece i najmanje vrednosti nadmorske visine.
12        Napomena: Ovde bi inicijalizacija najveca=-1 bila pogresna
13        jer moze da se desi da su svi uneti brojevi negativni i manji
14        od -1 i onda bi najveca i nakon izvorsavanja tela petlje ostala
15        -1. */
16     scanf("%d", &x);
17     najveca = x;
18     najmanja = x;
19
20     /* Ako nema unetih nadmorskih visina, ispisuje se odgovarajuca
21        poruka. */
22     if(x == 0)
23     {
24         printf("Nisu unete nadmorske visine.");
25         return 0;
26     }
27
28     /* Za svaki ucitani broj se proverava da li je manji od najmanje
29        ili veci od najvece i vrsi se azuriranje odgovarajucih
30        vrednosti. Petlja se prekida kada se unese broj 0.*/
31     while (1) {
32         scanf("%d", &x);
33
34         if(x == 0)
35             break;
36
37         if (x > najveca)
38             najveca = x;
39
40         if (x < najmanja)
41             najmanja = x;
42     }
43
44     /* Ispis rezultata. */
45     printf("Razlika: %d\n", najveca - najmanja);
```

```
47     return 0;
    }
```

### Rešenje 2.3.24

```
1  #include <stdio.h>
2  #include <stdlib.h>
3
4  int main()
5  {
6      /* Deklaracija potrebnih promenljivih. */
7      int n, d, i;
8      int x, y;
9      int broj_parova = 0;
10
11     /* Ucitavaju se vrednosti n i d. */
12     printf("Unesite brojeve n i d: ");
13     scanf("%d %d", &n, &d);
14
15     /* Vrsi se provera ispravnosti ulaza. */
16     if (n <= 1 || d < 0) {
17         printf("Greska: neispravan unos.\n");
18         return -1;
19     }
20
21     printf("Unesite n brojeva: ");
22
23     /* Prvi broj se ucitava pre petlje. */
24     scanf("%d", &x);
25
26     for (i = 1; i < n; i++) {
27         scanf("%d", &y);
28
29         /* Provera se da li su brojevi na rastojanju d. */
30         if (abs(y - x) == d)
31             broj_parova++;
32
33         /* Broj iz tekuce iteracije se cuva kako bi mogao da se
34            upotrebljava u narednoj iteraciji. */
35         x = y;
36     }
37
38     /* Ispis rezultata. */
39     printf("Broj parova: %d\n", broj_parova);
40
41     return 0;
42 }
```

### Rešenje 2.3.25

```
1  #include <stdio.h>
3  int main()
4  {
5      /* Deklaracija potrebnih promenljivih. */
6      int n, cifra;
7      unsigned int rezultat;
8      int pozicija;
9
10     /* Ucitava se vrednost broja n. */
11     printf("Unesite broj: ");
12     scanf("%d", &n);
13
14     /* Vrsi se provera ispravnosti ulaza. */
15     if (n <= 0) {
16         printf("Greska: neispravan unos.\n");
17         return -1;
18     }
19
20     /* Inicijalizacija pozicije i rezultata.
21        Pozicija oznacava tezinu trenutne cifre i moze imati vrednosti
22        1, 10, 100, 1000, ... */
23     pozicija = 1;
24     rezultat = 0;
25
26     /* U petlji se izdvaja cifra po cifra, dok god ima neobradjenih
27        cifara. */
28     while (n > 0) {
29         /* Izdvaja se poslednja cifra iz zapisa i ako je njena vrednost
30            paran broj, uvecava se za 1. */
31         cifra = n % 10;
32         if (cifra % 2 == 0)
33             cifra++;
34
35         /* Novi broj se formira tako sto se izdvojena cifra pomnozi
36            odgovarajucom tezinom (stepenom) pozicije i doda na tekuci
37            rezultat. */
38         rezultat += cifra * pozicija;
39
40         /* Uklanja se poslednja cifra broja. */
41         n /= 10;
42
43         /* Pozicija se mnozi sa 10. */
44         pozicija *= 10;
45     }
46
47     /* Ispisuje se izracunatu vrednost. */
48     printf("Rezultat: %d\n", rezultat);
49
50     return 0;
51 }
```

### Rešenje 2.3.26

```
1 #include <stdio.h>
2 #include <stdlib.h>
3
4 int main()
5 {
6     /* Deklaracije potrebnih promenljivih. */
7     int x, pozicija, rezultat, cifra;
8     int znak = 1;
9
10    /* Ucitava se vrednost polaznog broja. */
11    printf("Unesite broj: ");
12    scanf("%d", &x);
13
14    /* Ako je broj negativan, uzima se njegova apsolutna vrednost
15     i azurira se vrednost znaka broja. */
16    if (x < 0) {
17        x = abs(x);
18        znak = -1;
19    }
20
21    /* Pozicija oznacava tezinu trenutne cifre rezultata i moze imati
22     vrednosti 1, 10, 100, ... */
23    pozicija = 1;
24    rezultat = 0;
25
26    /* Ideja: u rezultatu se zadrzavaju cifre jedinice, stotine,...
27     Na primer, x=12345
28     Pre petlje: pozicija = 1, rezultat = 0
29     1. iteracija:
30     cifra = 5, rezultat = 0+5*1=5, x = 123, pozicija = 10
31     2. iteracija:
32     cifra = 3, rezultat = 5+3*10 = 35, x = 12, pozicija = 100
33     3. iteracija:
34     cifra = 4, rezultat = 35+4*100, x = 1, pozicija = 1000
35     Petlja se zavrшава jer x ima vrednost 0. */
36    while (x > 0) {
37        /* Izdvajanje poslednje cifre. */
38        cifra = x % 10;
39
40        /* Rezultat se uvecava za vrednost cifre pomnozene sa vrednoscu
41         tezine njene pozicije u broju. */
42        rezultat += cifra * pozicija;
43
44        /* Iz polaznog broja se uklanjaju poslednje dve cifre jer u
45         rezultatu treba da ostane svaka druga cifra polaznog
46         broja.*/
47        x /= 10;
48
49        /* Pozicija se mnozi sa 10, kako bi imala ispravnu vrednost u
50         sledecoj iteraciji. */
51    }
```

```
51     pozicija *= 10;
52 }
53
54 /* Ispis rezultata */
55 printf("Rezultat: %d\n", znak * rezultat);
56
57 return 0;
58 }
```

### Rešenje 2.3.27

```
1  #include <stdio.h>
2
3  int main()
4  {
5      /* Deklaracija potrebnih promenljivih. */
6      int n;
7      int c1, c2, c3;
8      int pozicija, rezultat;
9
10     /* Ucitava se vrednost broja n. */
11     printf("Unesite broj: ");
12     scanf("%d", &n);
13
14     /* Vrsi se provera ispravnosti ulaza. */
15     if (n <= 0) {
16         printf("Greska: neispravan unos.\n");
17         return -1;
18     }
19
20     /* Ako broj nema bar tri cifre, rezultat ima vrednost unetog
21        broja. */
22     if(n <= 99)
23     {
24         printf("Rezultat: %d\n", n);
25         return 0;
26     }
27
28     /* Izdvajaju se poslednje tri cifre polaznog broja. */
29     c1 = n%10;
30     c2 = (n/10)%10;
31     c3 = (n/100)%10;
32
33     /* Poslednja cifra se uvek nalazi u rezultatu jer ona nema
34        oba suseda. Zato rezultat inicijalizujemo na poslednju cifru,
35        a poziciju na 10. */
36     rezultat = c1;
37     pozicija = 10;
38
39     /* Petlja se izvrsava dok god broj ima bar tri cifre. */
40     while(n>99)
```

## 2 Kontrola toka

```
41 {
42     /* Proverava se da li c2 treba da se nadje u rezultatu. Ako
43        treba, rezultat se uvecava za vrednost cifre pomnozenu sa
44        vrednoscu tezine njene pozicije u rezultatu i tezina
45        pozicije se mnozi sa 10. */
46     if(c2 != c1 + c3)
47     {
48         rezultat += c2*pozicija;
49         pozicija *= 10;
50     }
51
52     /* Vrsi se pomeranje na sledece tri cifre polaznog broja.
53        Iz polaznog broja brisemo poslednju cifru. Prva i druga
54        cifra su vec izracunate, samo se vrsi njihovo premestanje
55        iz c2 i c3 u c1 i c2. Cifra c3 se racuna. */
56     n = n/10;
57     c1 = c2;
58     c2 = c3;
59     c3 = (n/100)%10;
60 }
61
62 /* Po završetku petlje, broj n je dvocifren i njegova cifra
63    desetica odgovara vodećoj cifri polaznog broja. Vodeća cifra
64    polaznog broja uvek treba da se nadje u rezultatu jer nema
65    oba suseda i iz tog razloga je dodajemo na tekuci rezultat. */
66 rezultat += (n/10)*pozicija;
67
68 /* Ispis rezultata. */
69 printf("%d\n", rezultat);
70
71 return 0;
72 }
```

### Rešenje 2.3.28

```
1  #include <stdio.h>
2
3  int main()
4  {
5      /* Deklaracija potrebnih promenljivih. */
6      int x, x_kopija, x_obrnuto;
7
8      /* Ucitava se vrednost pocetnog broja. */
9      printf("Unesite broj: ");
10     scanf("%d", &x);
11
12     /* Uzima se apsolutna vrednost unetog broja. */
13     if (x < 0)
14         x = -x;
15
16     /* Racuna se broj koji se dobije kada se broju x obrnu cifre.
17
18     */
19 }
```

```

17     Na primer, od 12345 treba da se dobije 54321.
    Broj se obrće tako što se u svakoj iteraciji njegova vrednost
19 pomnoži sa 10 i doda mu se sledeća cifra polaznog broja.
    Za x_kopija=12345, x_obruto = 0
21     1. iteracija: x_obruto = 0*10 + 5 = 5, x_kopija = 1234
    2. iteracija: x_obruto = 5*10 + 4 = 54, x_kopija = 123,
23     3. iteracija: x_obruto = 54*10 + 3 = 543, itd.*/
x_kopija = x;
25 x_obruto = 0;
while (x_kopija != 0) {
27     x_obruto = x_obruto * 10 + x_kopija % 10;
    x_kopija /= 10;
29 }

31 /* Broj je palindrom ako je jednak broju koji se dobije
    obrtanjem njegovih cifara.
33     Npr. x = 12321, x_obruto je takodje 12321.*/
if (x == x_obruto)
35     printf("Broj je palindrom.\n");
else
37     printf("Broj nije palindrom.\n");
39 return 0;
}

```

### Rešenje 2.3.29

```

1  #include <stdio.h>

3  int main()
{
5     /* Deklaracija potrebnih promenljivih. */
    int n, i;
7     int fib1 = 0, fib2 = 1, fib3;

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

13    /* Vrsi se provera ispravnosti ulaza. */
    if (n < 0) {
15        printf("Greska: neispravan unos.\n");
        return -1;
17    }

19    /* Ako je n=0, F[0] = 0, slicno ako je n=1 F[1] = 1. */
    if(n < 2){
21        printf("F[%d] = %d\n",n, n);
        return 0;
23    }
}

```

## 2 Kontrola toka

---

```
25 fib3 = fib1 + fib2;
26 for(i=2; i<n; i++) {
27     /* Vrsi se pomeranje na sledecu trojku. */
28     fib1 = fib2;
29     fib2 = fib3;
30     fib3 = fib1 + fib2;
31 }
32
33 /* Ispis rezultata. */
34 printf("F[%d] = %d\n", n, fib3);
35
36 return 0;
37 }
```

### Rešenje 2.3.30

```
#include<stdio.h>
2
3 int main()
4 {
5     /* Deklaracija potrebnih promenljivih. */
6     int an;
7
8     /* Ucitava se vrednost prvog clana. */
9     printf("Unesite prvi clan:");
10    scanf("%d", &an);
11
12    /* Vrsi se provera ispravnosti ulaza. */
13    if (an <= 0) {
14        printf("Greska: neispravan unos.\n");
15        return -1;
16    }
17
18    /* Dok se ne dodje do clana koji je 1, stampa se vrednost
19       trenutnog clana i vrsi se izracunavanje narednog, po
20       zadatoj formuli. */
21    while (an != 1) {
22        printf("%d ", an);
23
24        if (an % 2 != 0)
25            an = (3 * an + 1) / 2;
26        else
27            an = an / 2;
28    }
29
30    /* Na kraju se stampa i jedinica. */
31    printf("1\n");
32
33    return 0;
34 }
```



## Rešenje 2.3.31

```
1  #include <stdio.h>
2  #include <math.h>
3
4  int main()
5  {
6      /* Deklaracija potrebnih promenljivih. */
7      int format, i;
8      double sirina, duzina, nova_duzina;
9
10     /* Ucitava se format papira. */
11     printf("Unesite format papira: ");
12     scanf("%d", &format);
13
14     /* Vrsi se provera ispravnosti ulaza. */
15     if (format < 0) {
16         printf("Greska: neispravan unos.\n");
17         return -1;
18     }
19
20     /* duzina/sirina = 1/sqrt(2)
21        duzina*sirina = 1000mm x 1000mm
22        =>
23        duzina = sirina/sqrt(2)
24        duzina*sirina = 1000mm x 1000mm
25        =>
26        sirina*sirina/sqrt(2) = 1000*1000
27        sirina*sirina = sqrt(2) * 1000 * 1000
28        sirina = sqrt(sqrt(2) * 1000 * 1000)
29        =>
30        duzina = sirina/sqrt(2) */
31     sirina = sqrt(1000 * 1000 * sqrt(2));
32     duzina = sirina / sqrt(2);
33
34     /* U petlji se racunaju duzina i sirina za uneti format. */
35     for (i = 1; i <= format; i++) {
36         nova_duzina = sirina / 2;
37         sirina = duzina;
38         duzina = nova_duzina;
39     }
40
41     /* Ispis rezultata. Napomena: duzina i sirina celi brojevi. */
42     printf("%d %d\n", (int) duzina, (int) sirina);
43
44     return 0;
45 }
```

## Rešenje 2.3.32

```
1 #include <stdio.h>

3 int main()
{
5     char c;

7     /* I nacin ucitavanja:
        U samom uslovu petlje se vrsi ucitavanje jednog karaktera,
        njegovo smestanje u promenljivu c i provera da li je ucitani
        karakter tacka. Zgrade oko (c=getchar()) su obavezne jer
        relacioni operator != ima veci prioritet od dodele i kada ne
        bi postojale zgrade, redosled operacija bi bio:
        (c = (getchar() != '.')), sto znaci da bi se u c smestio
        rezultat poredjenja, odnosno 0 ili 1. */
15    while ((c = getchar()) != '.') {
        /* Proveravaju se uslovi i vrsi se ispis odgovarajuceg
        karaktera.*/
17        if (c >= 'A' && c <= 'Z')
19            putchar(c + 'a' - 'A');
        else if (c >= 'a' && c <= 'z')
21            putchar(c - 'a' + 'A');
        else
23            putchar(c);
    }

25    /*II nacin:
27    while(1) {
        c = getchar();
29        if(c == '.')
            break;

31        if (c >= 'A' && c <= 'Z')
33            putchar(c + 'a' - 'A');
        else if (c >= 'a' && c <= 'z')
35            putchar(c - 'a' + 'A');
        else
37            putchar(c);
    } */

39    return 0;
41 }
```

### Rešenje 2.3.33

```
1 #include <stdio.h>

3 int main()
{
5     /* Deklaracije i inicijalizacije. */
    char c;
7     int broj_velikih = 0, broj_malih = 0;
```

```

9      int broj_cifara = 0, suma_cifara = 0, broj_belina = 0;

11     /* Petlja se završava kada korisnik zada konstantu oznaku za kraj
        ulaza (konstanta EOF čija je vrednost -1). Ova konstanta se
        zadaje kombinacijom tastera CTRL+D. */
13     while ((c = getchar()) != EOF) {
        if (c >= 'A' && c <= 'Z')
15         broj_velikih++;
        else if (c >= 'a' && c <= 'z')
17         broj_malih++;
        else if (c >= '0' && c <= '9') {
19             broj_cifara++;
            suma_cifara = suma_cifara + c - '0';
21         }
        else if (c == '\t' || c == '\n' || c == ' ')
23             broj_belina++;
    }

25     /* Ispis rezultata. */
27     printf("velika: %d, mala: %d\n", broj_velikih, broj_malih);
    printf("cifre: %d, beline: %d\n", broj_cifara, broj_belina);
29     printf("suma cifara: %d\n", suma_cifara);

31     return 0;
}

```

### Rešenje 2.3.34

```

1  #include <stdio.h>

3  int main()
{
5     /* Deklaracija i inicijalizacija potrebnih promenljivih. */
    int n, i;
7     int broj_a = 0, broj_e = 0, broj_i = 0, broj_o = 0, broj_u = 0;
    char c;

9     /* Učitava se broj karaktera. */
11    printf("Unesite broj n: ");
    scanf("%d", &n);

13    /* Vrsi se provera ispravnosti ulaza. */
15    if (n < 0) {
        printf("Greska: neispravan unos.\n");
17        return -1;
    }

19    /* Kako je korisnik nakon unosa broja n uneo oznaku za novi red,
        potrebno je preskociti taj novi red jer bi u suprotnom on bio
        ucitan kao prvi od n karaktera (oznaka za novi red je
23    regularan karakter kao sto je to 'a' ili ' ').*/

```

```
    getchar();

25
    /* Ucitavaju se karakteri i broje se samoglasnici. */
27    for (i = 0; i < n; i++) {
        scanf("%c", &c);

29
        switch (c) {
31            case 'a':
            case 'A':
                broj_a++;
                break;
33            case 'e':
            case 'E':
                broj_e++;
                break;
35            case 'i':
            case 'I':
                broj_i++;
                break;
37            case 'o':
            case 'O':
                broj_o++;
                break;
39            case 'u':
            case 'U':
                broj_u++;
                break;
41
43        }
45    }
47
49    /* Ispis rezultata. */
51    printf("Samoglasnik a: %d\n", broj_a);
53    printf("Samoglasnik e: %d\n", broj_e);
55    printf("Samoglasnik i: %d\n", broj_i);
57    printf("Samoglasnik o: %d\n", broj_o);
59    printf("Samoglasnik u: %d\n", broj_u);
61
    return 0;
}
```

### Rešenje 2.3.35

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

3
4 int main()
5 {
6     /* Deklaracija i inicijalizacija potrebnih promenljivih. */
7     int n, i;
8     int broj_Z = 0, broj_i = 0, broj_m = 0, broj_a = 0;
9     char novi_red, c;
```

```

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

15  /* Vrsi se provera ispravnosti ulaza. */
    if (n <= 0) {
17      printf("Greska: neispravan unos.\n");
      return -1;
19  }

21  /* Ucitavaju se karakteri. */
    for (i = 1; i <= n; i++) {
23      printf("Unestite %d. karakter: ", i);

25      /* Prvo se cita belina koja se nalazi nakon prethodnog unosa,
        pa tek posle procitane beline se cita uneti karakter. */
27      scanf("%c%c", &novi_red, &c);

29      /* Obradjuje se ucitani karakter. */
        switch (c) {
31          case 'Z':
            broj_Z++;
33            break;
            case 'i':
            broj_i++;
35            break;
            case 'm':
            broj_m++;
37            break;
            case 'a':
            broj_a++;
41            break;
43        }
    }

45  /* Ako su svi brojacii razliciti od nule, rec "Zima" se moze
47  napisati pomocu unetih karaktera. */
    if (broj_Z && broj_i && broj_m && broj_a)
49      printf("Moze se napisati rec Zima.\n");
    else
51      printf("Ne moze se napisati rec Zima.\n");

53  return 0;
}

```

### Rešenje 2.3.36

```

1  #include <stdio.h>

3  int main()

```

```
{
5  /* Deklaracije potrebnih promenljivih. */
   int n, i;
7  int suma_kubova;

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

13  /* Vrsi se provera ispravnosti ulaza. */
   if (n <= 0) {
15     printf("Greska: neispravan unos.\n");
     return -1;
17  }

19  /* Racuna se suma kubova svih brojeva iz intervala [1,n]. */
   suma_kubova = 0;
21  for (i = 1; i <= n; i++)
       suma_kubova += i * i * i;

23  /* Ispis rezultata. */
25  printf("Suma kubova: %d\n", suma_kubova);

27  return 0;
}
```

Rešenje 2.3.37      Rešenje je analogno rešenju zadatka 2.3.36.

### Rešenje 2.3.38

```
1  #include <stdio.h>

3  int main()
   {
5     /* Deklaracija potrebnih promenljivih. */
       int n, i;
       float x, suma, x_i;

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

13     /* Vrsi se provera ispravnosti ulaza. */
       if (n <= 0) {
15         printf("Greska: neispravan unos.\n");
           return -1;
17     }

19     /* Vrednost sume se inicijalizuje na nulu, a vrednost x^i
       na x. */
   }
```

```
21 suma = 0;
22 x_i = x;
23
24 /* Promenljiva x^i ima vrednosti [x, x^2, ..., x^n].
25    Vrednost sume se u svakoj iteraciji uvecava za i*x^i. */
26 for (i = 1; i <= n; i++) {
27     suma += i * x_i;
28     x_i *= x;
29 }
30
31 /* Ispis rezultata. */
32 printf("S=%f\n", suma);
33
34 return 0;
35 }
```

## Rešenje 2.3.39

```
1 #include <stdio.h>
2
3 int main()
4 {
5     /* Deklaracija potrebnih promenljivih. */
6     int n, i;
7     float x, suma, x_i;
8
9     /* Ucitavaju se vrednosti x i n. */
10    printf("Unesite redom brojeve x i n: ");
11    scanf("%f %d", &x, &n);
12
13    /* Vrsi se provera ispravnosti ulaza. */
14    if (n <= 0) {
15        printf("Greska: neispravan unos.\n");
16        return -1;
17    }
18
19    /* Racuna se trazena suma. */
20    suma = 1;
21    x_i = x;
22    for (i = 1; i <= n; i++) {
23        suma += 1 / x_i;
24        x_i *= x;
25    }
26
27    /* Ispis rezultata. */
28    printf("S=%f\n", suma);
29
30    return 0;
31 }
```

### Rešenje 2.3.40

```
1 #include <stdio.h>
2 #include <math.h>
3
4 int main()
5 {
6     /* Deklaracije potrebnih promenljivih. */
7     int i;
8     float suma, clan;
9     float x, eps;
10
11     /* Ucitavaju se vrednosti x i eps. */
12     printf("Unesite x: ");
13     scanf("%f", &x);
14
15     printf("Unesite tacnost eps: ");
16     scanf("%f", &eps);
17
18     /* Inicijalizacija sume, prvog clana i brojaca. */
19     suma = 0;
20     clan = 1;
21     i = 1;
22
23     /* U svakoj iteraciji na sumu se dodaje prethodno izracunati
24     clan sume i zatim se racuna sledeci clan. Petlja se prekida
25     kada vrednost sledeceg clana postane manja ili jednaka eps. */
26     while (clan > eps) {
27         suma += clan;
28         clan = clan * x / i;
29         i++;
30     }
31
32     /* Ispis rezultata. */
33     printf("S=%f\n", suma);
34
35     return 0;
36 }
```

### Rešenje 2.3.41

```
1 #include <stdio.h>
2 #include <math.h>
3
4 int main()
5 {
6     /* Deklaracije potrebnih promenljivih. */
7     int i;
8     float suma;
9     float x, eps, clan;
```



```

10  /* Ucitavaju se vrednosti x i eps. */
12  printf("Unesite x: ");
13  scanf("%f", &x);
14
15  printf("Unesite tacnost eps: ");
16  scanf("%f", &eps);
17
18  /* Inicijalizacije. */
19  suma = 0;
20  clan = 1;
21  i = 1;
22
23  /* Kako clanovi sume mogu biti negativni, potrebno je posmatrati
24     apsolutnu vrednost clana. */
25  while (fabs(clan) > eps) {
26      suma += clan;
27
28      /* U svakoj iteraciji se racuna novi clan i mnozi se sa -1.
29         Na taj nacin se postize da je vrednost clana naizmenicno
30         pozitivna i negativna. */
31      clan = clan * x / i;
32      clan *= -1;
33
34      i++;
35  }
36
37  /* Ispis rezultata. */
38  printf("S=%f\n", suma);
39
40  return 0;
41 }

```

### Rešenje 2.3.42

```

1  #include <stdio.h>
2  #include <math.h>
3
4  int main()
5  {
6      /* Deklaracije potrebnih promenljivih. */
7      int n, i;
8      double x, x_i, proizvod;
9
10     /* Ucitavaju se vrednosti x i n. */
11     printf("Unesite redom brojeve x i n: ");
12     scanf("%lf %d", &x, &n);
13
14     /* Vrsi se provera ispravnosti ulaza. */
15     if (n <= 0) {
16         printf("Greska: neispravan unos.\n");
17     }
18 }

```

## 2 Kontrola toka

---

```
17     return -1;
18 }
19
20 /* Racuna se trazeni proizvod. */
21 x_i = 1;
22 proizvod = 1;
23 for (i = 0; i < n; i++) {
24     x_i *= x;
25     proizvod *= 1 + cos(x_i);
26 }
27
28 /* Ispis rezultata. */
29 printf("P = %lf\n", proizvod);
30
31 return 0;
32 }
```

### Rešenje 2.3.43

```
1  #include <stdio.h>
2
3  int main()
4  {
5      /* Deklaracije potrebnih promenljivih. */
6      int n, i;
7      double razlomak;
8
9      /* Ucitava se vrednost broja n. */
10     printf("Unesite broj n: ");
11     scanf("%d", &n);
12
13     /* Vrsi se provera ispravnosti ulaza. */
14     if (n <= 0) {
15         printf("Greska: neispravan unos.\n");
16         return -1;
17     }
18
19     /* Razlomak se izracunava "od nazad", odnosno, krece se od
20        najnizeg razlomka 1/n i od njega se nadalje formira sledeci,
21        "visi" razlomak itd. Zavrшава se kada se stigne do koraka 0 +
22        1/R. */
23     razlomak = n;
24     for (i = n - 1; i >= 0; i--)
25         razlomak = i + 1 / razlomak;
26
27     /* Ispis rezultata. */
28     printf("R = %lf\n", razlomak);
29
30     return 0;
31 }
```

## Rešenje 2.3.44

```
1 #include <stdio.h>
3 int main () {
4     /* Deklaracije potrebnih promenljivih. */
5     int i , n;
6     float suma, x, clan;
7
8     /* Ucitavaju se vrednosti x i n. */
9     printf("Unesite redom brojeve x i n: ");
10    scanf("%f%d", &x, &n);
11
12    /* Vrsi se provera ispravnosti ulaza. */
13    if (n <= 0) {
14        printf("Greska: neispravan unos.\n");
15        return -1;
16    }
17
18    /* Inicijalizacije. */
19    suma = 1;
20    clan = 1;
21    i = 2;
22
23    /* Racuna se trazena suma. */
24    while (i <= 2 * n) {
25        /* Svaki clan suma se od prethodnog clana razlikuje za
26            $x^2/(i*(i-1))$ . */
27        clan = clan * x * x / (i * (i - 1));
28        clan *= -1;
29        suma += clan;
30        i += 2;
31    }
32
33    /* Ispis rezultata. */
34    printf("S=%f\n", suma);
35
36    return 0;
37 }
```

## Rešenje 2.3.45

```
1 #include <stdio.h>
3 int main()
4 {
5     /* Deklaracije potrebnih promenljivih. */
6     int n, i;
7     double clan, proizvod = 1;
```

## 2 Kontrola toka

---

```
9  /* Ucitava se vrednost broja n. */
   printf("Unesite broj n: ");
11  scanf("%d", &n);

13  /* Vrsi se provera ispravnosti ulaza. */
   if (n <= 0) {
15      printf("Greska: neispravan unos.\n");
      return -1;
17  }

19  /* Racuna se trazeni proizvod. */
   clan = 1;
21  for (i = 2; i <= n; i++) {
      clan = clan / i;
23      proizvod *= 1 + clan;
   }

25  /* Ispis rezultata. */
27  printf("P = %lf\n", proizvod);

29  return 0;
}
```

### Rešenje 2.3.46

```
1  #include <stdio.h>

3  int main()
   {
5      /* Deklaracija potrebnih promenljivih. */
      int n, i;
      long int clan, suma = 0;

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

13     /* Vrsi se provera ispravnosti ulaza. */
      if (n < 5 || n % 2 == 0) {
15         printf("Greska: neispravan unos.\n");
         return -1;
17     }

19     /* Izracunava se trazena suma. */
      clan = -1 * 3;
21     for (i = 5; i <= n; i += 2) {
         clan = -1 * clan * i;
23         suma += clan;
     }

25     /* Ispis rezultata. */
```

```

27     printf("S = %ld\n", suma);
29     return 0;
}

```

### Rešenje 2.3.47

```

#include <stdio.h>

2 int main()
4 {
    /* Deklaracije potrebnih promenljivih. */
    6 int n, i;
    double rezultat;
    8 double x, a;

    /* Ucitavaju se vrednosti ulaznih promenljivih. */
    10 printf("Unesite brojeve x i a: ");
    12 scanf("%lf%lf", &x, &a);
    printf("Unesite broj n: ");
    14 scanf("%d", &n);

    /* Vrsi se provera ispravnosti ulaza. */
    16 if (n <= 0) {
    18     printf("Greska: neispravan unos.\n");
        return -1;
    20 }

    /* Racuna se vrednost zadatog izraza. Krece se od
        rezultat = (x+a)^2 i ide se ka spolja.
    22     Svaki put vrednost rezultata treba zameniti sa
        (rezultat + a)^2. */
    24 rezultat = x;
    for (i = 0; i < n; i++)
    26     rezultat = (rezultat + a) * (rezultat + a);

    /* Ispis rezultata. */
    30 printf("Izraz = %lf\n", rezultat);
    32 return 0;
    34 }

```

### Rešenje 2.3.48

Rešenje (a)

```

#include <stdio.h>

2

```

## 2 Kontrola toka

---

```
1 int main()
2 {
3     /* Deklaracije potrebnih promenljivih. */
4     unsigned int n, i, j;
5
6     /* Ucitava se vrednost broja n. */
7     printf("Unesite broj n: ");
8     scanf("%u", &n);
9
10    /* Ispis tablice mnozenja dimenzije n*n. */
11    for (i = 1; i <= n; i++) {
12        for (j = 1; j <= n; j++){
13            /* Vrednost svakog polja je proizvod vrste i kolone. */
14            printf("%3d ", i * j);
15        }
16        /* Na kraju svake vrste se ispisuje novi red. */
17        printf("\n");
18    }
19
20    return 0;
21 }
```

### Rešenje (b)

```
1 #include <stdio.h>
2
3 int main()
4 {
5     /* Deklaracije potrebnih promenljivih. */
6     unsigned int n, i, j;
7
8     /* Ucitava se vrednost broja n. */
9     printf("Unesite broj n: ");
10    scanf("%u", &n);
11
12    /* Brojac koji broji koliko brojeva je ispisano u jednom redu. */
13    j = 0;
14    for (i = 1; i <= n * n; i++) {
15        printf("%3d ", i);
16
17        j++;
18        /* Kada je ispisano n brojeva u jednom redu, ispisuje se znak
19         za novi red, da bi ispis krenuo u novom redu i vrednost
20         brojaca j se postavlja na 0 jer u novom redu jos ni jedan
21         broj nije ispisan. */
22        if (j == n) {
23            j = 0;
24            printf("\n");
25        }
26    }
27 }
```

```
    return 0;
29 }
```

*Rešenje (c)*

```
1  #include <stdio.h>
3  int main()
4  {
5      /* Deklaracije potrebnih promenljivih. */
6      unsigned int n, i, j;
7
8      /* Ucitava se vrednost broja n. */
9      printf("Unesite broj n: ");
10     scanf("%u", &n);
11
12     /* Ispis trazene tablice. */
13     for (i = 1; i <= n; i++) {
14         for (j = 0; j < n; j++)
15             if ((j + i) % n == 0)
16                 printf("%3d", n);
17             else
18                 printf("%3d", (j + i) % n);
19
20         printf("\n");
21     }
22
23     return 0;
24 }
```

*Rešenje (d)*

```
1  #include <stdio.h>
2
3  int main()
4  {
5      /* Deklaracije potrebnih promenljivih. */
6      unsigned int n, i, j;
7
8      /* Ucitava se vrednost broja n. */
9      printf("Unesite broj n: ");
10     scanf("%u", &n);
11
12     /* Ispis trazenog trougla. */
13     for (i = 0; i < n; i++) {
14         for (j = 0; j < n - i; j++)
15             printf("(%d, %d)", i, j);
16
17         printf("\n");
18     }
19 }
```

```
18     }
20     return 0;
    }
```

### Rešenje 2.3.49

#### Rešenje (a)

```
#include <stdio.h>

2 int main()
3 {
4     /* Deklaracija potrebnih promenljivih. */
5     unsigned int n, i, j;
6
7     /* Ucitava se vrednost broja n. */
8     printf("Unesite broj n: ");
9     scanf("%u", &n);
10
11     /* Kvadrat predstavlja tabelu sa n vrsta gde svaka vrsta sadrzi
12        n polja, a svako polje je isto i predstavlja karakter *. */
13     for (i = 0; i < n; i++) {
14         for (j = 0; j < n; j++)
15             printf("*");
16         printf("\n");
17     }
18
19     return 0;
20 }
```

#### Rešenje (b)

```
1 #include <stdio.h>
2
3 int main()
4 {
5     /* Deklaracija potrebnih promenljivih. */
6     unsigned int n, i, j;
7
8     /* Ucitava se vrednost broja n. */
9     printf("Unesite broj n: ");
10    scanf("%u", &n);
11
12    /* Kvadrat predstavlja tabelu sa n vrsta gde su ivice karakter *,
13       a untrasnjost kvadrata je karakter blanko. */
14    for (i = 0; i < n; i++) {
15        for (j = 0; j < n; j++){
```



```

17     /* Provera se da li je u pitanju ivica. */
18     if (j == 0 || j == n - 1 || i == 0 || i == n - 1)
19         printf("*");
20     else
21         printf(" ");
22 }
23 printf("\n");
24 }
25 return 0;
26 }

```

Rešenje (c)

```

#include <stdio.h>

2 int main()
3 {
4     /* Deklaracija potrebnih promenljivih. */
5     unsigned int n, i, j;

6     /* Ucitava se vrednost broja n. */
7     printf("Unesite broj n: ");
8     scanf("%u", &n);

9     /* Kvadrat predstavlja tabelu sa n vrsta gde su ivice karakter *,
10      a untrasnjost kvadrata je karakter blanko osim na mestima na
11      kojima je glavna dijagonala. */
12     for (i = 0; i < n; i++) {
13         for (j = 0; j < n; j++){
14             /* Provera da li je ivica ili glavna dijagonala. */
15             if (j == 0 || j == n - 1 || i == 0 || i == n - 1 || i == j)
16                 printf("*");
17             else
18                 printf(" ");
19         }
20         printf("\n");
21     }
22     return 0;
23 }

```

### Rešenje 2.3.50

```

1 #include <stdio.h>
2
3 int main()
4 {
5     /* Deklaracija potrebnih promenljivih. */

```

## 2 Kontrola toka

```
    unsigned int n, i, j;

7
    /* Ucitava se vrednost broja n. */
9    printf("Unesite broj n: ");
    scanf("%u", &n);

11
    /* Veliko slovo X se dobija tako sto se na dijagonalama kvadrata
13     ispisuju karakteri *, a na ostalim mestima blanko. */
    for (i = 0; i < n; i++) {
15        for (j = 0; j < n; j++){
17            /* Provera da li je mesto glavne ili sporedne dijagonale. */
            if (i == j || i + j == n - 1)
                printf("*");
19            else
                printf(" ");
21        }
        printf("\n");
23    }

25    return 0;
}
```

### Rešenje 2.3.51

```
1 #include <stdio.h>

3 int main()
{
5     /* Deklaracija potrebnih promenljivih. */
    unsigned int n, i, j;

7
    /* Ucitava se vrednost broja n. */
9    printf("Unesite broj n: ");
    scanf("%u", &n);

11
    /* Iscrtava se znak plus tako sto se na pozicijama koje
13     odgovaraju sredisnjoj vrsti i sredisnjoj kolini ispisuje
    +, a na ostalim pozicijama se ispisuje blanko. */
    for (i = 0; i < n; i++) {
15        for (j = 0; j < n; j++)
17            if (i == n / 2 || j == n / 2)
                printf("+");
19            else
                printf(" ");
21        printf("\n");
    }

23
25    return 0;
}
```

## Rešenje 2.3.52

Rešenje (a)

```
1  #include <stdio.h>
2
3  int main()
4  {
5      /* Deklaracija potrebnih promenljivih. */
6      unsigned int n, i, j;
7
8      /* Ucitava se vrednost broja n. */
9      printf("Unesite broj n: ");
10     scanf("%u", &n);
11
12     /* Iscrtava se trazeni trougao. */
13     for (i = 0; i < n; i++) {
14         for (j = 0; j < n - i; j++)
15             printf("*");
16         printf("\n");
17     }
18
19     return 0;
20 }
```

Rešenje (b)

```
1  #include <stdio.h>
2
3  int main()
4  {
5      /* Deklaracija potrebnih promenljivih. */
6      unsigned int n, i, j;
7
8      /* Ucitava se vrednost broja n. */
9      printf("Unesite broj n: ");
10     scanf("%u", &n);
11
12     /* Iscrtava se trazeni trougao. */
13     for (i = 0; i < n; i++) {
14         for (j = 0; j <= i; j++)
15             printf("*");
16         printf("\n");
17     }
18
19     return 0;
20 }
```

Rešenje (c)

## 2 Kontrola toka

---

```
#include <stdio.h>

2
int main()
4 {
    /* Deklaracija potrebnih promenljivih. */
6     unsigned int n, i, j;

    /* Ucitava se vrednost broja n. */
8     printf("Unesite broj n: ");
10    scanf("%u", &n);

    /* Iscrtava se trazeni trougao. */
12    for (i = 0; i < n; i++) {
14        /* Prvo se ispisuju beline koje prethode karakterima *. */
        for (j = 0; j < i; j++)
16            printf(" ");
        /* Posle belina se ispisuje potreban broj karaktera *. */
18        for (j = 0; j < n - i; j++)
            printf("*");
20        printf("\n");
    }
22
24    return 0;
}
```

Rešenje (d)

```
#include <stdio.h>

2
int main()
4 {
    /* Deklaracija potrebnih promenljivih. */
6     unsigned int n, i, j;

    /* Ucitava se vrednost broja n. */
8     printf("Unesite broj n: ");
10    scanf("%u", &n);

    /* Iscrtava se trazeni trougao. */
12    for (i = 0; i < n; i++) {
14        /* Prvo se ispisuju beline koje prethode karakterima *. */
        for (j = 0; j < n - i - 1; j++)
16            printf(" ");
        /* Posle belina se ispisuje potreban broj karaktera *. */
18        for (j = 0; j <= i; j++)
            printf("*");
20        printf("\n");
    }
22
    return 0;
}
```

24 | }

*Rešenje (e)*

```

#include <stdio.h>

2 int main()
4 {
    /* Deklaracija potrebnih promenljivih. */
6     unsigned int n, i, j;

    /* Ucitava se vrednost broja n. */
8     printf("Unesite broj n: ");
10    scanf("%u", &n);

    /* Iscrtava se gornji deo trazenog trougla. */
12    for (i = 0; i < n; i++) {
        /* Prvo se ispisuju beline koje prethode karakterima *. */
14        for (j = 0; j < n - i - 1; j++)
            printf(" ");
        /* Posle belina se ispisuje potreban broj karaktera *. */
18        for (j = 0; j <= i; j++)
            printf("*");
20        printf("\n");
    }

22    /* Iscrtava se donji deo trazenog trougla. */
24    for (i = 1; i < n; i++) {
        /* Prvo se ispisuju beline koje prethode karakterima *. */
26        for (j = 0; j < i; j++)
            printf(" ");
        /* Posle belina se ispisuje potreban broj karaktera *. */
28        for (j = 0; j < n - i; j++)
            printf("*");
30        printf("\n");
    }

32    return 0;
34 }

```

*Rešenje (f)*

```

1  #include <stdio.h>

3  int main()
4  {
5      /* Deklaracije potrebnih promenljivih. */
        unsigned int n, i, j;
7      char c, novi_red;

```

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

13  /* Ucitava se karakter koji ce se koristiti za iscrtavanje.
   Napomena: voditi racuna da treba preskociti novi red koji
15  korisnik zadaje nakon unosa broja n. */
   printf("Unesite karakter c: ");
17  scanf("%c%c", &novi_red, &c);

19  /* Iscrtavanje trazenog trougla. Iscrtavaju se samo ivice
   trougla, ostalo se popunjava belinama. */
21  for (i = 0; i < n; i++) {
       for (j = 0; j <= i; j++)
23         if (i == n - 1 || j == 0 || j == i)
             printf("%c", c);
25         else
             printf(" ");
27         printf("\n");
   }
29
31  return 0;
}
```

### Rešenje 2.3.53

#### Rešenje (a)

```
1  #include <stdio.h>

3  int main()
   {
5     /* Deklaracija potrebnih promenljivih. */
     unsigned int n, i, j;

7     /* Ucitava se vrednost broja n. */
     printf("Unesite broj n: ");
     scanf("%u", &n);

11    /* Brojac i odredjuje koji red slike se trenutno ispisuje. */
13    for (i = 0; i < n; i++) {
         /* Prvo se ispisuju beline koje prethode karakterima *. */
15         for (j = 0; j < n - i - 1; j++)
             printf(" ");
17         /* Posle belina se ispisuje potreban broj karaktera *. */
         for (j = 0; j < 2 * i + 1; j++)
19             printf("*");
         printf("\n");
21     }
}
```

```
23     return 0;
    }
```

*Rešenje (b)*

```
1  #include <stdio.h>
2
3  int main()
4  {
5      /* Deklaracija potrebnih promenljivih. */
6      unsigned int n;
7      int i, j;
8
9      /* Ucitava se vrednost broja n. */
10     printf("Unesite broj n: ");
11     scanf("%u", &n);
12
13     /* Brojac i odredjuje koliko redova se ispisuje. Radi lakseg
14        izracunavanja koliko zvezdica i praznina je potrebno ispisati
15        u svakom redu, i se postavlja na n-1 i smanjuje u svakoj
16        iteraciji petlje. */
17     for (i = n - 1; i >= 0; i--) {
18         /* Prvo se ispisuju beline koje prethode karakterima *. */
19         for (j = 0; j < n - i - 1; j++)
20             printf(" ");
21         /* Posle belina se ispisuje potreban broj karaktera *. */
22         for (j = 0; j < 2 * i + 1; j++)
23             printf("*");
24         printf("\n");
25     }
26
27     return 0;
28 }
```

*Rešenje (c)*

```
1  #include <stdio.h>
2
3  int main()
4  {
5      /* Deklaracija potrebnih promenljivih. */
6      unsigned int n;
7      int i, j;
8
9      /* Ucitava se vrednost broja n. */
10     printf("Unesite broj n: ");
11     scanf("%u", &n);
12 }
```

```
14  /* Slika se crta iz dva dela. */
16  /* Brojac i odredjuje koji red slike se trenutno ispisuje. */
18  for (i = 0; i < n; i++) {
20      /* Prvo se ispisuju beline koje prethode karakterima *. */
22      for (j = 0; j < n - i - 1; j++)
24          printf(" ");
26      /* Posle belina se ispisuje potreban broj karaktera *. */
28      for (j = 0; j < 2 * i + 1; j++)
30          printf("*");
32      printf("\n");
34  }
36
38  /* II deo: crtanje donjeg trougla. Kako je prvi red donjeg
39     trougla vec ispisan (poslednji red gornjeg trougla), potrebno
40     je naciniti jednu iteraciju manje. */
41  for (i = n - 2; i >= 0; i--) {
42      /* Prvo se ispisuju beline koje prethode karakterima *. */
43      for (j = 0; j < n - i - 1; j++)
44          printf(" ");
45      /* Posle belina se ispisuje potreban broj karaktera *. */
46      for (j = 0; j < 2 * i + 1; j++)
47          printf("*");
48      printf("\n");
49  }
50
51  return 0;
52 }
```

### Rešenje (d)

```
1  #include <stdio.h>
2
3  int main()
4  {
5      /* Deklaracija potrebnih promenljivih. */
6      unsigned int n;
7      int i, j;
8
9      /* Ucitava se vrednost broja n. */
10     printf("Unesite broj n: ");
11     scanf("%u", &n);
12
13     /* Brojac i odredjuje koji red slike se trenutno ispisuje. */
14     for (i = 0; i < n; i++) {
15         /* Prvo se ispisuju beline koje prethode karakterima *. */
16         for (j = 0; j < n - i - 1; j++)
17             printf(" ");
18         /* Posle belina se ispisuje sam trougao. Ako je brojac na
19            ivici onda se ispisuje karakter *, a inace praznina.
20            Takodje, proverava se da li se ispisuje poslednji red (i==n)
21            */
22     }
```



```

    i u njemu se ispisuje svaki drugi put *, a inace praznina. */
22   for (j = 0; j < 2 * i + 1; j++)
    if (j == 0 || j == 2 * i || (i == n - 1 && j % 2 == 0))
24       printf("*");
    else
26       printf(" ");
    printf("\n");
28 }

30 return 0;
}

```

## Rešenje (c)

```

1  #include <stdio.h>

3  int main()
{
5   /* Deklaracija potrebnih promenljivih. */
   unsigned int n;
7   int i, j;

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

13  /* Brojac i odredjuje koji red slike se trenutno ispisuje. */
   for (i = 0; i < n; i++) {
15     /* Prvo se ispisuju beline koje prethode karakterima *. */
     for (j = 0; j < n - i - 1; j++)
17         printf(" ");
     /* Posle belina se ispisuje sam trougao. */
19     for (j = 0; j < 2 * i + 1; j++)
        if (j == 0 || j == 2 * i || (i == n - 1 && j % 2 == 0))
21         printf("*");
        else
23         printf(" ");
     printf("\n");
25 }

27 /* II deo: crtanje donjeg trougla. Kako je prvi red donjeg
   trougla vec ispisan (poslednji red gornjeg trougla), potrebno
29   je naciniti jednu iteraciju manje. */
   for (i = n - 2; i >= 0; i--) {
31     /* Prvo se ispisuju beline koje prethode karakterima *. */
     for (j = 0; j < n - i - 1; j++)
33         printf(" ");
     /* Posle belina se ispisuje potreban broj karaktera *. */
35     for (j = 0; j < 2 * i + 1; j++)
        if (j == 0 || j == 2 * i)
37         printf("*");

```

```
        else
39         printf(" ");
        printf("\n");
41     }

43     return 0;
}
```

### Rešenje 2.3.54

```
1  #include <stdio.h>

3  int main()
{
5     /* Deklaracija potrebnih promenljivih. */
    unsigned int n;
7     int i, j;

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

13    /* Strelica se moze posmatrati kao spojena dva pravougla trougla
        kojima se ispisuje hipotenuza i jedna kateta. */

15    /* Brojac i odredjuje koji red slike se trenutno ispisuje. */
17    for (i = 0; i < n; i++) {
        for (j = 0; j <= i; j++)
19            /* Proverava se da li se ispisuje karakter na hipotenuzi
                (j == i) ili da se ispisuje poslednji red (i == n-1). */
21            if (j == i || i == n - 1)
                printf("*");
            else
23                printf(" ");
25        printf("\n");
    }

27    /* II deo: crtanje donjeg dela slike, odnosno donji trougao.
        Brojac i odredjuje koji red donjeg trougla se trenutno iscrtava.
        Kako je prvi red donjeg trougla vec iscrtan (to je poslednji
        red gornjeg trougla), brojac se postavlja na 1. */
31    for (i = 1; i < n; i++) {
        for (j = 0; j < n - i; j++)
33            /* Provera da li se ispisuje hipotenuza. */
35            if (j == n - i - 1)
                printf("*");
            else
37                printf(" ");
39        printf("\n");
    }

41 }
```

```
43     return 0;
    }
```

### Rešenje 2.3.55

```
1  #include <stdio.h>
3  int main()
4  {
5      /* Deklaracija potrebnih promenljivih. */
6      unsigned int n;
7      int i, j, k;
8
9      /* Ucitava se vrednost broja n. */
10     printf("Unesite broj n: ");
11     scanf("%u", &n);
12
13     /* Brojac j odredjuje koliko ukupno karaktera (praznina i
14     karakteri *) u svakom redu se ispisuje. U svakom drugom redu
15     ovaj broj se povecava za 2. Na pocetku je 1 (jer se ispisuje
16     samo jedna zvezda). */
17     j = 1;
18
19     /* Brojac i odredjuje koji red slike se trenutno ispisuje. */
20     for (i = 1; i <= n; i++) {
21         /* U svakom drugom redu broj karaktera koji treba da se
22         ispisu se uvecava za 2. */
23         if (i % 2 == 0)
24             j += 2;
25
26         /* Ispisuje se j karaktera. */
27         for (k = 0; k < j; k++)
28             /* U svakom parnom redu se naizmenicno
29             ispisuju * i praznina. */
30             if (i % 2 == 0) {
31                 if (k % 2 == 0)
32                     printf("*");
33                 else
34                     printf(" ");
35             }
36             else
37             {
38                 /*U svakom neparnom redu se ispisuju samo *. */
39                 printf("*");
40             }
41         printf("\n");
42     }
43
44     return 0;
45 }
```

### Rešenje 2.3.56

```
2  #include <stdio.h>
3
4  int main()
5  {
6      /* Deklaracija potrebnih promenljivih. */
7      unsigned int n, m;
8      int i, j;
9
10     /* Ucitavaju se dimenzije slike. */
11     printf("Unesite brojeve n i m: ");
12     scanf("%u%u", &n, &m);
13
14     /* Brojac i odredjuje koji red slike se trenutno ispisuje.
15        Ukupno ima m redova. */
16     for (i = 1; i <= m; i++) {
17         /* Brojac j oznacava koja kolona se trenutno ispisuje.
18            Za svaki kvadrat se racuna duzina bez poslednje ivice.
19            Kvadrat je sastavljen od (m-1) zvezdice i (m-1) praznine
20            (praznine se nalaze izmedju zvezdica). Znacni ukupna duzina
21            je 2*(m-1) karakter, a kako ima n kvadrata plus jedna kolona
22            za najdesniju ivicu, duzina je n*2*(m-1) + 1. */
23         for (j = 0; j <= n * 2 * (m - 1); j++)
24             /* Provera da li se ispisuje prvi ili poslednji red. */
25             if (i == 1 || i == m)
26                 /* Naizmenicno se ispisuje * i praznina. */
27                 if (j % 2 == 0)
28                     printf("*");
29                 else
30                     printf(" ");
31             else
32                 /* Na ivicama kvadrata se iscrtavaju * a na ostalim mestima
33                    beline. */
34                 if (j % (2 * (m - 1)) == 0)
35                     printf("*");
36                 else
37                     printf(" ");
38         printf("\n");
39     }
40
41     return 0;
42 }
```

### Rešenje 2.3.57

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

```

5  {
6  /* Deklaracija potrebnih promenljivih. */
7  unsigned int n;
8  int i, j;
9
10 /* Ucitava se vrednost broja n. */
11 printf("Unesite broj n: ");
12 scanf("%u", &n);
13
14 /* Romb se crta crtanjem dva spojena trougla koji se nezavisno
15    iscrtavaju. */
16
17 /* Brojac i odredjuje koji red slike se trenutno ispisuje. */
18 for (i = 0; i < n; i++) {
19     /* Prvo se ispisuju * koje prethode karakterima -. */
20     for (j = 0; j < n - i; j++)
21         printf("*");
22     /* Potom se ispisuju karakteri -. */
23     for (j = 0; j < 2 * i; j++)
24         printf("-");
25     /* Potom se ispisuju * koje su nakon karaktera -. */
26     for (j = 0; j < n - i; j++)
27         printf("*");
28     printf("\n");
29 }
30
31 /* II deo: crtanje donjeg trougla. Kako je prvi red donjeg
32    trougla vec ispisan (poslednji red gornjeg trougla), potrebno
33    je naciniti jednu iteraciju manje. */
34 for (i = n - 2; i >= 0; i--) {
35     /* Prvo se ispisuju * koje prethode karakterima -. */
36     for (j = 0; j < n - i; j++)
37         printf("*");
38     /* Potom se ispisuju karakteri -. */
39     for (j = 0; j < 2 * i; j++)
40         printf("-");
41     /* Potom se ispisuju * koje su nakon karaktera -. */
42     for (j = 0; j < n - i; j++)
43         printf("*");
44     printf("\n");
45 }
46
47 return 0;
48 }

```

### Rešenje 2.3.58

```

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

```

## 2 Kontrola toka

---

```
5  /* Deklaracija potrebnih promenljivih. */
   unsigned int n;
7  int i, j;

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

13 /* Slika se sastoji iz dva dela, trougla i kvadrata i svaki deo
   se nezavisno iscrtava. */

15
17 /* I deo: crtanje trougla (krova). */
   for (i = 0; i < n - 1; i++) {
       /* Prvo se ispisuju beline koje prethode karakterima *. */
19       for (j = 0; j < n - i - 1; j++)
           printf(" ");

21       /* Posle belina se ispisuje sam trougao.*/
23       for (j = 0; j < 2 * i + 1; j++)
           if (j == 0 || j == 2 * i)
25               printf("*");
           else
27               printf(" ");
           printf("\n");
29   }

31 /* II deo: crtanje kvadrata. Da bi iscrtavanje bilo lakse
   istovremeno se ispisuju dva karaktera. */
33   for (i = 0; i < n; i++) {
       for (j = 0; j < n; j++)
35           /* Provera da li je ivica. */
           if (j == 0 || j == n - 1 || i == 0 || i == n - 1)
37               printf("* ");
           else
39               printf(" ");
           printf("\n");
41   }

43   return 0;
}
```

### Rešenje 2.3.59

```
1  #include <stdio.h>

3  int main()
   {
5      /* Deklaracija potrebnih promenljivih. */
       unsigned int n;
7       int i, j;
```

```

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

13  /* Prva petlja oznacava broj 'serija' koje ce se ispisati.
   Na primer, za n=5, prva serija je 1 2 3 4 5, druga serija je
15  2 3 4 i treca serija je 3.
   Kako se u svakoj sledecoj seriji broj brojeva smanjuje za 2,
17  do 0 karaktera u seriji se dolazi posle n/2 koraka, ali
   zaokruženo navise (5/2 = 2.5 --> 3), a to je isto sto i
19  celobrojno (n+1)/2. */
   for (i = 1; i <= (n + 1) / 2; i++) {
21     /* U svakoj seriji se ispisuju brojevi izmedju i i n-i+1. */
     for (j = i; j <= n + 1 - i; j++)
23         printf("%d ", j);
   }
25
   return 0;
27 }

```

### Rešenje 2.3.60

```

1  #include <stdio.h>

3  int main()
   {
5     /* Deklaracija potrebnih promenljivih. */
     unsigned int n;
     int i, j;

7

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

13    /* Brojac i je redni broj vrste koja se ispisuje. */
     for (i = 1; i <= n; i++) {
15         /* U svakoj vrsti se ispisuju brojevi izmedju 1 i n,
           sa korakom i. */
17         for (j = 1; j <= n; j+=i)
             printf("%d ", j);

19         printf("\n");
21     }

23     return 0;
   }

```

## 2.5 Funkcije

**Zadatak 2.5.1** Napisati funkciju `int min(int x, int y, int z)` koja izračunava minimum tri broja. Napisati program koji učitava tri cela broja i ispisuje njihov minimum.

*Primer 1*

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesite brojeve: 19 8 14  
|| Minimum: 8
```

*Primer 2*

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesite brojeve: -6 11 -12  
|| Minimum: -12
```

[Rešenje 2.5.1]

**Zadatak 2.5.2** Napisati funkciju `float razlomljeni_deo(float x)` koja izračunava razlomljeni deo broja  $x$ . Napisati program koji učitava jedan realan broj i ispisuje njegov razlomljeni deo.

*Primer 1*

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesite broj: 8.235  
|| Razlomljeni deo: 0.235000
```

*Primer 2*

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesite broj: -5.11  
|| Razlomljeni deo: 0.110000
```

[Rešenje 2.5.2]

**Zadatak 2.5.3** Napisati funkciju `int zbir_delilaca(int n)` koja izračunava zbir delilaca broja  $n$ . Napisati program koji učitava ceo pozitivan broj  $k$  i ispisuje zbir delilaca svakog broja od 1 do  $k$ . U slučaju neispravnog unosa, ispisati odgovarajuću poruku o grešci.

*Primer 1*

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesite broj k: 6  
|| 1 3 4 7 6 12
```

*Primer 2*

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesite broj k: -2  
|| Greska: neispravan unos.
```

[Rešenje 2.5.3]

**Zadatak 2.5.4** Napisati funkciju `int je_stepen(unsigned x, unsigned n)` koja za dva broja  $x$  i  $n$  utvrđuje da li je  $x$  neki stepen broja  $n$ . Ukoliko jeste, funkcija vraća izložilac stepena, a u suprotnom vraća  $-1$ . Napisati program koji učitava dva neoznačena broja i ispisuje da li vrednost prvog broja odgovara vrednosti nekog stepena drugog broja.



*Primer 1*

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite dva broja: 81 3
|| Jeste: 81 = 34
```

*Primer 2*

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite dva broja: 162 11
|| Broj 162 nije stepen broja 11.
```

[Rešenje 2.5.4]

**Zadatak 2.5.5** Napisati funkciju `int euklid(int x, int y)` koja za dva data cela broja određuje najvećeg zajedničkog delioca primenom Euklidovog algoritma. Napisati program koji učitava dva cela broja i ispisuje vrednost njihovog najvećeg zajedničkog delioca.

*Primer 1*

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite dva cela broja: 1024 832
|| Najveci zajednicki delilac: 64
```

*Primer 2*

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite dva cela broja: -900 112
|| Najveci zajednicki delilac: 4
```

[Rešenje 2.5.5]

**Zadatak 2.5.6** Napisati funkciju `float zbir_reciprocnih(int n)` koja za dato  $n$  vraća zbir recipročnih vrednosti brojeva od 1 do  $n$ . Napisati program koji učitava ceo pozitivan broj  $n$  i ispisuje odgovarajući zbir zaokružen na dve decimalne. U slučaju neispravnog unosa, ispisati odgovarajuću poruku o grešci.

*Primer 1*

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite broj n: 10
|| Zbir reciprocnih: 2.93
```

*Primer 2*

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite broj n: 100
|| Zbir reciprocnih: 5.19
```

*Primer 3*

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite broj n: -100
|| Greska: neispravan unos.
```

[Rešenje 2.5.6]

**Zadatak 2.5.7** Napisati funkciju `int prebrojavanje(float x)` koja prebrojava koliko puta se broj  $x$  pojavljuje u nizu brojeva koji se unose sve do unosa broja nula. Napisati program koji učitava vrednost broja  $x$  i ispisuje koliko puta se njegova vrednost pojavila u unetom nizu.

### Primer 1

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesite broj x: 2.84  
|| Unesite brojeve:  
|| 8.13 2.84 5 21.6 2.84 11.5 0  
|| Broj pojavljivanja broja 2.84: 2
```

### Primer 2

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesite broj x: -1.17  
|| Unesite brojeve:  
|| -128.35 8.965 8.968 89.36 0  
|| Broj pojavljivanja broja -1.17: 0
```

[Rešenje 2.5.7]

**Zadatak 2.5.8** Broj je prost ako je deljiv samo sa 1 i sa samim sobom.

- Napisati funkciju `int prost(int x)` koja ispituje da li je dati ceo broj prost. Funkcija treba da vrati jedinicu ako je broj prost i nulu u suprotnom.
- Napisati funkciju `void prvih_n_prostih(int n)` koja ispisuje prvih  $n$  prostih brojeva.
- Napisati funkciju `void prosti_brojevi_manji_od_n(int n)` koja ispisuje sve proste brojeve manje od broja  $n$ .

Napisati program koji učitava pozitivan ceo broj  $n$  i ispisuje prvih  $n$  prostih brojeva, kao i sve proste brojeve manje od  $n$ . U slučaju neispravnog unosa, ispisati odgovarajuću poruku o grešci.

### Primer 1

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesite broj n: 5  
|| Prvih n prostih: 2 3 5 7 11  
|| Prosti manji od n: 2 3
```

### Primer 2

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesite broj n: 2  
|| Prvih n prostih: 2 3  
|| Prosti manji od n: ne postoje
```

### Primer 3

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesite broj n: -11  
|| Greska: neispravan unos.
```

[Rešenje 2.5.8]

**Zadatak 2.5.9** Rešiti sledeće zadatke korišćenjem funkcija.

- Zadatak 1.1.2 rešiti korišćenjem funkcija `int kvadrat(int x)` koja računa kvadrat datog broja i `int kub(int x)` koja računa kub datog broja.
- Zadatak 2.1.2 rešiti korišćenjem funkcije `float apsolutna_vrednost(float x)` koja izračunava apsolutnu vrednost datog broja.

- c) Zadatak 2.3.7 rešiti korišćenjem funkcije `float stepen(float x, int n)` koja računa vrednost  $n$ -tog stepena realnog broja  $x$ .
- d) Zadatak 2.3.29 rešiti korišćenjem funkcije `int fibonaci(int n)` koja računa  $n$ -ti element Fibonačijevog niza.

**Zadatak 2.5.10** Napisati funkciju `float aritmeticka_sredina(int n)` koja računa aritmetičku sredinu cifara datog broja. Napisati i program koji učitava ceo broj i ispisuje aritmetičku sredinju njegovih cifara zaokruženu na tri decimalne.

*Primer 1*

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite broj: 461
|| 3.667
```

*Primer 2*

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite broj: 1001
|| 0.500
```

*Primer 3*

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite broj: -84723
|| 4.800
```

[Rešenje 2.5.10]

**Zadatak 2.5.11** Napisati funkciju `int sadrzi(int x, int c)` koja ispituje da li se cifra  $c$  nalazi u zapisu celog broja  $x$ . Funkcija treba da vrati jedinicu ako se cifra nalazi u broju, a nulu inače. Napisati program koji učitava jedan ceo broj i jednu cifru i u zavisnosti od toga da li se uneta cifra nalazi u zapisu unetog broja, ispisuje odgovarajuću poruku. U slučaju neispravnog unosa, ispisati odgovarajuću poruku o grešci.

*Primer 1*

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite broj i cifru: 17890 7
|| Cifra 7 se nalazi u zapisu broja 17890.
```

*Primer 2*

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite broj i cifru: 19 6
|| Cifra 6 se ne nalazi u zapisu broja 19.
```

*Primer 3*

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite broj i cifru: 17890 26
|| Greska: neispravan unos.
```

*Primer 4*

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite broj i cifru: -1982 9
|| Cifra 9 se nalazi u zapisu broja -1982.
```

[Rešenje 2.5.11]

**Zadatak 2.5.12** Napisati funkciju `int broj_neparnih_cifara(int x)` koja određuje broj neparnih cifara u zapisu datog celog broja. Napisati program koji učitava cele brojeve sve do unosa broja nula i ispisuje broj neparnih cifara svakog unetog broja.

### Primer 1

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite cele brojeve:
|| 2341
|| Broj neparnih cifara: 2
|| 78
|| Broj neparnih cifara: 1
|| 800
|| Broj neparnih cifara: 0
|| -99761
|| Broj neparnih cifara: 4
|| 0
```

### Primer 2

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite cele brojeve:
|| 987611
|| Broj neparnih cifara: 4
|| 135
|| Broj neparnih cifara: 3
|| -701
|| Broj neparnih cifara: 2
|| 602
|| Broj neparnih cifara: 0
|| -884
|| Broj neparnih cifara: 0
|| 79901
|| Broj neparnih cifara: 4
|| 0
```

[Rešenje 2.5.12]

**Zadatak 2.5.13** Napisati program za ispitivanje svojstava cifara datog celog broja.

- (a) Napisati funkciju `int sve_parne_cifre(int x)` koja ispituje da li se dati ceo broj sastoji isključivo iz parnih cifara. Funkcija treba da vrati jedinicu ako su sve cifre broja parne, a nulu inače.
- (b) Napisati funkciju `int sve_cifre_jednake(int x)` koja ispituje da li su sve cifre datog celog broja jednake. Funkcija treba da vrati jedinicu ako su sve cifre broja jednake, a nulu inače.

Program učitava ceo broj `i` u zavisnosti od toga da li su navedena svojstva ispunjena ili ne, ispisuje odgovarajuću poruku.

### Primer 1

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite broj: 86422
|| Sve cifre broja su parne.
|| Cifre broja nisu jednake.
```

### Primer 2

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite broj: 55555
|| Broj sadrzi bar jednu neparnu cifru.
|| Cifre broja su jednake.
```

### Primer 3

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite broj: -88
|| Sve cifre broja su parne.
|| Cifre broja su jednake.
```

### Primer 4

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite broj i cifru: -342
|| Broj sadrzi bar jednu neparnu cifru.
|| Cifre broja nisu jednake.
```

[Rešenje 2.5.13]

**Zadatak 2.5.14** Napisati funkciju `int ukloni(int n, int p)` koja menja broj  $n$  tako što iz njegovog zapisa uklanja cifru na poziciji  $p$ . Pozicije se broje sa desna na levo. Cifra jedinica ima poziciju 1. Napisati program koji učitava redni broj pozicije  $i$  zatim za cele brojeve koji se unose sve do unosa broja nula, ispisuje brojeve kojima je uklonjena cifra na poziciji  $p$ . U slučaju neispravnog unosa, ispisati odgovarajuću poruku o grešci.

*Primer 1*

```
INTERAKCIJA SA PROGRAMOM:
Unesite poziciju: 3
Unesite broj: 1210
110
Unesite broj: 18
18
Unesite broj: 3856
356
Unesite broj: 0
```

*Primer 2*

```
INTERAKCIJA SA PROGRAMOM:
Unesite poziciju: 1
Unesite broj: -9632
-963
Unesite broj: -2
0
Unesite broj: 246
24
Unesite broj: -52
-5
Unesite broj: 0
```

*Primer 3*

```
INTERAKCIJA SA PROGRAMOM:
Unesite poziciju: 0
Greska: neispravan unos.
```

[Rešenje 2.5.14]

**Zadatak 2.5.15** Napisati funkciju `int zapis(int x, int y)` koja proverava da li se brojevi  $x$  i  $y$  zapisuju pomoću istih cifara. Funkcija treba da vrati jedinicu ako je uslov ispunjen, a nulu inače. Napisati program koji učitava dva cela broja i ispisuje da li je za njih pomenuti uslov ispunjen ili ne.

*Primer 1*

```
INTERAKCIJA SA PROGRAMOM:
Unesite dva broja: 251 125
Uslov je ispunjen.
```

*Primer 2*

```
INTERAKCIJA SA PROGRAMOM:
Unesite dva broja: 8898 9988
Uslov nije ispunjen.
```

*Primer 3*

```
INTERAKCIJA SA PROGRAMOM:
Unesite dva broja: -7391 1397
Uslov je ispunjen.
```

*Primer 4*

```
INTERAKCIJA SA PROGRAMOM:
Unesite dva broja: -777 77
Uslov nije ispunjen.
```

[Rešenje 2.5.15]

**Zadatak 2.5.16** Napisati funkciju `int neopadajuce(int n)` koja ispituje da li su cifre datog celog broja u neopadajućem poretku. Funkcija treba da vrati jedinicu ako cifre ispunjavaju uslov, a nulu inače. Napisati program koji učitava ceo broj  $i$  i ispisuje poruku da li su cifre unetog broja u neopadajućem poretku.

### Primer 1

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesite broj: 2289  
|| Cifre su u neopadajućem poretku.
```

### Primer 2

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesite broj: 5  
|| Cifre su u neopadajućem poretku.
```

### Primer 3

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesite broj: 6628  
|| Cifre nisu u neopadajućem poretku.
```

### Primer 4

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesite broj: -23  
|| Cifre su u neopadajućem poretku.
```

[Rešenje 2.5.16]

**Zadatak 2.5.17** Napisati funkciju `int par_nepar(int n)` koja ispituje da li su cifre datog celog broja naizmenično parne i neparne. Funkcija treba da vrati jedinicu ako cifre ispunjavaju uslov, a nulu inače. Napisati program koji učitava ceo broj i ispisuje da li on ispunjava pomenuti uslov ili ne.

### Primer 1

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesite broj n: 2749  
|| Broj ispunjava uslov.
```

### Primer 2

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesite broj n: -963  
|| Broj ispunjava uslov.
```

### Primer 3

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesite broj n: 27449  
|| Broj ne ispunjava uslov.
```

### Primer 4

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesite broj n: 5  
|| Broj ispunjava uslov.
```

[Rešenje 2.5.17]

**Zadatak 2.5.18** Napisati funkciju `int rotacija(int n)` koja rotira cifre zadatog celog broja za jednu poziciju u levo. Napisati program koji za brojeve koji se unose sve do unosa broja nula ispisuje odgovarajuće rotirane brojeve.

### Primer 1

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesite broj: 146  
|| 461  
|| Unesite broj: 18  
|| 81  
|| Unesite broj: 3856  
|| 8563  
|| Unesite broj: 7  
|| 7  
|| Unesite broj: 0
```

### Primer 2

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesite broj: 89  
|| 98  
|| Unesite broj: -369  
|| -693  
|| Unesite broj: -55281  
|| -52815  
|| Unesite broj: 0
```

[Rešenje 2.5.18]

**Zadatak 2.5.19** Za dati broj može se formirati niz tako da je svaki sledeći član niza dobijen kao suma cifara prethodnog člana niza. Broj je *srećan* ako se dati niz završava jedinicom. Napisati funkciju `int srećan(int x)` koja vraća jedinicu ako je broj srećan, a nulu inače. Napisati program koji za uneti pozitivan ceo broj  $n$  ispisuje sve srećne brojeve od 1 do  $n$ . U slučaju neispravnog unosa, ispisati odgovarajuću poruku o grešci.

*Primer 1*

```
INTERAKCIJA SA PROGRAMOM:
Unesite broj n: 100
Srećni brojevi:
1 10 19 28 37 46 55 64 73 82 91 100
```

*Primer 2*

```
INTERAKCIJA SA PROGRAMOM:
Unesite broj n: 0
Greska: neispravan unos.
```

[Rešenje 2.5.19]

**Zadatak 2.5.20** Prirodan broj  $a$  je Armstrongov ako je jednak sumi  $n$ -tih stepena svojih cifara, pri čemu je  $n$  broj cifara broja  $a$ . Napisati funkciju `int armstrong(int x)` koja vraća jedinicu ako je broj Armstrongov, a nulu inače. Napisati program koji za učitani pozitivan ceo broj proverava da li je Armstrongov. U slučaju neispravnog unosa, ispisati odgovarajuću poruku o grešci.

*Primer 1*

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

*Primer 2*

```
INTERAKCIJA SA PROGRAMOM:
Unesite broj: 1634
Broj je Armstrongov.
```

*Primer 3*

```
INTERAKCIJA SA PROGRAMOM:
Unesite broj: 118
Broj nije Armstrongov.
```

*Primer 4*

```
INTERAKCIJA SA PROGRAMOM:
Unesite broj: 0
Greska: neispravan unos.
```

[Rešenje 2.5.20]

**Zadatak 2.5.21** Napisati funkciju `double e_na_x(double x, double eps)` koja računa vrednost  $e^x$  kao parcijalnu sumu reda  $\sum_{n=0}^{\infty} \frac{x^n}{n!}$ , pri čemu se sumiranje vrši dok je razlika sabiraka u redu po apsolutnoj vrednosti manja od  $\varepsilon$ . Napisati program koji učitava dva realna broja  $x$  i  $eps$  i ispisuje izračunatu vrednost  $e^x$ .

### Primer 1

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesite broj x: 5  
|| Unesite eps: 0.001  
|| Rezultat: 148.412951
```

### Primer 2

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesite broj x: -3  
|| Unesite eps: 0.0001  
|| Rezultat: 0.049796
```

[Rešenje 2.5.21]

**Zadatak 2.5.22** Napisati funkciju `void ispis(float x, float y, int n)` koja za dva realna broja  $x$  i  $y$  i jedan pozitivan ceo broj  $n$  ispisuje vrednosti sinusne funkcije u  $n$  ravnomerno raspoređenih tačaka intervala  $[x, y]$ . Napisati program koji učitava granice intervala i broj tačaka i ispisuje odgovarajuće vrednosti sinusne funkcije, zaokružene na četiri decimale. U slučaju neispravnog unosa, ispisati odgovarajuću poruku o grešci.

### Primer 1

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesite dva realna broja: 7 32  
|| Unesite broj n: 10  
|| 0.6570 -0.3457 -0.0108 0.3659 -0.6731  
|| 0.8922 -0.9945 0.9666 -0.8122
```

### Primer 2

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesite dva realna broja: 20.5 -8.32  
|| Unesite broj n: 5  
|| -0.8934 -0.8979 -0.1920 0.6658 0.9968
```

### Primer 3

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesite dva realna broja: 8 8  
|| Greska: neispravan unos.
```

### Primer 1

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesite dva realna broja: 7 32  
|| Unesite broj n: -10  
|| Greska: neispravan unos.
```

[Rešenje 2.5.22]

**Zadatak 2.5.23** Napisati funkciju `char sifra(char c, int k)` koja za dati karakter  $c$  određuje šifru na sledeći način: ukoliko je  $c$  slovo, šifra je karakter koji se nalazi  $k$  pozicija pre njega u abecedi. Karakteri koji nisu slova se ne šifruju. Šifrovanje treba da bude kružno, što znači da je, na primer, šifra za karakter  $b$  i pomeraj 2 karakter  $z$ . Napisati program koji učitava nenegativan ceo broj  $k$ , a zatim i karaktere sve do kraja ulaza i nakon svakog učitanoog karaktera ispisuje njegovu šifru. U slučaju neispravnog unosa, ispisati odgovarajuću poruku o grešci.



*Primer 1*

```

INTERAKCIJA SA PROGRAMOM:
Unesite broj k: 2
Unesite tekst (CTRL+D za prekid):
c
a
8
8
+
+
Z
X

```

*Primer 2*

```

INTERAKCIJA SA PROGRAMOM:
Unesite broj k: -2
Greska: neispravan unos.

```

[Rešenje 2.5.23]

**Zadatak 2.5.24** Rešiti sledeće zadatke korišćenjem funkcija.

- Zadatak 2.3.32 rešiti korišćenjem funkcije `char konverzija(char c)` koja malo slovo pretvara u odgovarajuće veliko i obrnuto.
- Zadatak 2.3.33 rešiti korišćenjem funkcije `void prebrojavanje()` koja učitava karaktere sve do kraja ulaza i ispisuje broj malih slova, velikih slova, cifara, belina, kao i sumu svih unetih cifara.

**Zadatak 2.5.25** Napisati program koji učitava tri cela broja i ispisuje datum sledećeg dana. Zadatak rešiti korišćenjem narednih funkcija.

- `int prestupna(int godina)` koja za zadatu godinu proverava da li je prestupna. Funkcija treba da vrati jedinicu ako je godina prestupna ili nulu ako nije.
- `int broj_dana(int mesec, int godina)` koja za dati mesec i godinu vraća broj dana u datom mesecu.
- `int ispravan(int dan, int mesec, int godina)` koja za dati datum proverava da li je ispravan.
- `void sledeci_dan(int dan, int mesec, int godina)` koja za dati datum ispisuje datum sledećeg dana.

U slučaju neispravnog unosa, ispisati odgovarajuću poruku o grešci.

*Primer 1*

```

INTERAKCIJA SA PROGRAMOM:
Unesite datum: 24.8.1998.
Datum sledeceg dana je: 25.8.1998.

```

*Primer 2*

```

INTERAKCIJA SA PROGRAMOM:
Unesite datum: 31.12.1789.
Datum sledeceg dana je: 1.1.1790.

```

### Primer 3

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesite datum: 28.2.2003.  
|| Datum sledeceg dana je: 1.3.2004.
```

### Primer 4

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesite datum: 31.4.2004.  
|| Greska: neispravan unos.
```

[Rešenje 2.5.25]

**Zadatak 2.5.26** Napisati funkciju `int od_nove_godine(int dan, int mesec, int godina)` koja određuje koliko je dana proteklo od Nove godine do datog datuma. Napisati program koji učitava tri cela broja i ispisuje koliko dana je proteklo od Nove godine. U slučaju neispravnog unosa, ispisati odgovarajuću poruku o grešci.

### Primer 1

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesite datum: 24.8.1998.  
|| Broj dana od Nove godine je: 235
```

### Primer 2

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesite datum: 31.12.1680.  
|| Broj dana od Nove godine je: 366
```

### Primer 3

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesite datum: 28.2.2003.  
|| Broj dana od Nove godine je: 58
```

### Primer 4

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesite datum: 31.4.2004.  
|| Greska: neispravan unos.
```

[Rešenje 2.5.26]

**Zadatak 2.5.27** Napisati funkciju `int do_kraja_godine(int dan, int mesec, int godina)` koja određuje broj dana od datog datuma do kraja godine. Napisati program koji učitava tri cela broja i ispisuje broj dana do kraja godine. U slučaju neispravnog unosa, ispisati odgovarajuću poruku o grešci.

### Primer 1

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesite datum: 24.8.1998.  
|| Broj dana do Nove godine je: 129
```

### Primer 2

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesite datum: 31.12.1680.  
|| Broj dana od Nove godine je: 0
```

### Primer 3

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesite datum: 28.2.2004.  
|| Broj dana od Nove godine je: 307
```

### Primer 4

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesite datum: 31.4.2004.  
|| Greska: neispravan unos.
```

[Rešenje 2.5.27]

**Zadatak 2.5.28** Napisati funkciju `int broj_dana_izmedju(int dan1, int mesec1, int godina1, int dan2, int mesec2, int godina2)` koja određuje broj dana između dva datuma. Napisati program koji učitava šest celih brojeva, koji označavaju dva datuma, i na standardni izlaz ispisuje broj dana između ta dva datuma. U slučaju neispravnog unosa, ispisati odgovarajuću poruku o grešci.

*Primer 1*

```
INTERAKCIJA SA PROGRAMOM:
Unesite prvi datum: 12.3.2008.
Unesite drugi datum: 5.12.2008.
Broj dana izmedju dva datuma je: 268
```

*Primer 2*

```
INTERAKCIJA SA PROGRAMOM:
Unesite prvi datum: 26.9.1986.
Unesite drugi datum: 2.2.1701.
Broj dana izmedju dva datuma je: 104301
```

*Primer 3*

```
INTERAKCIJA SA PROGRAMOM:
Unesite prvi datum: 24.8.1998.
Unesite drugi datum: 12.10.2010.
Broj dana izmedju dva datuma je: 4440
```

*Primer 4*

```
INTERAKCIJA SA PROGRAMOM:
Unesite prvi datum: 24.8.1998.
Unesite drugi datum: 31.4.2004.
Greska: neispravan unos.
```

[Rešenje 2.5.28]

**Zadatak 2.5.29** Napisati funkciju `void romb(int n)` koja iscrtava romb čija je stranica dužine  $n$ . Napisati program koji učitava ceo pozitivan broj i iscrtava odgovarajuću sliku. U slučaju neispravnog unosa, ispisati odgovarajuću poruku o grešci.

*Primer 1*

```
INTERAKCIJA SA PROGRAMOM:
Unesite broj n: 5
*****
*****
*****
*****
*****
```

*Primer 2*

```
INTERAKCIJA SA PROGRAMOM:
Unesite broj n: 2
**
**
```

*Primer 3*

```
INTERAKCIJA SA PROGRAMOM:
Unesite broj n: -5
Greska: neispravan unos.
```

[Rešenje 2.5.29]

**Zadatak 2.5.30** Napisati funkciju `void grafikon_h(int a, int b, int c, int d)` koja iscrtava horizontalni prikaz zadatih vrednosti. Napisati program koji učitava četiri pozitivna cela broja i iscrtava odgovarajuću sliku. U slučaju neispravnog unosa, ispisati odgovarajuću poruku o grešci.

*Primer 1*

```
INTERAKCIJA SA PROGRAMOM:
Unesite brojeve: 4 1 7 5
****
*
*****
****
```

*Primer 2*

```
INTERAKCIJA SA PROGRAMOM:
Unesite brojeve: 5 2 2 10
*****
**
**
*****
```

*Primer 3*

```
INTERAKCIJA SA PROGRAMOM:
Unesite brojeve: 8 -2 5 4
Greska: neispravan unos.
```

[Rešenje 2.5.30]

**Zadatak 2.5.31** Napisati funkciju `void grafikon_v(int a, int b, int c, int d)` koja iscrtava vertikalni prikaz zadatih vrednosti. Napisati program koji učitava četiri pozitivna cela broja i iscrtava odgovarajuću sliku. U slučaju neispravnog unosa, ispisati odgovarajuću poruku o grešci.

*Primer 1*

```
INTERAKCIJA SA PROGRAMOM:
Unesite brojeve: 4 1 7 5
*
*
**
* **
* **
* **
****
```

*Primer 2*

```
INTERAKCIJA SA PROGRAMOM:
Unesite brojeve: 5 2 2 4
*
* *
* *
****
****
```

*Primer 3*

```
INTERAKCIJA SA PROGRAMOM:
Unesite brojeve: 8 -2 5 4
Greska: neispravan unos.
```

[Rešenje 2.5.31]

## 2.6 Rešenja

### Rešenje 2.5.1

```
1 #include <stdio.h>
3 /* Funkcija koja racuna minimum tri cela broja.
   Promenljive u listi argumenata funkcije (x, y i z), kao i one
   deklarisane u samoj funkciji (minimum), lokalne su za tu
   funkciju, sto znaci da im se ne moze pristupiti nigde izvan
   funkcije min. */
7 int min(int x, int y, int z)
9 {
   /* Vrednost minimuma se postavlja na vrednost broja x. */
11  int minimum = x;
```

```

13  /* Vrsi se poredjenje sa druga dva broja i po potrebi
    azuriranje vrednosti minimuma. */
15  if (minimum > y)
        minimum = y;
17  if (minimum > z)
        minimum = z;
19
    /* Vrednost minimuma se vraca kao povratna vrednost funkcije. */
21  return minimum;
    }
23
int main()
25 {
    /* Deklaracije potrebnih promenljivih. */
27  int x, y, z;

    /* Ucitavaju se vrednosti tri broja. */
29  printf("Unesite brojeve: ");
    scanf("%d%d%d", &x, &y, &z);
31

    /* Poziv funkcije i ispis rezultata. */
33  printf("Minimum: %d\n", min(x, y, z));
35
    return 0;
37 }

```

### Rešenje 2.5.2

```

1  #include <stdio.h>
    #include <math.h>
3
    /* Funkcija koja vraca razlomljeni deo prosledjenog broja. */
5  float razlomljeni_deo(float x)
    {
7      /* Napomena: Funkcija fabs racuna apsolutnu vrednost realnog
        broja i njena deklaracija se nalazi u zaglavlju math.h.
        Funkcija abs racuna apsolutnu vrednost celog broja i njena
        deklaracija se nalazi u zaglavlju stdlib.h. */
11     x = fabs(x);

13     /* Razlomljeni deo broja se dobija tako sto se od samog broja
        oduzme njegov ceo deo. */
15     return x - (int) x;
    }
17
int main()
19 {
    /* Deklaracija potrebnih promenljivih. */
21  float n;

```

```
23  /* Ucitava se ulazna vrednost. */
    printf("Unesite broj:");
25  scanf("%f", &n);

27  /* Ispis rezultata. */
    printf("Razlomljeni deo: %.6f\n", razlomljeni_deo(n));
29
    return 0;
31 }
```

### Rešenje 2.5.3

```
1  #include <stdio.h>

3  /* Funkcija koja racuna zbir delilaca broja x. */
    int zbir_delilaca(int x) {
5      int i;

7      /* Inicijalizacija zbira na 0. */
        int zbir = 0;

9
11     /* Svaki broj i izmedju 1 i sqrt(x) koji deli broj x se dodaje
        u zbir. Ako je u pitanju broj koji za koji vazi da je i*i
        jednako x, onda se dodaje samo vrednost i, a ako nije, onda
13     se pored vrednosti i dodaje i x/i.
        Na primer, za x=6, kada je i=2, dodaju se i 2 i 6/2 = 3, a za
15     x = 4, kada je i=2, dodaje se samo 2. */
        for (i = 1; i*i <= x; i++) {
17             if (x % i == 0)
                {
19                 zbir += i;
                    if(i != x/i)
21                     zbir += x/i;
                }
23     }

25     /* Povratna vrednost funkcije je dobijeni zbir. */
        return zbir;
27 }

29 int main()
    {
31     /* Deklaracija potrebnih promenljivih. */
        int k, i;

33
35     /* Ucitava se broj k. */
        printf("Unesite broj k:");
        scanf("%d", &k);

37
39     /* Vrsi se provera ispravnosti ulaznih podataka. */
        if (k <= 0){
```

```

41     printf("Greska: neispravan unos.\n");
    return -1;
}

43
44 /* Za svaki broj od 1 do k se ispisuje zbir delilaca. */
45 for (i = 1; i <= k; i++)
    printf("%d ", zbir_delilaca(i));
47 printf("\n");
49
    return 0;
}

```

### Rešenje 2.5.4

```

1  #include <stdio.h>

3  /* Funkcija za dva neoznacena broja x i n utvrđuje da li je
   x neki stepen broja n. Ukoliko jeste, funkcija vraća izlozilac
5  stepena, a u suprotnom vraća -1. */
int je_stepen(unsigned int x, unsigned int n)
7  {
    /* Na pocetku, s = n^i = n^1 = n. */
    int i = 1;
    unsigned int s = n;

11
    /* U svakoj iteraciji petlje, s se azurira tako da ima
13    vrednost n^i. Postupak se ponavlja dok je s manji od x. */
    while (s < x) {
15        s = s * n;
        i++;
17    }

19    /* Kako s ima vrednost n^i, ako vazi da je s jednako x, onda
       je bas brojac i trazeni izlozilac. */
21    if (s == x)
        return i;

23
    /* Ako nije, onda se vraća -1. */
25    return -1;
}

27
int main()
29 {
    /* Deklaracija potrebnih promenljivih. */
31    unsigned int x, n;
    int st;

33
    /* Ucitavaju se vrednosti x i n. */
35    printf("Unesite dva broja: ");
    scanf("%u%u", &x, &n);
37

```

```
/* Poziva se napisana funkcija. */
39 st = je_stepen(x, n);

41 /* U zavisnosti od povratne vrednosti funkcije, vrsi se
   ispis rezultata. */
43 if (st != -1)
    printf("%u=%u^%d\n", x, n, st);
45 else
    printf("%u nije stepen broja %u\n", x, n);
47
49 return 0;
}
```

### Rešenje 2.5.5

```
1 #include <stdio.h>

3 /* Funkcija koja racuna nzd(x,y) primenom Euklidovog algoritma. */
int euklid(int x, int y)
5 {
    int ostatak;
    /* Euklidov algoritam: trazi se nzd(x,y).
       Na primer nzd(12,18). Postupak koji se primenjuje je sledeci:
       1. ostatak = x % y = 12 % 18 = 12.
       2. x postaje y => x = 18
       3. y postaje ostatak => y = 12
       =>
       1. ostatak = x % y = 18 % 12 = 6
       2. x postaje y => x = 12
       3. y postaje ostatak => y = 6
       =>
       1. ostatak = x % y = 12 % 6 = 0
       2. x postaje y => x = 6
       3. y postaje ostatak => y = 0
       => procedura se zavrsava jer je y jednako 0, a
       rezultat je poslednji ne-nula ostatak, tj. x.*/
    while (y) {
        ostatak = x % y;
        x = y;
        y = ostatak;
    }

    /* Kao povratna vrednost funkcije se vraca x. */
    return x;
}

31
int main()
33 {
    /* Deklaracija potrebnih promenljivih. */
    int a, b;
35 }
```



```

37  /* Ucitavaju se vrednosti a i b. */
    printf("Unesite dva cela broja:");
39  scanf("%d%d", &a, &b);

41  /* Ispis rezultata. */
    printf("Najveci zajednicki delilac: %d\n", euklid(a, b));
43
    return 0;
45 }

```

### Rešenje 2.5.6

```

1  #include <stdio.h>

3  /* Funkcija koja racuna zbir reciprocnih vrednosti brojeva
   iz intervala [1,n]. */
5  float zbir_reciprocnih(int n)
   {
7      float zbir = 0;
       int i;

9      /* Za svako i izmedju 1 i n na zbir se dodaje vrednost 1/i.
       Napomena: zbog celobrojnog deljenja mora da stoji 1.0/i. */
11     for (i = 1; i <= n; i++)
13         zbir += 1.0 / i;

15     /* Kao povratna vrednost funkcije se vraca izracunati zbir. */
       return zbir;
17 }

19 int main()
   {
21     /* Deklaracija potrebne promenljive. */
       int n;

23     /* Ucitava se vrednost broja n. */
25     printf("Unesite broj n:\n");
       scanf("%d", &n);

27     /* Vrsi se provera ispravnosti ulaza. */
29     if(n <= 0)
       {
31         printf("Greska: neispravan unos.\n");
           return -1;
33     }

35     /* Ispis rezultata. */
       printf("Zbir reciprocnih: %.2f\n", zbir_reciprocnih(n));
37
       return 0;
39 }

```

### Rešenje 2.5.7

```
1  #include <stdio.h>
3  /* Funkcija broji koliko puta se realan broj x javlja u nizu unetih
   brojeva. */
5  int prebrojavanje(float x)
   {
7     float y;
     int broj_pojavljivanja = 0;
9
10    /* Brojevi se ucitavaju sve do unosa broja nula.
11       Svaki put kada se unese broj koji je jednak broju x,
       brojac pojavljivanja se uveca za 1. */
13    printf("Unesite brojeve:\n");
     while(1){
15        scanf("%f", &y);
17
         if(y == 0)
             break;
19
         if (x == y)
             broj_pojavljivanja++;
21     }
23
     return broj_pojavljivanja;
25 }
27 int main()
   {
29     /* Deklaracija potrebnih promenljivih. */
     float x;
     int rezultat;
31
32
33     /* Ucitava se vrednost broja x. */
     printf("Unesite broj x: ");
35     scanf("%f", &x);
37
38     /* Poziva se napisana funkcija i u promenljivoj rezultat se
       cuva njena povratna vrednost. */
39     rezultat = prebrojavanje(x);
41
42     /* Ispis rezultata. */
     printf("Broj pojavljivanja broja %.2f: %d\n", x, rezultat);
43
     return 0;
45 }
```

## Rešenje 2.5.8

```
1  #include <stdio.h>
2  #include <math.h>
3
4  /* Funkcija vraca 1 ako je broj prost, a 0 u suprotnom. */
5  int prost(int x){
6      int i;
7
8      /* Brojevi 2 i 3 su prosti. */
9      if (x == 2 || x == 3)
10         return 1;
11
12     /* Parni brojevi nisu prosti. */
13     if (x % 2 == 0)
14         return 0;
15
16     /* Ako se naidje na broj koji deli broj x, onda broj x nije
17        prost. Provera se vrsi za sve neparne brojeve izmedju 3 i
18        sqrt(x), jer kada bi x imao parnog delioca, onda bi i broj 2
19        delio x, a taj uslov je vec proveren. */
20     for (i = 3; i <= sqrt(x); i += 2)
21         if (x % i == 0)
22             return 0;
23
24     /* Ako nijedan od prethodnih uslova nije bio ispunjen, to znaci
25        da nijedan broj ne deli x, pa je on prost. */
26     return 1;
27 }
28
29 /* Funkcija ispisuje prvih n prostih brojeva.
30    Kljucna rec void oznacava da funkcija nema povratnu vrednost. */
31 void prvih_n_prostih(int n){
32     int broj_prostih = 0;
33     int k = 2;
34
35     /* Petlja se izvrsava dok god se ne istampa n prostih brojeva. */
36     while(broj_prostih < n)
37     {
38         /* Ako se naidje na broj koji je prost, ispisuje se njegova
39            vrednost i uvecava se brojac. */
40         if(prost(k))
41         {
42             printf("%d ", k);
43             broj_prostih++;
44         }
45
46         /* Prelazi se na sledeci broj. */
47         k++;
48     }
49     printf("\n");
50 }
```

```
51  /* Funkcija ispisuje sve proste brojeve cija je vrednost manja
53     od n. */
54  void prosti_brojevi_manji_od_n(int n)
55  {
56      /* Ukoliko je n manje ili jednako 2, onda nema prostih brojeva
57         koji su manji od njega. U tom slucaju se ispisuje odgovarajuca
58         poruka i naredbom return; se izlazi iz funkcije. */
59      if(n<=2)
60      {
61          printf("ne postoje\n");
62          return;
63      }
64
65      /* Za svaki broj k izmedju 2 i n-1 se vrši provera da li je prost
66         i ako jeste, ispisuje se njegova vrednost. */
67      int k = 2;
68      while(k < n)
69      {
70          if(prost(k))
71              printf("%d ", k);
72
73          k++;
74      }
75      printf("\n");
76  }
77
78  int main()
79  {
80      /* Deklaracija potrebnih promenljivih. */
81      int n;
82
83      /* Ucitava se broj n. */
84      printf("Unesite broj n:");
85      scanf("%d", &n);
86
87      /* Vrsi se provera ispravnosti ulaza. */
88      if(n <= 0)
89      {
90          printf("Greska: neispravan unos.\n");
91          return -1;
92      }
93
94      /* Ispis rezultata. */
95      printf("Prvih n prostih: ");
96      prvih_n_prostih(n);
97      printf("Prosti manji od n: ");
98      prosti_brojevi_manji_od_n(n);
99
100     return 0;
101 }
```

## Rešenje 2.5.10

```
1  #include <stdio.h>
2  #include <stdlib.h>
3
4  /* Funkcija racuna aritmeticku sredinu cifara datog celog broja. */
5  float aritmeticka_sredina(int x)
6  {
7      /* Aritmeticka sredina broja 0 je 0. */
8      if (x == 0)
9          return 0;
10
11     /* Deklaracija i inicijalizacija brojac. */
12     int zbir_cifara = 0;
13     int broj_cifara = 0;
14
15     /* Uzima se apsolutna vrednost broja x kako bi program ispravno
16        radio i za negativne brojeve. */
17     x = abs(x);
18
19     /* Dok god ima neobradjenih cifara, na zbir se dodaje poslednja
20        cifra, brojac cifara se uvecava za 1 i sa broja x se uklanja
21        poslednja cifra. */
22     while (x) {
23         zbir_cifara += x % 10;
24         broj_cifara++;
25         x /= 10;
26     }
27
28     /* Kao povratna vrednost funkcije se vraca odgovarajuci
29        kolicnik. */
30     return (float) zbir_cifara / broj_cifara;
31 }
32
33 int main()
34 {
35     /* Deklaracija potrebne promenljive. */
36     int x;
37
38     /* Ucitava se vrednost broja x. */
39     printf("Unesite broj: ");
40     scanf("%d", &x);
41
42     /* Ispis rezultata. */
43     printf("%.3f\n", aritmeticka_sredina(x));
44
45     return 0;
46 }
```

## Rešenje 2.5.11

```
1 #include<stdio.h>
2 #include<stdlib.h>
3
4 /* Funkcija ispituje da li se cifra c nalazi u zapisu celog broja
5    x. Vraca 1 ako je uslov ispunjen i 0 u suprotnom. */
6 int sadrzi(int x, int c)
7 {
8     /* Uzima se apsolutna vrednost broja x. */
9     x = abs(x);
10
11     /* Izdvaja se cifra po cifra broja x.
12        Ako se naidje na cifru cija je vrednost c, onda se kao
13        rezultat funkcije vraca 1 (jer x sadrzi c). */
14     while (x) {
15         if (x % 10 == c)
16             return 1;
17         x /= 10;
18     }
19
20     /* Ako se petlja zavrсила, znaci da se nijednom nije naislo
21        na cifru c, sto znaci da broj x ne sadrzi cifru c i kao
22        povratna vrednost funkcije se vraca 0. */
23     return 0;
24 }
25
26 int main()
27 {
28     /* Deklaracija potrebnih promenljivih. */
29     int x, c;
30
31     /* Ucitavaju se vrednosti x i c. */
32     printf("Unesite broj i cifru:");
33     scanf("%d%d", &x, &c);
34
35     /* Vrsi se provera ispravnosti ulaza. */
36     if(c < 0 || c > 9)
37     {
38         printf("Greska: neispravan unos.\n");
39         return -1;
40     }
41
42     /* U zavisnosti od povratne vrednosti funkcije, vrsi se ispis
43        odgovarajuće poruke. */
44     if (sadrzi(x, c))
45         printf("Cifra %d se nalazi u zapisu broja %d\n", c, x);
46     else
47         printf("Cifra %d se ne nalazi u zapisu broja %d\n", c, x);
48     return 0;
49 }
```

## Rešenje 2.5.12

```
1  #include<stdio.h>
2  #include<stdlib.h>
3
4  /* Funkcija odredjuje broj neparnih cifara u zapisu datog celog
5     broja. */
6  int broj_neparnih_cifara(int x)
7  {
8      int brojac_neparnih = 0;
9      char cifra;
10     x = abs(x);
11
12     while (x) {
13         /* Izdvaja se poslednja cifra broja. */
14         cifra = x % 10;
15         /* Moze se izbesci koriscenje naredbe if pomocu narednog izraza.
16            Naime, vrednost izraza cifra%2 je 1 kada je cifra neparna,
17            odnosno 0 kada je parna. Tako ce na broj neparnih cifara
18            biti dodata jednica ako je cifra neparna, a ako je parna
19            bice dodata 0, sto jeste zeljeno ponasanje. */
20         brojac_neparnih += (cifra % 2);
21         x /= 10;
22     }
23
24     return brojac_neparnih;
25 }
26
27 int main()
28 {
29     /* Deklaracija potrebne promenljive. */
30     int x;
31
32     /* Ucitavaju se brojevi sve do unosa broja nula i vrsi se ispis
33        broja neparnih cifara za svaki ucitani broj. */
34     printf("Unesite cele brojeve:\n");
35     while(1)
36     {
37         scanf("%d", &x);
38         if(x == 0)
39             break;
40
41         printf("Broj neparnih cifara: %d\n", broj_neparnih_cifara(x));
42     }
43
44     return 0;
45 }
```

## Rešenje 2.5.13

```
1  #include <stdio.h>
2  #include <stdlib.h>
3
4  /* Funkcija proverava da li su sve cifre broja x parne i vraca
5     1 ako je uslov ispunjen i 0 ako nije. */
6  int sve_parne_cifre(int x)
7  {
8     char cifra;
9     x = abs(x);
10
11     /* Ako se naidje na cifru koja nije parna, onda se kao povratna
12        vrednost funkcije vraca 0. */
13     while (x > 0) {
14         cifra = x % 10;
15         if (cifra % 2 == 1)
16             return 0;
17         x /= 10;
18     }
19
20     /* Ako se doslo do kraja petlje, znaci da se nije naislo ni na
21        jednu neparnu cifru, sto znaci da su sve cifre parne i da
22        treba da se vrati 1. */
23     return 1;
24 }
25
26 /* Funkcija proverava da li su sve cifre broja x jednake i vraca
27     1 ako jesu, a 0 u suprotnom. */
28 int sve_cifre_jednake(int x)
29 {
30     char poslednja_cifra;
31     x = abs(x);
32
33     /* Izdvaja se poslednja cifra broja. */
34     poslednja_cifra = x % 10;
35     x /= 10;
36
37     /* Za sve ostale cifre se proverava da li su jednake poslednjoj.
38        Ako se naidje na neku koja nije, onda nisu sve cifre broja
39        x jednake i kao povratna vrednost se vraca 0. */
40     while (x) {
41         if (x % 10 != poslednja_cifra)
42             return 0;
43
44         x /= 10;
45     }
46
47     /* Ako se stiglo do kraja petlje, znaci da su sve cifre broja
48        bile jednake poslednjoj cifri, pa se kao povratna vrednost
49        vraca 1. */
50     return 1;
51 }
52
```



```

int main()
{
    /* Deklaracija potrebne promenljive. */
    int x;

    /* Ucitava se broj x. */
    printf("Unesite broj:");
    scanf("%d", &x);

    /* U zavisnosti od povratne vrednosti napisanih funkcija
       vrsi se ispis odgovarajucih poruka. */
    if (sve_parne_cifre(x))
        printf("Sve cifre broja su parne.\n");
    else
        printf("Broj sadrzi bar jednu neparnu cifru.\n");

    if (sve_cifre_jednake(x))
        printf("Cifre broja su jednake.\n");
    else
        printf("Cifre broja nisu jednake.\n");

    return 0;
}

```

### Rešenje 2.5.14

```

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

5  /* Funkcija koja uklanja cifru sa pozicije p iz broja n.
   Cifra jedinica ima poziciju 1, desetica 2, itd.*/
7  int ukloni(int n, int p)
   {
9      int znak, tezina_pozicije, levi_deo, desni_deo;

11     /* Pamti se znak broja. */
       znak = n < 0 ? 1 : -1;

13     /* Uzima se apsolutna vrednost. */
       n = abs(n);

15     /* Racuna se tezina prosledjene pozicije. */
       tezina_pozicije = pow(10, p-1);

17     /* Broj se deli na dva dela - deo levo od cifre koja se izbacuje
       i deo desno od cifre koja se izbacuje. */
21     levi_deo = n/(10*tezina_pozicije);
23     desni_deo = n%tezina_pozicije;

25     /* Povratna vrednost funkcije se dobija spajanjem levog i desnog

```

## 2 Kontrola toka

```
    dela i mnozenjem sa znakom pocetnog broja. */
27     return znak * (levi_deo*10 + desni_deo);
    }
29
30 int main()
31 {
32     /* Deklaracija potrebnih promenljivih. */
33     int broj, p;
34
35     /* Ucitava se vrednost pozicije. */
36     printf("Unesite poziciju: ");
37     scanf("%d", &p);
38
39     /* Vrsi se provera ispravnosti ulaza. */
40     if(p <= 0)
41     {
42         printf("Greska: neispravan unos.\n");
43         return -1;
44     }
45
46     /* Ucitavaju se brojevi dok se ne unese nula i za svaki
47        ucitani broj se ispisuje broj koji se dobije uklanjajem
48        cifre koja se nalazi na poziciji p. */
49     while (1) {
50         printf("Unesite broj: ");
51         scanf("%d", &broj);
52
53         if (broj == 0)
54             break;
55
56         printf("%d\n", ukloni(broj, p));
57     }
58
59     return 0;
60 }
```

### Rešenje 2.5.15

```
1  #include <stdio.h>
2  #include <stdlib.h>
3
4  /* Funkcija proverava da li se neka cifra nalazi u zapisu celog
5     broja i ako se nalazi vraca odgovarajucu poziciju (tj. njenu
6     tezinu koja je neki stepen broja 10), a u suprotnom vraca -1.
7     Na primer, za broj = 1234 i cifra = 2, funkcija vraca 100. */
8  int pozicija_cifre(int broj, int cifra)
9  {
10     int tezina_pozicije = 1;
11
12     while(broj)
13     {
```

```
15     if(broj%10 == cifra)
16         return tezina_pozicije;
17
18     tezina_pozicije *= 10;
19     broj /= 10;
20 }
21
22 return -1;
23 }
24
25 /* Funkcija koja iz zapisa broja izbacuje cifru koja se nalazi
26    na prosledjenoj poziciji. Pozicija je stepen broja 10.
27    Na primer, za x=1234 i pozicija = 10, treba da se izbací 3.
28    levi_deo = 1234/(10*10) = 12
29    desni_deo = 1234%10 = 4
30    Povratna vrednost je 12*10 + 4 = 124. */
31 int izbaci_cifru(int broj, int pozicija)
32 {
33     int levi_deo = broj/(pozicija*10);
34     int desni_deo = broj%pozicija;
35     return levi_deo*10 + desni_deo;
36 }
37
38 /* Funkcija proverava da li su dva cela broja napisana pomocu istih
39    cifara. Vraca 1 ako je uslov ispunjen, a 0 u suprotnom. */
40 int zapis(int x, int y)
41 {
42     int pozicija;
43     x = abs(x);
44     y = abs(y);
45
46     while (x) {
47         /* Proverava se da li y sadrzi poslednju cifru broja x. */
48         pozicija = pozicija_cifre(y, x % 10);
49
50         /* Ako ne sadrzi, x i y se ne zapisuju pomocu istih cifara. */
51         if(pozicija == -1)
52             return 0;
53
54         /* Ako sadrzi, iz x se izbacuje poslednja cifra, a iz y se
55            izbacuje ista ta cifra (koja se nalazi na pronadjenoj
56            poziciji. */
57         x /= 10;
58         y = izbaci_cifru(y, pozicija);
59     }
60
61     /* Na kraju petlje iz x su izbacene sve cifre, a vazi da su
62        brojevi zapisani pomocu istih cifara samo ukoliko ni u y
63        nema preostalih cifara. */
64     return y == 0;
65 }
```

## 2 Kontrola toka

---

```
int main()
67 {
    /* Deklaracija potrebnih promenljivih. */
69     int x, y;

71     /* Ucitavaju se vrednosti x i y. */
    printf("Unesite dva cela broja: ");
73     scanf("%d%d", &x, &y);

75     /* U zavisnosti od povratne vrednosti napisane funkcija,
        ispisuje se odgovarajuca poruka. */
77     if (zapis(x, y))
        printf("Uslov je ispunjen.\n");
79     else
        printf("Uslov nije ispunjen.\n");
81
83     return 0;
}
```

### Rešenje 2.5.16

```
1  #include <stdio.h>
   #include <stdlib.h>
3
   /* Funkcija proverava da li se cifre u zapisu broja nalaze u
   5   neopadajućem poretku. */
   int neopadajuće(int n)
7   {
       int tekuća_cifra, prethodna_cifra;
9       n = abs(n);

11      /* Izvan petlje se izdvaja poslednja cifra u zapisu broja da bi u
          petlji mogla da se poredi sa sledećom. */
13      prethodna_cifra = n % 10;
          n /= 10;

15      /* U petlji se proverava poredak svake dve susedne cifre. Ukoliko
          se detektuje da je poredak narusen, izlazi se iz funkcije i
          vraća se vrednost 0. */
17      while (n) {
          tekuća_cifra = n % 10;

21          if (tekuća_cifra > prethodna_cifra)
23              return 0;

25          /* Tekuća cifra postaje prethodna za narednu iteraciju. */
          prethodna_cifra = tekuća_cifra;
27          n /= 10;
      }

29      /* Nakon izlaska iz petlje povratna vrednost funkcije je 1 jer u
```

```

31     slucaju da je poredak u nekom trenutku narusen iz funkcije bi
        se izaslo jos u petlji. */
33     return 1;
    }
35
37     int main()
38     {
        /* Deklaracija potrebne promenljive. */
39     int n;

41     /* Ucitava se vrednost broja n. */
        printf("Unesite broj: ");
43     scanf("%d", &n);

45     /* U zavisnosti od povratne vrednosti napisane funkcije vrsi
        se ispis odgovarajuce poruke. */
47     if (neopadajuce(n))
        printf("Cifre su u neopadajucem poretku.\n");
49     else
        printf("Cifre nisu u neopadajucem poretku.\n");
51
        return 0;
53     }

```

### Rešenje 2.5.17

```

1  #include <stdio.h>
    #include <stdlib.h>
3
    /* Funkcija proverava da li su cifre broja naizmenicno parne i
        neparne. Ako je uslov ispunjen vraca 1, u suprotnom vraca 0. */
5  int par_nepar(int x)
6  {
7      int prethodna_cifra, tekuca_cifra;
9      x = abs(x);

11     /* Poslednja cifra broja se izdvaja van petlje da bi u petlji
        moglo da se vrsi poredjenje. */
13     prethodna_cifra = x % 10;
        x /= 10;

15     while (x) {
17         tekuca_cifra = x % 10;

19         /* Ukoliko su uzastopne cifre iste parnosti, uslov nije
            ispunjen, rad petlje i funkcije se prekida i vraca se 0. */
21         if (tekuca_cifra % 2 == prethodna_cifra % 2)
            return 0;

23         /* Tekuca cifra postaje prethodna cifra za narednu iteraciju. */
25         prethodna_cifra = tekuca_cifra;
    }

```

```

    x /= 10;
27 }

29 /* Sve uzastopne cifre su razlicite parnosti jer ni jednom u
    petlji uslov da su cifre iste parnosti nije bio ispunjen. */
31 return 1;
}

33
35 int main()
36 {
37     /* Deklaracija potrebne promenljive. */
38     int n;

39     /* Ucitava se vrednost broja n. */
40     printf("Unesite broj n: ");
41     scanf("%d", &n);

42
43     /* U zavisnosti od povratne vrednosti napisane funkcije, vrsi
        se ispis odgovarajuće poruke. */
44     if (par_nepar(n))
45         printf("Broj ispunjava uslov.\n");
46     else
47         printf("Broj ne ispunjava uslov.\n");
48
49     return 0;
51 }
```

### Rešenje 2.5.18

```

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

4
/* Funkcija koja racuna broj cifara celog broja n. */
6 int broj_cifara(int n)
7 {
8     int brojac = 0;
9     n = abs(n);

10
11     if(n < 10)
12         return 1;

13
14     while(n)
15     {
16         brojac++;
17         n /= 10;
18     }

19
20     return brojac;
21 }
22
```

```

24  /* Funkcija koja racuna broj koji se dobija rotacijom broja n za
    jedno mesto u levo. */
26  int rotacija(int n)
28  {
    int znak, prva_cifra, n_bez_prve_cifre, br_cifara;

    znak = (n < 0) ? -1 : 1;
    n = abs(n);
    br_cifara = broj_cifara(n);

    /* Izdvajaju se prva cifra i deo broja bez prve cifre.
       Na primer: ako je n = 1234 onda je br_cifara = 4
       prva_cifra se dobija kao:
       n / (10^(br_cifara-1)) = 1234/1000 = 1.
       n_bez_prve_cifre se dobija kao: n%1000 = 234. */
    int tezina_pozicije = pow(10, br_cifara-1);
    prva_cifra = n / tezina_pozicije;
    n_bez_prve_cifre = n % tezina_pozicije;

    /* Rezultat se dobija nadovezivanjem prve cifre na kraj i
       mnozenjem sa znakom pocetnog broja. */
    return znak * (n_bez_prve_cifre*10 + prva_cifra);
26  }

46  int main()
48  {
    /* Deklaracija potrebne promenljive. */
    int n;

    /* Brojevi se ucitavaju sve do unosa broja nula i ispisuju
       se brojevi dobijeni kao rezultat izvršavanja funkcije rotacija
       nad unetim brojevima. */
    while (1) {
    56     printf("Unesite broj: ");
        scanf("%d", &n);

    58     if (n == 0)
    60         break;

    62     printf("%d\n", rotacija(n));
    }

    64     return 0;
    66 }

```

### Rešenje 2.5.19

```

1  #include<stdio.h>

3  /* Funkcija koja vraca zbir cifara datog broja x. */
   int zbir_cifara(int x)

```

```
5 {
    int zbir = 0;
7   while (x) {
        zbir += x%10;
9       x /= 10;
    }
11  return zbir;
}

13
/* Funkcija koja vraca 1 ako je broj srecan, a 0 u suprotnom. */
15 int srecan(int x)
{
17     /* Dok god u broju x ima vise od 2 cifre, vrednost broja x se
        zamenjuje sa zbirom njegovih cifara. Na primer, za pocetno
19     x = 7698, nakon prve iteracije x postaje 7+6+9+8 = 30, nakon
        druge iteracije x postaje 3 + 0 = 3 i zatim se izlazi iz
21     petlje. */
    while(x <= 10)
23         x = zbir_cifara(x);

25     /* Broj je srecan ako na kraju x ima vrednost 1. */
    return (x == 1);
27 }

29 int main()
{
31     /* Deklaracija potrebnih promenljivih. */
    int n, i;

33
    /* Ucitava se vrednost broja n. */
35     printf("Unesite broj n: ");
    scanf("%d", &n);

37
    /* Vrsi se provera ispravnosti ulaza. */
39     if(n <= 0)
    {
41         printf("Greska: neispravan unos.\n");
        return -1;
43     }

45     /* Ispisuju se svi srecni brojevi manji ili jednaki n. */
    printf("Srecni brojevi: ");
47     for (i = 1; i <= n; i++)
        if (srecan(i))
49         printf("%d ", i);

51     printf("\n");
    return 0;
53 }
```

### Rešenje 2.5.20



```
1  #include <stdio.h>
2  #include <math.h>
3  #include <stdlib.h>
4
5  /* Funkcija koja racuna broj cifara celog broja n. */
6  int broj_cifara(int n)
7  {
8      int brojac = 0;
9      n = abs(n);
10
11     if(n < 10)
12         return 1;
13
14     while(n)
15     {
16         brojac++;
17         n /= 10;
18     }
19
20     return brojac;
21 }
22
23 /* Funkcija proverava da li je broj Armstrongov. */
24 int armstrong(int x)
25 {
26     int suma = 0;
27     int n = broj_cifara(x);
28     int x_pocetno = x;
29
30     /* Racuna se suma n-tih stepena cifara broja x. */
31     while (x) {
32         suma += pow(x % 10, n);
33         x /= 10;
34     }
35
36     /* Ako je suma jednaka pocetnoj vrednosti broja x, broj je
37        Armstrongov, u suprotnom nije. */
38     return x_pocetno == suma;
39 }
40
41 int main()
42 {
43     /* Deklaracija potrebne promenljive. */
44     int x;
45
46     /* Ucitava se vrednost broja x. */
47     printf("Unesite broj: ");
48     scanf("%d", &x);
49
50     /* Proverava se da li je x Armstrongov broj i ispisuje se
51        odgovarauca poruka. */
```

```
    if (armstrong(x))
53         printf("Broj je Armstrongov.\n");
    else
55         printf("Broj nije Armstrongov.\n");

57     return 0;
}
```

### Rešenje 2.5.21

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

4 /* Funkcija racuna vrednost e^x kao parcijalnu sumu reda
   suma(x^n/n!), gde indeks n ide od 0 do beskonacno, pri cemu
6   se sumiranje vrši dok je razlika sabiraka u redu po apsolutnoj
   vrednosti manja od eps. */
8 double e_na_x(double x, double eps){
    double s = 1;
10    double clan = 1;
    int n = 1;

12    /* Parcijalnu suma se formira tako sto se u svakoj iteraciji
14     petlje promenljivoj s doda jedan sabirak sume oblika (x^n)/n!
     koji se cuva u promenljivoj clan.

16     Svaki sabirak se dobija na osnovu prethodnog tako sto se
18     prethodni pomnozi sa x i podeli sa n (n predstavlja redni broj
     sabirka u sumi).

20     Prvi sabirak (kome odgovara n=0) iznosi 1; zbog toga
22     promenljive s i clan se inicijalizuju na vrednost 1.

24     Sumiranje se sprovodi sve dok je sabirak po apsolutnoj
     vrednosti veci od trazene tacnosti eps. */
26    do {
        clan = (clan * x) / n;
28        s += clan;
        n++;
30    } while (fabs(clan) > eps);

32    return s;
}

34
36 int main()
{
38     /* Deklaracija potrebnih promenljivih. */
    double x, eps;

40     /* Ucitavavaju se vrednosti x i eps. */
    printf("Unesite broj x: ");
```

```

42 scanf("%lf", &x);
   printf("Unesite eps: ");
44 scanf("%lf", &eps);

46 /* Ispis rezultata. */
   printf("Rezultat: %f\n", e_na_x(x, eps));
48 return 0;
}

```

### Rešenje 2.5.22

```

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

4  /* Funkcija ispisuje vrednosti funkcije sin(x) u n ravnomeno
   rasporedjenih tacaka na intervalu [a,b]. */
6  void ispis(float a, float b, int n)
   {
8     float i;
     float korak = (b - a) / (n - 1);

10    for (i = a; i <= b; i += korak)
12        printf("%.4f ", sin(i));

14    printf("\n");
   }

16  int main()
18  {
   /* Deklaracija potrebnih promenljivih. */
20    float a, b;
     int n;

22    /* Ucitavaju se granice intervala i vrsi se provera ispravnosti
   ulaza. */
24    printf("Unesite dva realna broja:");
     scanf("%f%f", &a, &b);
26    if(b >= a)
   {
28        printf("Greska: neispravan unos.\n");
         return -1;
   }

32    /* Ucitava se broj n i vrsi se provera ispravnosti ulaza. */
34    printf("Unesite broj n:");
     scanf("%d", &n);
36    if (n <= 1) {
         printf("Greska: neispravan unos.\n");
38        return -1;
   }

40

```

```
42  /* Ispis rezultata. */
    ispis(a, b, n);
44  return 0;
}
```

### Rešenje 2.5.23

```
1  #include <stdio.h>

3  /* Funkcija vraća karakter koji se u abecedi nalazi k mesta pre
   datog karaktera c. */
5  char sifra(char c, int k)
{
7  /* Provera da li je karakter malo slovo. */
   if (c >= 'a' && c <= 'z') {
9      /* Ako karakter koji je k pozicija pre datog karaktera ispada
       iz opsega malih slova. */
11     if (c - k < 'a')
        /* Od k se oduzima rastojanje izmedju c i 'a' (jer je za
13        toliko karaktera vec vrateno u nazad), kako bi se odredilo
        koliko preostali broj karaktera koji treba preskociti od
15        karaktera 'z'. */
        return 'z' - (k - (c - 'a') - 1);
17     else
        /* U suprotnom, karakter c-k ne ispada iz opsega malih slova,
19        te je dovoljno njega vratiti. */
        return c - k;
21 }
   else if (c >= 'A' && c <= 'Z') {
23     /* Postupak se ponavlja i za velika slova. */
       if (c - k < 'A')
25         return 'Z' - (k - (c - 'A') - 1);
       else
27         return c - k;
   }
29
   /* Ako nije ni malo ni veliko slovo, karakter se ne menja. */
31   return c;
}

33
35 int main()
{
37     /* Deklaracija potrebnih promenljivih. */
   int k;
   char c;
39
   /* Ucitava se vrednost k. */
41   printf("Unesite broj k: ");
   scanf("%d", &k);
43
```

```
45  /* Ucitavaju se karakteri sve do kraja ulaza i ispisuje se
    njihova sifra. */
47  printf("Unesite tekst (CTRL + D za prekid): ");
    while ((c = getchar()) != EOF)
        putchar(sifra(c, k));
49
    return 0;
51 }
```

### Rešenje 2.5.25

```
1  #include<stdio.h>

3  /* Funkcija koja proverava da li je godina prestupna. */
    int prestupna(int godina)
5  {
        if ((godina % 100 != 0 && godina % 4 == 0) || godina % 400 == 0)
7            return 1;
        else
9            return 0;
    }

11 /* Funkcija odredjuje broj dana u datom mesecu. */
    int broj_dana(int mesec, int godina)
13 {
15     switch (mesec) {
16         case 1:
17         case 3:
18         case 5:
19         case 7:
20         case 8:
21         case 10:
22         case 12:
23             return 31;
24         case 4:
25         case 6:
26         case 9:
27         case 11:
28             return 30;
29         case 2:
30             if (prestupna(godina))
31                 return 29;
32             else
33                 return 28;
34     }
35     return -1;
36 }

37 /* Funkcija proverava da li je datum ispravan. Ako je datum
38 ispravan funkcija vraca 1, inace vraca 0. */
    int ispravan(int dan, int mesec, int godina)
```

```
41 {
42     /* Ako je godina negativna, datum nije ispravan. */
43     if (godina < 0)
44         return 0;
45
46     /* Ako mesec nije u opsegu od 1 do 12, datum nije ispravan. */
47     if (mesec < 1 || mesec > 12)
48         return 0;
49
50     /* Ako je dan manji od 1 ili veci od broja dana u datom mesecu,
51        datum nije ispravan. */
52     if (dan < 1 || dan > broj_dana(mesec, godina))
53         return 0;
54
55     return 1;
56 }
57
58 /* Funkcija koja racuna sledeci dan. */
59 void sledeci_dan(int dan, int mesec, int godina)
60 {
61     /* Za kraj godine, odnosno za datum 31.12. sledeci datum je 1.1.
62        i godina se uvecava za jedan. */
63     if (mesec == 12 && dan == 31)
64         printf("1.1.%d.\n", godina + 1);
65     /* Ukoliko je dan jednak poslednjem danu u tom mesecu, odnosno
66        ako je jednak broju dana u tom mesecu, onda je sledeci datum
67        kada se mesec uveca za 1, a dan postane 1. Bitan je redosled
68        ovih naredbi. Ako bi ovo ispitivanje bilo prvo, onda bi se
69        mesec mogao uvecati na 13. sto ne bi bio ispravan datum. Zato
70        se prvo proverava da li je kraj godine, pa tek onda da li je
71        kraj meseca. */
72     else if (dan == broj_dana(mesec, godina))
73         printf("1.%d.%d.\n", mesec + 1, godina);
74     /* Ako nije ni jedan od prethodna dva slucaja, onda se dan moze
75        uvecati na 1, bez bojazni da ce se prekoračiti broj dana u
76        datom mesecu. */
77     else
78         printf("%d.%d.%d.\n", dan + 1, mesec, godina);
79 }
80
81 int main()
82 {
83     /* Deklaracija potrebnih promenljivih. */
84     int dan, mesec, godina;
85
86     /* Ucitavaju se vrednosti dana, meseca i godine. */
87     printf("Unesite datum:");
88     scanf("%d.%d.%d.", &dan, &mesec, &godina);
89
90     /* Vrsi se provera ispravnosti datuma. */
91     if (!ispravan(dan, mesec, godina)) {
92         printf("Uneti datum nije ispravan.\n");
93     }
```

```
93     return -1;
94 }
95
96 /* Poziva se funkcija za ispis sledeceg dana. */
97 printf("Datum sledeceg dana je:");
98 sledeci_dan(dan, mesec, godina);
99
100 return 0;
101 }
```

### Rešenje 2.5.26

Za rešavanje ovog zadatka koristi se funkcija `od_nove_godine` koja je definisana u rešenju zadatka 2.5.28.

### Rešenje 2.5.27

Za rešavanje ovog zadatka koristi se funkcija `do_kraja_godine` koja je definisana u rešenju zadatka 2.5.28.

### Rešenje 2.5.28

```
#include<stdio.h>
2
3 /* Funkcija koja proverava da li je godina prestupna. */
4 int prestupna(int godina)
5 {
6     if ((godina % 100 != 0 && godina % 4 == 0) || godina % 400 == 0)
7         return 1;
8     else
9         return 0;
10 }
11
12 /* Funkcija odredjuje broj dana u datom mesecu. */
13 int broj_dana(int mesec, int godina)
14 {
15     switch (mesec) {
16     case 1:
17     case 3:
18     case 5:
19     case 7:
20     case 8:
21     case 10:
22     case 12:
23         return 31;
24     case 4:
25     case 6:
26     case 9:
27     case 11:
28         return 30;
```

```

30     case 2:
31         if (prestupna(godina))
32             return 29;
33         else
34             return 28;
35     }
36     return -1;
37 }
38
39 /* Funkcija proverava da li je datum ispravan. Ako je datum
40    ispravan funkcija vraca 1, inace vraca 0. */
41 int ispravan(int dan, int mesec, int godina)
42 {
43     /* Ako je godina negativna, datum nije ispravan. */
44     if (godina < 0)
45         return 0;
46
47     /* Ako mesec nije u opsegu od 1 do 12, datum nije ispravan. */
48     if (mesec < 1 || mesec > 12)
49         return 0;
50
51     /* Ako je dan manji od 1 ili veci od broja dana u datom mesecu,
52        datum nije ispravan. */
53     if (dan < 1 || dan > broj_dana(mesec, godina))
54         return 0;
55
56     return 1;
57 }
58
59 /* Funkcija odredjuje koliko dana je proteklo od pocetka godine. */
60 int od_nove_godine(int dan, int mesec, int godina)
61 {
62     int suma_dana = 0, i;
63
64     /* Za sve mesece pre datog datuma dodaje se broj dana za dati
65        mesec. */
66     for (i = 1; i < mesec; i++)
67         suma_dana += broj_dana(mesec, godina);
68
69     /* Na kraju se dodaje koliko je dana proteklo u datom mesecu, a
70        to je zadato sa promenljivom dan. */
71     return suma_dana + dan;
72 }
73
74 /* Funkcija odredjuje koliko dana ima do kraja godine. */
75 int do_kraja_godine(int dan, int mesec, int godina)
76 {
77     int suma_dana = 0, i;
78
79     /* Za sve mesece posle datog datuma dodaje se broj dana za dati
80        mesec. */
81     for (i = mesec + 1; i <= 12; i++)

```



```
    suma_dana += broj_dana(mesec, godina);
82
    /* Na kraju se dodaje koliko je dana je ostalo u datom mesecu. */
84    return suma_dana + broj_dana(mesec, godina) - dan;
}

86
/* Funkcija vraca 1 ako je prvi datum pre drugog datuma. U
88    suprotnom vraca 0. */
int prethodi(int dan1, int mesec1, int godina1, int dan2,
90             int mesec2, int godina2)
{
92     if (godina1 < godina2)
        return 1;
94     else if (godina1 > godina2)
        return 0;
96     else if (mesec1 < mesec2)
        return 1;
98     else if (mesec1 > mesec2)
        return 0;
100    else if (dan1 < dan2)
        return 1;
102    else
        return 0;
104}

106/* Funkcija vraca broj dana u datoj godini. */
int broj_dana_u_godini(int godina)
108{
    if (prestupna(godina))
110        return 366;
    else
112        return 365;
}

114
/* Funkcija koja racuna broj dana izmedju dva datuma. */
116int broj_dana_izmedju(int dan1, int mesec1, int godina1, int dan2,
                       int mesec2, int godina2) {
118    int pom, i;
    int suma_dana = 0;

120
    /* Vrsi se provera koji od datuma je ranije i ukoliko je to
122    potrebno, razmenjuju se tako da broj 1 ide uz prvi datum. */
    if (!prethodi(dan1, mesec1, godina1, dan2, mesec2, godina2)) {
124        pom = dan1;
        dan1 = dan2;
126        dan2 = pom;

128        pom = mesec1;
        mesec1 = mesec2;
130        mesec2 = pom;

132        pom = godina1;
```

```
134     godina1 = godina2;
135     godina2 = pom;
136 }
137
138 /* Ako su godine razlicite. */
139 if (godina1 != godina2) {
140     /* Za manji datum dodaje se broj dana do kraja godine. */
141     suma_dana = do_kraja_godine(dan1, mesec1, godina1);
142
143     /* Za sve godine koje su izmedju dve date godine dodaje se broj
144     dana u tim godinama. */
145     for (i = godina1 + 1; i < godina2; i++)
146         suma_dana += broj_dana_u_godini(i);
147
148     /* Za veci datum dodaje se broj dana od pocetka godine. */
149     suma_dana += od_nove_godine(dan2, mesec2, godina2);
150 }
151 /* Ako su godine iste, ali meseci razliciti. */
152 else if (mesec1 != mesec2) {
153     /* Dodaje se broj dana do kraja prvog meseca. */
154     suma_dana = broj_dana(mesec1, godina1) - dan1;
155
156     /* Dodaje se broj dana za svaki mesec koji je izmedju dva data
157     meseca. Kako su godina1 i godina2 jednake svejedno je koja
158     od ove dve promenljive se koristi u pozivu funkcije. */
159     for (i = mesec1 + 1; i < mesec2; i++)
160         suma_dana += broj_dana(i, godina1);
161
162     /* Dodaje se broj dana od pocetka meseca. */
163     suma_dana += dan2;
164 }
165 /* Ako su i godine i meseci jednaki. */
166 else
167     suma_dana = dan2 - dan1;
168
169 return suma_dana;
170 }
171
172 int main()
173 {
174     /* Deklaracija potrebnih promenljivih. */
175     int dan1, mesec1, godina1, dan2, mesec2, godina2;
176
177     /* Ucitavaju se datumi. */
178     printf("Unesite prvi datum:");
179     scanf("%d.%d.%d.", &dan1, &mesec1, &godina1);
180
181     printf("Unesite drugi datum:");
182     scanf("%d.%d.%d.", &dan2, &mesec2, &godina2);
183
184     /* Vrsi se provera ispravnosti unetih datuma. */
185     if (!ispravan(dan1, mesec1, godina1)
```

```

186     || !ispravan(dan2, mesec2, godina2)) {
188     printf("Uneti datum nije ispravan.\n");
188     return -1;
188 }

190 /* Ispis rezultata. */
190 printf("Broj dana izmedju dva datuma je: %d\n",
192     broj_dana_izmedju(dan1, mesec1, godina1, dan2, mesec2,
194     godina2));
194
194     return 0;
196 }

```

### Rešenje 2.5.29

```

1  #include<stdio.h>

3  /* Funkcija koja iscrtava romb. */
3  void romb(int n)
5  {
5      int i, j;

7      /* Petlja iscrtava liniju po liniju romba. */
9      for (i = 0; i < n; i++) {
11         /* U svakoj liniji prvo se ispisuje n-i-1 razmaka. */
11         for (j = 0; j < n - i - 1; j++)
13             printf(" ");

13         /* Ispisuje se n zvezdica. */
15         for (j = 0; j < n; j++)
17             printf("*");

17         /* Ispisuje se novi red. */
19         printf("\n");
19     }
21 }

23 int main()
23 {
25     /* Deklaracija potrebne promenljive. */
25     int n;

27     /* Ucitava se vrednost broja n. */
29     printf("Unesite broj n: ");
29     scanf("%d", &n);

31     /* Vrsi se provera ispravnosti ulaza. */
33     if (n <= 0){
33         printf("Greska: neispravan unos.\n");
35         return -1;
35     }

```

```
37      /* Iscrtava se romb. */
39      romb(n);
41      return 0;
}
```

### Rešenje 2.5.30

```
1  #include<stdio.h>
3  /* Funkcija koja stampa n zvezdica za kojima sledi znak za novi
   red. */
5  void stampaj_zvezdice(int n)
6  {
7      int i;
8      for (i = 0; i < n; i++)
9          printf("*");
11     printf("\n");
12 }
13
14 /* Funkcija koja crta grafikon. */
15 void grafikon_h(int a, int b, int c, int d)
16 {
17     /* Prvo se ispisuje a zvezdica. */
18     stampaj_zvezdice(a);
19
20     /* Postupak se ponavlja za vrednosti b, c i d. */
21     stampaj_zvezdice(b);
22     stampaj_zvezdice(c);
23     stampaj_zvezdice(d);
24 }
25
26 int main()
27 {
28     /* Deklaracija potrebnih promenljivih. */
29     int a, b, c, d;
30
31     /* Ucitavaju se vrednosti a,b,c,d. */
32     printf("Unesite brojeve: ");
33     scanf("%d%d%d%d", &a, &b, &c, &d);
34
35     /* Vrsi se provera ispravnosti ulaza i ispisuje se rezultat. */
36     if (a < 0 || b < 0 || c < 0 || d < 0) {
37         printf("Greska: neispravan unos.\n");
38         return -1;
39     }
40     else
41         grafikon_h(a, b, c, d);
}
```

```
43     return 0;
    }
```

### Rešenje 2.5.31

```
1  #include<stdio.h>

3  /* Funkcija koja racuna najveći od 4 prosledjena broja. */
   int maksimum(int a, int b, int c, int d)
5  {
       int max;

7      max = a;
9      if (b > max)
           max = b;
11     if (c > max)
           max = c;
13     if (d > max)
           max = d;

15     return max;
17 }

19 /* Pomocna funkcija za stampanje beline ili zvezdice. */
   void stampa_j_znak(int polje, int granica)
21 {
       if (polje < granica)
23         printf(" ");
       else
25         printf("*");
   }

27 /* Funkcija koja iscrtava vertikalni grafikon. */
   void grafikon_v(int a, int b, int c, int d)
29 {
       int i, max;

33     /* Na pocetku je potrebno pronaci najveću od ove cetiri
           vrednosti. */
35     max = maksimum(a, b, c, d);

37     /* Grafikon ukupno ima max horizontalnih linija. */
       for (i = 0; i < max; i++) {
39
           /* U svakoj od horizontalnih linija se nalazi po 4 polja: polje
41             za a,b,c i d uspravnu liniju. U svako od polja treba da se
           upise ili * ili belina, u zavisnosti od vrednosti i toga
43             koja linija se trenutno ispisuje. */

45     /* Stampa se znak za polje a. */
           stampa_j_znak(i, max - a);
```

```
47      /* Stampa se znak za polje b. */
49      stampaj_znak(i, max - b);

51      /* Stampa se znak za polje c. */
53      stampaj_znak(i, max - c);

55      /* Stampa se znak za polje d. */
57      stampaj_znak(i, max - d);

59      /* Na kraju svake horizontalne linije stampa se novi red. */
61      printf("\n");
62  }
63  }
64
65  int main()
66  {
67      /* Deklaracija potrebnih promenljivih. */
68      int a, b, c, d;

69      /* Ucitavaju se vrednosti cetiri broja. */
70      printf("Unesite brojeve: ");
71      scanf("%d%d%d%d", &a, &b, &c, &d);

72      /* Vrsi se provera ispravnosti ulaza i poziva se funkcija za
73         ispis grafikona. */
74      if (a < 0 || b < 0 || c < 0 || d < 0){
75          printf("Greska: neispravan unos.\n");
76          return -1;
77      }
78      else
79          grafikon_v(a, b, c, d);

80      return 0;
81  }
```

## 3

# Predstavljanje podataka

## 3.1 Nizovi

**Zadatak 3.1.1** Ako su  $a = (a_1, \dots, a_n)$  i  $b = (b_1, \dots, b_n)$  vektori dimenzije  $n$ , njihov skalarni proizvod se definiše kao  $a \cdot b = a_1 \cdot b_1 + \dots + a_n \cdot b_n$ . Napisati program koji računa skalarni proizvod dva vektora. Vektori se zadaju kao celobrojni nizovi sa najviše 100 elemenata. Program učitava dimenziju i elemente nizova, a na izlaz ispisuje vrednost skalarnog proizvoda. U slučaju neispravnog unosa, ispisati odgovarajuću poruku o grešci.

### Primer 1

```
INTERAKCIJA SA PROGRAMOM:
Unesite dimenziju vektora:
5
Unesite koordinate vektora a:
8 -2 0 2 4
Unesite koordinate vektora b:
35 12 5 -6 -1
Skalarni proizvod vektora a i b:
240
```

### Primer 2

```
INTERAKCIJA SA PROGRAMOM:
Unesite dimenziju vektora:
3
Unesite koordinate vektora a:
-1 0 1
Unesite koordinate vektora b:
5 5 5
Skalarni proizvod vektora a i b:
0
```

### Primer 3

```
INTERAKCIJA SA PROGRAMOM:
Unesite dimenziju vektora:
120
Greska: neispravan unos.
```

[Rešenje 3.1.1]

**Zadatak 3.1.2** Napisati program koji za učitani niz ispisuje:

### 3 Predstavljanje podataka

---

- (a) elemente niza koji se nalaze na parnim pozicijama.
- (b) parne elemente niza.

Maksimalan broj elemenata niza je 100. U slučaju neispravnog unosa, ispisati odgovarajuću poruku o grešci.

#### Primer 1

```
INTERAKCIJA SA PROGRAMOM:
Unesite dimenziju niza:
6
Unesite elemente niza:
1 8 2 -5 -13 75
Elementi niza na parnim pozicijama:
1 2 -13
Parni elementi niza:
8 2
```

#### Primer 2

```
INTERAKCIJA SA PROGRAMOM:
Unesite dimenziju niza:
3
Unesite elemente niza:
11 81 -63
Elementi niza na parnim pozicijama:
11 -63
Parni elementi niza:
```

#### Primer 3

```
INTERAKCIJA SA PROGRAMOM:
Unesite dimenziju niza:
-4
Greska: neispravan unos.
```

[Rešenje 3.1.2]

**Zadatak 3.1.3** Takmičari na Beogradskom maratonu su označeni rednim brojevima počevši od 0, a vreme za koje su istrčali maraton je dato minutima. Ovi podaci zadati su nizom celih brojeva, pri čemu indeks niza označava redni broj takmičara, a vrednosti u nizu označavaju vreme trčanja. Napisati funkcije za rad sa ovim nizom.

- (a) Napisati funkciju `void ucitaj(int a[], int n)` koja učitava elemente niza  $a$  dimenzije  $n$ .
- (b) Napisati funkciju `void stampaj(int a[], int n)` koja štampa elemente niza  $a$  dimenzije  $n$ .
- (c) Napisati funkciju `int suma(int a[], int n)` koja računa i vraća ukupno vreme trčanja svih takmičara.
- (d) Napisati funkciju `float prosek(int a[], int n)` koja računa i vraća prosečno vreme (aritmetičku sredinu) trčanja takmičara.
- (e) Napisati funkciju `int maksimum(int a[], int n)` koja izračunava i vraća najduže vreme trčanja takmičara.



- (f) Napisati funkciju `int pozicija_minimum(int a[], int n)` koja vraća redni broj pobjednika Beogradskog maratona, tj. onog takmičara koji je najkraće trčao. U slučaju da ima više takvih takmičara, vratiti onog sa najmanjim indeksom.

Napisati program koji testira rad zadatih funkcija. Maksimalan broj takmičara je 1000. U slučaju neispravnog unosa, ispisati odgovarajuću poruku o grešci.

#### Primer 1

```
INTERAKCIJA SA PROGRAMOM:
Unesite dimenziju niza:
5
19 47 27 34 16
Vreme trcanja takmicara: 19 47 27 34 16
Ukupno vreme: 143
Prosecno vreme trcanja: 28.6
Maksimalno vreme trcanja: 47
Indeks pobjednika: 4
```

#### Primer 2

```
INTERAKCIJA SA PROGRAMOM:
Unesite dimenziju niza:
-5
Greska: neispravan unos.
```

[Rešenje 3.1.3]

**Zadatak 3.1.4** Napisati funkciju `int min_max(int a[], int n)` koja pronalazi indekse najmanjeg i najvećeg elementa u nizu  $a$  dimenzije  $n$  koristeći samo jedan prolaz kroz niz. Funkcija kao povratnu vrednost vraća manji od ta dva indeksa. Napisati program koji testira ovu funkciju za učitane nizove celih brojeva maksimalne dužine 100 elemenata. U slučaju neispravnog unosa, ispisati odgovarajuću poruku o grešci.

#### Primer 1

```
INTERAKCIJA SA PROGRAMOM:
Unesite broj elemenata niza:
7
5 8 -4 11 17 89 1
2
```

#### Primer 2

```
INTERAKCIJA SA PROGRAMOM:
Unesite broj elemenata niza:
3
Unesite elemente niza:
9 11 6
1
```

#### Primer 3

```
INTERAKCIJA SA PROGRAMOM:
Unesite broj elemenata niza:
-45
Greska: neispravan unos.
```

**Zadatak 3.1.5** Napisati funkciju `int prebrojavanje(int a[], int n)` koja izračunava broj elemenata celobrojnog niza  $a$  dužine  $n$  koji su manji od poslednjeg elementa niza. Napisati i program koji testira rad funkcije. Maksimalan broj

### 3 Predstavljanje podataka

---

elemenata niza je 100. U slučaju neispravnog unosa, ispisati odgovarajuću poruku o grešci.

#### Primer 1

```
INTERAKCIJA SA PROGRAMOM:
Unesite broj elemenata niza: 4
Unesite elemente niza: 11 2 4 9
2
```

#### Primer 2

```
INTERAKCIJA SA PROGRAMOM:
Unesite broj elemenata niza: 7
Unesite elemente niza: 7 2 1 14 65 2 8
4
```

#### Primer 3

```
INTERAKCIJA SA PROGRAMOM:
Unesite broj elemenata niza: 5
Unesite elemente niza: 25 18 29 30 14
0
```

#### Primer 4

```
INTERAKCIJA SA PROGRAMOM:
Unesite broj elemenata niza:
-45
Greska: neispravan unos.
```

[Rešenje 3.1.5]

**Zadatak 3.1.6** Napisati funkciju `int prebrojavanje(int a[], int n)` koja izračunava broj parnih elemenata niza celih brojeva  $a$  dužine  $n$  koji pretihode maksimalnom elementu niza. Napisati i program koji testira rad funkcije. Maksimalan broj elemenata niza je 100. U slučaju neispravnog unosa, ispisati odgovarajuću poruku o grešci.

#### Primer 1

```
INTERAKCIJA SA PROGRAMOM:
Unesite broj elemenata niza: 4
Unesite elemente niza: 11 2 4 9
0
```

#### Primer 2

```
INTERAKCIJA SA PROGRAMOM:
Unesite broj elemenata niza: 7
Unesite elemente niza: 7 2 1 14 65 2 8
2
```

#### Primer 3

```
INTERAKCIJA SA PROGRAMOM:
Unesite broj elemenata niza: 5
Unesite elemente niza: 25 18 29 30 14
1
```

[Rešenje 3.1.6]

**Zadatak 3.1.7** Elementi niza celih brojeva su podaci o nadmorskim visinama u nekom području sveta. Na kartografskoj mapi su indeksima označene različite tačke, a u nizu je svakom indeksu pridružen neki ceo broj (odnosno nadmorska visina). Napisati funkcije za rad sa ovim nizom.

- (a) Napisati funkciju koja proverava da li niz sadrži zadatu nadmorsku visinu  $m$ . Povratna vrednost funkcije je 1 ako je vrednost sadržana u nizu ili 0 ako nije.

- (b) Napisati funkciju koja vraća vrednost prve pozicije na kojoj se nalazi element koji ima nadmorsku visinu  $m$  ili  $-1$  ukoliko element nije u nizu.
- (c) Napisati funkciju koja vraća vrednost poslednje pozicije na kojoj se nalazi element koji ima nadmorsku visinu  $m$  ili  $-1$  ukoliko element nije u nizu.

Napisati i program koji testira rad napisanih funkcija za uneti broj  $m$ . Maksimalan broj elemenata niza je 100. U slučaju neispravnog unosa, ispisati odgovarajuću poruku o grešci.

#### Primer 1

```
INTERAKCIJA SA PROGRAMOM:
Unesite dimenziju niza:
7
800 1100 -200 1400 -200 1100 800
Ucitani niz: 800 1100 -200 1400 -200 1100 800
Unesite jedan ceo broj:
1100
Niz sadrzi element cija je vrednost 1100.
Indeks njegovog prvog pojavljivanja u nizu je 1.
Indeks njegovog poslednjeg pojavljivanja u nizu je 5.
```

#### Primer 2

```
INTERAKCIJA SA PROGRAMOM:
Unesite dimenziju niza:
-5
Greska: neispravan unos.
```

[Rešenje 3.1.7]

**Zadatak 3.1.8** Marko skuplja sličice za Svetsko prvenstvo u fudbalu. U celobrojnom nizu  $a$  se nalaze brojevi onih sličica koje je već sakupio. Marko je primetio da mu se neke sličice ponavljaju i rešio je da ih razmeni sa drugarima. Napisati program koji od datog niza  $a$  formira niz  $b$  sličica koje se u nizu  $a$  pojavljuju više od dva puta. Maksimalan broj elemenata niza je 100. U slučaju neispravnog unosa, ispisati odgovarajuću poruku o grešci.

#### Primer 1

```
INTERAKCIJA SA PROGRAMOM:
Unesite broj elemenata niza: 8
Unesite elemente niza a:
4 11 4 6 8 4 6 6
Elementi niza b: 4 6
```

#### Primer 2

```
INTERAKCIJA SA PROGRAMOM:
Unesite broj elemenata niza: 13
Unesite elemente niza a:
8 26 7 2 1 1 7 2 2 2 7 5 1
Elementi niza b: 7 2 1
```

#### Primer 3

```
INTERAKCIJA SA PROGRAMOM:
Unesite broj elemenata niza: 2
Unesite elemente niza a:
9 5
Elementi niza b:
```

[Rešenje 3.1.8]

**Zadatak 3.1.9** Sa standardnog ulaza se učitava dimenzija niza, elementi niza i jedan ceo broj  $k$ . Napisati program koji štampa indekse elemenata koji su deljivi sa  $k$ . Maksimalan broj elemenata niza je 100. U slučaju neispravnog unosa, ispisati odgovarajuću poruku o grešci.

#### Primer 1

```
INTERAKCIJA SA PROGRAMOM:
Unesite dimenziju niza: 4
Unesite elemente niza: 10 14 86 20
Unesite broj k: 5
0 3
```

#### Primer 2

```
INTERAKCIJA SA PROGRAMOM:
Unesite dimenziju niza: 4
Unesite elemente niza: 6 14 8 9
Unesite broj k: 5
U nizu nema elemenata koji su
deljivi brojem 5.
```

#### Primer 3

```
INTERAKCIJA SA PROGRAMOM:
Unesite dimenziju niza: 6
Unesite elemente niza: 8 9 11 -4 8 11
Unesite broj k: 2
0 3 4
```

[Rešenje 3.1.9]

**Zadatak 3.1.10** Autobusi su označeni rednim brojevima (počevši od 1) i u nizu se čuva vreme putovanja svakog autobusa u minutima. Međutim, zbog radova na putu između Požege i Užica, svi autobusi koji saobraćaju na tom potezu (autobusi označeni rednim brojevima od  $k$  do  $t$ ) saobraćaju  $m$  minuta duže. Uneti potrebne izmene u niz i ispisati elemente niza. Maksimalan broj autobusa je 200. U slučaju neispravnog unosa, ispisati odgovarajuću poruku o grešci.

*Primer 1*

```

INTERAKCIJA SA PROGRAMOM:
Unesite dimenziju niza:
8
Unesite vreme putovanja:
24 78 13 124 56 90 205 45
Unesite redne brojeve autobusa koji
putuju na potezu Pozega, Uzice:
3 6
Unesite novo vreme:
23
Vreme putovanja nakon izmena:
24 78 36 147 79 113 205 45

```

*Primer 2*

```

INTERAKCIJA SA PROGRAMOM:
Unesite dimenziju niza:
8
Unesite vreme putovanja:
24 78 13 124 56 90 205 45
Unesite redne brojeve autobusa koji
putuju na potezu Pozega, Uzice:
3 15
Redni brojevi autobusa nisu
u dozvoljenom opsegu.

```

[Rešenje 3.1.10]

**Zadatak 3.1.11** Napisati funkciju `int zbir(int a[], int n, int i, int j)` koja računa zbir elemenata niza celih brojeva  $a$  dužine  $n$  od pozicije  $i$  do pozicije  $j$ . Napisati i program koji testira rad funkcije. Maksimalan broj elemenata niza je 100. U slučaju neispravnog unosa, ispisati odgovarajuću poruku o grešci.

*Primer 1*

```

INTERAKCIJA SA PROGRAMOM:
Unesite broj elemenata niza: 5
Unesite elemente niza: 11 5 6 48 8
Unesite vrednosti za i i j: 0 2
Zbir je: 22

```

*Primer 2*

```

INTERAKCIJA SA PROGRAMOM:
Unesite broj elemenata niza: 3
Unesite elemente niza: -2 8 1
Unesite vrednosti za i i j: 8 12
Greska: neispravan unos.

```

*Primer 3*

```

INTERAKCIJA SA PROGRAMOM:
Unesite broj elemenata niza: 7
Unesite elemente niza: -2 5 9 11 6 -3 -4
Unesite vrednosti za i i j: 2 5
Zbir: 23

```

[Rešenje 3.1.11]

**Zadatak 3.1.12** Napisati program koji transformiše uneti niz tako što kvadrira sve negativne elemente niza. Maksimalan broj elemenata niza je 100. U slučaju neispravnog unosa, ispisati odgovarajuću poruku o grešci.

#### Primer 1

```
INTERAKCIJA SA PROGRAMOM:
Unesite broj elemenata niza: 6
Unesite elemente niza:
12.34 -6 1 8 32.4 -16
12.34 36 1 8 32.4 256
```

#### Primer 2

```
INTERAKCIJA SA PROGRAMOM:
Unesite broj elemenata niza: 9
Unesite elemente niza:
-8.25 6 17 2 -1.5 1 -7 2.65 -125.2
68.0625 6 17 2 2.25 1 49 2.65 15675.04
```

#### Primer 3

```
INTERAKCIJA SA PROGRAMOM:
Unesite broj elemenata niza: 4
Unesite elemente niza:
9.53 5 1 4.89
9.53 5 1 4.89
```

[Rešenje 3.1.12]

**Zadatak 3.1.13** Napisati funkciju `void kvadriranje(float a[], int n)` koja kvadrira elemente realnog niza  $a$  dužine  $n$  koji se nalaze na parnim pozicijama. Napisati program koji transformiše na ovaj način uneti niz. Maksimalan broj elemenata niza je 100. U slučaju neispravnog unosa, ispisati odgovarajuću poruku o grešci.

#### Primer 1

```
INTERAKCIJA SA PROGRAMOM:
Unesite broj elemenata niza: 8
Unesite elemente niza:
2.34 1 -12.7 5.2 -8 -6.2 7 14.2
5.4756 1 161.29 5.2 64 -6.2 49 14.2
```

#### Primer 2

```
INTERAKCIJA SA PROGRAMOM:
Unesite broj elemenata niza: 3
Unesite elemente niza:
-6 -8.14 -15
36 -8.14 225
```

#### Primer 3

```
INTERAKCIJA SA PROGRAMOM:
Unesite broj elemenata niza: 1
Unesite elemente niza:
-35.11
1232.71
```

[Rešenje 3.1.13]

**Zadatak 3.1.14** Napisati funkciju `float zbir_pozitivnih(float a[], int n, int k)` koja izračunava zbir prvih  $k$  pozitivnih elemenata realnog niza  $a$  dužine  $n$ . Napisati i program koji testira rad funkcije. Maksimalan broj elemenata niza je 100. U slučaju neispravnog unosa, ispisati odgovarajuću poruku o grešci.

*Primer 1*

```

INTERAKCIJA SA PROGRAMOM:
Unesite broj elemenata niza: 8
Unesite elemente niza:
2.34 1 -12.7 5.2 -8 -6.2 7 14.2
Unesite vrednost za k: 3
Zbir je: 8.54

```

*Primer 2*

```

INTERAKCIJA SA PROGRAMOM:
Unesite broj elemenata niza: 3
Unesite elemente niza:
-6.598 -8.14 -15
Unesite vrednost za k: 4
Zbir je: 0.00

```

*Primer 3*

```

INTERAKCIJA SA PROGRAMOM:
Unesite broj elemenata niza: 7
Unesite elemente niza:
-35.11 5.29 -1.98 12.1 12.2 -3.33 -4.17
Unesite vrednost za k: 15
Zbir: 29.59

```

[Rešenje 3.1.14]

**Zadatak 3.1.15** Napisati funkciju `int blizu_3(int a[], int n)` koja pronalazi i vraća indeks elementa niza koji je po vrednosti najbliži aritmetičkoj sredini onih elemenata niza koji su deljivi brojem tri. Napisati program koji testira rad funkcije. Maksimalan broj elemenata niza je 200. U slučaju neispravnog unosa, ispisati odgovarajuću poruku o grešci.

*Primer 1*

```

INTERAKCIJA SA PROGRAMOM:
Unesite broj elemenata niza: 5
Unesite elemente niza:
1 2 3 4 5
3

```

*Primer 2*

```

INTERAKCIJA SA PROGRAMOM:
Unesite broj elemenata niza: 5
Unesite elemente niza:
3 6 2 4 7
4

```

[Rešenje 3.1.15]

**Zadatak 3.1.16** Napisati program koji za učitani ceo broj, ispisuje broj pojavljivanja svake od cifara u zapisu tog broja. UPUTSTVO: Za evidenciju broja pojavljivanja svake cifre pojedinačno, koristiti niz.

*Primer 1*

```

INTERAKCIJA SA PROGRAMOM:
Unesite ceo broj:
2355623
U zapisu broja 2355623, cifra 2 se pojaviljuje 2 puta
U zapisu broja 2355623, cifra 3 se pojaviljuje 2 puta
U zapisu broja 2355623, cifra 5 se pojaviljuje 2 puta
U zapisu broja 2355623, cifra 6 se pojaviljuje 1 puta

```

### 3 Predstavljanje podataka

---

#### Primer 2

```
INTERAKCIJA SA PROGRAMOM:
Unesite ceo broj:
-39902
U zapisu broja -39902, cifra 0 se pojaviljuje 1 puta
U zapisu broja -39902, cifra 2 se pojaviljuje 1 puta
U zapisu broja -39902, cifra 3 se pojaviljuje 1 puta
U zapisu broja -39902, cifra 9 se pojaviljuje 2 puta
```

[Rešenje 3.1.16]

**Zadatak 3.1.17** Napisati funkciju `int cifre(char s[], int n)` koja izračunava broj cifara u nizu karaktera *a* dužine *n*. Napisati program koji za karaktere koji se unose u zasebnim redovima ispisuje broj unetih cifara. Maksimalan broj elemenata niza je 100. U slučaju neispravnog unosa, ispisati odgovarajuću poruku o grešci.

#### Primer 1

```
INTERAKCIJA SA PROGRAMOM:
Unesite broj elemenata niza: 5
Unesite elemente niza:
4
+
A
u
8
Broj cifara je: 2
```

#### Primer 2

```
INTERAKCIJA SA PROGRAMOM:
Unesite broj elemenata niza: 7
Unesite elemente niza:
J
M
a
5
5
-
2
Broj cifara je: 3
```

#### Primer 3

```
INTERAKCIJA SA PROGRAMOM:
Unesite broj elemenata niza: 3
Unesite elemente niza:
e
k
F
Broj cifara je: 0
```

[Rešenje 3.1.17]

**Zadatak 3.1.18** Napisati program koji učitava karaktere sa standardnog ulaza sve do kraja ulaza i izračunava koliko se puta u unetom tekstu pojavila svaka od cifara, svako malo slovo i svako veliko slovo. Ispisati broj pojavljivanja samo za karaktere koji su se u unetom tekstu pojavili barem jednom. UPUTSTVO:



Za evidenciju broja pojavljivanja cifara, malih i velikih slova koristiti pojedinačne nizove.

#### Primer 1

```
INTERAKCIJA SA PROGRAMOM:
123 abcabcabc 123
Karakter 1 se pojavljuje 2 puta
Karakter 2 se pojavljuje 2 puta
Karakter 3 se pojavljuje 2 puta
Karakter a se pojavljuje 3 puta
Karakter b se pojavljuje 3 puta
Karakter c se pojavljuje 3 puta
```

#### Primer 2

```
INTERAKCIJA SA PROGRAMOM:
Programiranje 1 je zanimljivo!!
Karakter 1 se pojavljuje 1 puta
Karakter a se pojavljuje 3 puta
Karakter e se pojavljuje 2 puta
Karakter g se pojavljuje 1 puta
Karakter i se pojavljuje 3 puta
Karakter j se pojavljuje 3 puta
Karakter l se pojavljuje 1 puta
Karakter m se pojavljuje 2 puta
Karakter n se pojavljuje 2 puta
Karakter o se pojavljuje 2 puta
Karakter r se pojavljuje 3 puta
Karakter v se pojavljuje 1 puta
Karakter z se pojavljuje 1 puta
Karakter P se pojavljuje 1 puta
```

[Rešenje 3.1.18]

**Zadatak 3.1.19** Sa standardnog ulaza se unosi jedna linija teksta. Napisati program koji izračunava i ispisuje koliko puta se pojavilo svako od slova engleskog alfabeta u unetom tekstu. Ne praviti razliku između malih i velikih slova.

#### Primer 1

```
INTERAKCIJA SA PROGRAMOM:
Tasi, tasi, TaNaNa i SVILENA marama.....
a:9 b:0 c:0 d:0 e:1 f:0 g:0 h:0 i:4 j:0 k:0 l:1 m:2
n:3 o:0 p:0 q:0 r:1 s:3 t:3 u:0 v:1 w:0 x:0 y:0 z:0
```

#### Primer 2

```
INTERAKCIJA SA PROGRAMOM:
Mihailo Petrovic Alas (6 Maj 1868 - 8 Jun 1943)
a:4 b:0 c:1 d:0 e:1 f:0 g:0 h:1 i:3 j:2 k:0 l:2 m:2
n:1 o:2 p:1 q:0 r:1 s:1 t:1 u:1 v:1 w:0 x:0 y:0 z:0
```

#### Primer 3

```
INTERAKCIJA SA PROGRAMOM:
Alan Matison Turing (London, 23. jun 1912 - Cesir, 7. jun 1954)
a:3 b:0 c:1 d:1 e:1 f:0 g:1 h:0 i:3 j:3 k:0 l:2 m:1
n:7 o:3 p:0 q:0 r:2 s:2 t:2 u:3 v:0 w:0 x:0 y:0 z:0
```

[Rešenje 3.1.19]

### 3 Predstavljanje podataka

---

**Zadatak 3.1.20** Napisati program koji učitane karaktere (najviše njih 100, učitavaju se sve do pojave karaktera *\**) ispisuje u redosledu suprotnom od redosleda čitanja. U slučaju neispravnog unosa, ispisati odgovarajuću poruku o grešci.

#### Primer 1

```
INTERAKCIJA SA PROGRAMOM:
Unesite karakter: a
Unesite karakter: 8
Unesite karakter: 5
Unesite karakter: Y
Unesite karakter: I
Unesite karakter: o
Unesite karakter: ?
Unesite karakter: *
? o I Y 5 8 a
```

#### Primer 2

```
INTERAKCIJA SA PROGRAMOM:
Unesite karakter: g
Unesite karakter: g
Unesite karakter: 2
Unesite karakter: 2
Unesite karakter: )
Unesite karakter: )
Unesite karakter: *
) ) 2 2 g g
```

#### Primer 3

```
INTERAKCIJA SA PROGRAMOM:
Unesite karakter: U
Unesite karakter: 4
Unesite karakter: a
Unesite karakter: u
Unesite karakter: *
u a 4 U
```

[Rešenje 3.1.20]

**Zadatak 3.1.21** Palindrom je tekst koji se isto čita i sa leve i sa desne strane. Napisati funkciju koja proverava da li elementi niza karaktera čine palindrom (zanemariti velika/mala slova). Maksimalan broj elemenata niza je 200. U slučaju neispravnog unosa, ispisati odgovarajuću poruku o grešci.

#### Primer 1

```
INTERAKCIJA SA PROGRAMOM:
Unesite dimenziju niza:
15
Unesite elemente niza:
AnaVoliMilovana
Jeste palindrom.
```

#### Primer 2

```
INTERAKCIJA SA PROGRAMOM:
Unesite dimenziju niza:
26
Unesite elemente niza:
Zanimljivo je programirati!
Nije palindrom.
```

[Rešenje 3.1.21]

**Zadatak 3.1.22** Napisati program koji učitava dimenziju i elemente niza i štampa niz u kojem su najveći i najmanji element niza razmenili mesta. Ukoliko se najmanji ili najveći element više puta pojavljuju u nizu, uzeti u obzir njihova prva

pojavljivanja. Maksimalan broj elemenata niza je 100. U slučaju neispravnog unosa, ispisati odgovarajuću poruku o grešci.

*Primer 1*

```
INTERAKCIJA SA PROGRAMOM:
Unesite dimenziju niza: 5
Unesite elemente niza: 8 -2 11 19 4
8 19 11 -2 4
```

*Primer 2*

```
INTERAKCIJA SA PROGRAMOM:
Unesite dimenziju niza: 10
Unesite elemente niza:
46 -2 51 8 -5 66 2 8 3 14
46 -2 51 8 66 -5 2 8 3 14
```

*Primer 3*

```
INTERAKCIJA SA PROGRAMOM:
Unesite dimenziju niza: 145
Greska: neispravan unos.
```

[Rešenje 3.1.22]

**Zadatak 3.1.23** Korišćenjem nizova moguće je predstaviti skupove podataka. Napisati program koji demonstrira osnovne operacije nad skupovima — unija, presek i razlika. Pomoću dva niza predstaviti dva skupa celih brojeva. Maksimalan broj elemenata niza je 500. U slučaju neispravnog unosa, ispisati odgovarajuću poruku o grešci.

*Primer 1*

```
INTERAKCIJA SA PROGRAMOM:
Unesite broj elemenata niza a: 5
Unesite elemente niza a: 2 8 1 5 2
Unesite broj elemenata niza b: 3
Unesite elemente niza b: 5 7 8
Unija: 2 8 1 5 2 5 7 8
Presek: 5
Razlika: 2 1 2
```

*Primer 2*

```
INTERAKCIJA SA PROGRAMOM:
Unesite broj elemenata niza a: 3
Unesite elemente niza a: 11 4 4
Unesite broj elemenata niza b: 2
Unesite elemente niza b: 18 9
Unija: 11 4 4 18 9
Presek:
Razlika: 11 4 4
```

*Primer 3*

```
INTERAKCIJA SA PROGRAMOM:
Unesite broj elemenata niza a: 6
Unesite elemente niza a: 12 7 9 12 5 1
Unesite broj elemenata niza b: 4
Unesite elemente niza b: 1 12 22 12
Unija: 12 7 9 12 5 1 1 12 22 12
Presek: 12 12 1
Razlika: 7 9 5
```

[Rešenje 3.1.23]

### 3 Predstavljanje podataka

---

**Zadatak 3.1.24** Napisati program koji za dva učitana niza  $a$  i  $b$  dimenzije  $n$  formira i na izlaz ispisuje niz  $c$  koji se dobija naizmeničnim raspoređivanjem elemenata nizova  $a$  i  $b$ , tj.  $c = [a_0, b_0, a_1, b_1, \dots, a_{n-1}, b_{n-1}]$ . Maksimalan broj elemenata niza je 100. U slučaju neispravnog unosa, ispisati odgovarajuću poruku o grešci.

#### Primer 1

```
INTERAKCIJA SA PROGRAMOM:
Unesite dimenziju nizova:
5
Unesite elemente niza a:
2 -5 11 4 8
Unesite elemente niza b:
3 3 9 -1 17
Rezultujući niz:
2 3 -5 3 11 9 4 -1 8 17
```

#### Primer 2

```
INTERAKCIJA SA PROGRAMOM:
Unesite dimenziju nizova:
105
Greska: neispravan unos.
```

[Rešenje 3.1.24]

**Zadatak 3.1.25** Napisati program koji za dva učitana niza  $a$  i  $b$  dimenzije  $n$  formira i na izlaz ispisuje niz  $c$  čija prva polovina odgovara elementima niza  $b$ , a druga polovina elementima niza  $a$ , tj.  $c = [b_0, b_1, \dots, b_{n-1}, a_0, a_1, \dots, a_{n-1}]$ . Maksimalan broj elemenata niza je 100. U slučaju neispravnog unosa, ispisati odgovarajuću poruku o grešci.

#### Primer 1

```
INTERAKCIJA SA PROGRAMOM:
Unesite broj n: 3
Unesite elemente niza a: 4 -8 32
Unesite elemente niza b: 5 2 11
5 2 11 4 -8 32
```

#### Primer 2

```
INTERAKCIJA SA PROGRAMOM:
Unesite broj n: 4
Unesite elemente niza a: 1 0 -1 0
Unesite elemente niza b: 5 5 5 3
5 5 5 3 1 0 -1 0
```

#### Primer 3

```
INTERAKCIJA SA PROGRAMOM:
Unesite dimenziju niza: 145
Greska: neispravan unos.
```

[Rešenje 3.1.25]

\* **Zadatak 3.1.26** Sa standardnog ulaza se učitava ceo broj  $n$  manji od 100 i elementi dvaju celobrojnih, sortiranih neopadajuće nizova  $a$  i  $b$  dimenzije  $n$ . Napisati program koji formira i ispisuje niz  $c$  koji se dobija spajanjem nizova  $a$  i  $b$  u treći, takođe sortiran neopadajuće, niz. U slučaju neispravnog unosa, ispisati odgovarajuću poruku o grešci.

*Primer 1*

```

INTERAKCIJA SA PROGRAMOM:
Uneti broj elemenata niza: 5
Uneti elemente sortiranog niza:
2 11 28 40 63
Uneti elemente sortiranog niza:
-19 -5 5 11 52
Niz c:
-19 -5 2 5 11 11 28 40 52 63

```

*Primer 2*

```

INTERAKCIJA SA PROGRAMOM:
Uneti broj elemenata niza: 3
Uneti elemente sortiranog niza:
-2 4 8
Uneti elemente sortiranog niza:
6 15 19
Niz c:
-2 4 6 8 15 19

```

*Primer 3*

```

INTERAKCIJA SA PROGRAMOM:
Uneti broj elemenata niza: 145
Greska: neispravan unos.

```

[Rešenje 3.1.26]

**Zadatak 3.1.27** Napisati program koji sa standardnog ulaza učitava 10 celih brojeva i razdvaja ih na parne i neparne tako što parne brojeve upisuje na početak niza, a neparne brojeve na kraj niza. Ispisati niz dobijen na ovaj način. NAPOMENA: *Nije dozvoljeno koristiti pomoćne nizove.*

*Primer 1*

```

INTERAKCIJA SA PROGRAMOM:
Unesite 10 brojeva:
-2 8 11 53 59 20 17 -8 3 14
Rezultujući niz:
14 142 -6 -278 28 34 33 -69 -9 9

```

*Primer 2*

```

INTERAKCIJA SA PROGRAMOM:
Unesite 10 brojeva:
9 142 -9 -278 -69 33 34 28 -6 14
Rezultujući niz:
-2 8 14 -8 20 59 17 53 3 11

```

[Rešenje 3.1.27]

**Zadatak 3.1.28** Napisati funkcije za rad sa nizovima celih brojeva.

- Napisati funkciju koja obrće elemente niza.
- Napisati funkciju koja rotira niz ciklično za jedno mesto u levo.
- Napisati funkciju koja rotira niz ciklično za  $k$  mesta u levo.

Napisati i program koji testira rad napisanih funkcija za uneti broj  $m$ . Maksimalan broj elemenata niza je 100. U slučaju neispravnog unosa, ispisati odgovarajuću poruku o grešci.

#### Primer 1

```
INTERAKCIJA SA PROGRAMOM:
Unesite dimenziju niza:
6
7 -3 11 783 26 -19
Elementi niza nakon obrtanja:
-17 28 785 13 -1 9
Elementi niza nakon rotiranja za 1 mesto ulevo:
28 785 13 -1 9 -17
Unesite jedan pozitivan ceo broj:
3
Elementi niza nakon rotiranja za 3 mesto ulevo:
-1 9 -17 28 785 13
```

#### Primer 2

```
INTERAKCIJA SA PROGRAMOM:
Unesite dimenziju niza:
252
Greska: neispravan unos.
```

[Rešenje 3.1.28]

**Zadatak 3.1.29** Prilikom ulaska u banku klijent dobija redni broj, a u nizu se čuva redosled opsluživanja klijenata. Tako, prvi klijent u nizu će biti prvi uslužen, a klijent koji je poslednji dosao se nalazi na kraju niza. Redni brojevi se izdaju počevši od 1 svakog radnog dana, ali se niz za redosled stalno menja. Dodatno, postoje specijalni klijenti (npr. oni koji plaćaju platni promet ili oni koji podižu stambeni kredit) koji mogu dobiti i negativan redni broj da bi se razlikovali od uobičajenih usluga koje banka omogućava. Pomozite radniku obezbeđenja da lako prati redosled opsluživanja klijenata.

- (a) Napisati funkciju koja ubacuje datog klijenta sa rednim brojem  $x$  na kraj niza.
- (b) Napisati funkciju koja ubacuje datog klijenta sa rednim brojem  $x$  na početak niza (lica sa posebnim potrebama, trudnice, stara lica i ostale ugrožene kategorije).
- (c) Napisati funkciju koja ubacuje datog klijenta sa rednim brojem  $x$  na datu poziciju  $k$  (manje prioritetna lica, recimo službena lica ili roditelji sa decom, poziciju  $k$  bira radnik obezbeđenja).
- (d) Napisati funkciju koja izbacuje prvi element niza (usluženi klijent).
- (e) Napisati funkciju koja izbacuje poslednji element niza (klijent je odustao jer je shvatio da ima mnogo klijenata ispred njega).
- (f) Napisati funkciju koja izbacuje element sa date pozicije  $k$  (klijent je odustao jer je dugo čekao).

Napisati program koji testira rad funkcija. Maksimalan broj klijenata u jednom danu je 2000. U slučaju neispravnog unosa, ispisati odgovarajuću poruku o grešci.

*Primer 1*

```

INTERAKCIJA SA PROGRAMOM:
Unesite trenutni broj klijenata:
8
Unesite niz sa rednim brojevima klijenata:
2 5 -2 16 33 19 8 11
Unesite klijenta kojeg treba ubaciti u niz:
35
Niz nakon ubacivanja klijenta: 2 5 -2 16 33 19 8 11 35
Unesite prioritnog klijenta kojeg treba ubaciti u niz:
36
Niz nakon ubacivanja klijenta: 36 2 5 -2 16 33 19 8 11 35
Unesite prioritnog klijenta kojeg treba ubaciti u niz i njegovu poziciju:
-6 2
Niz nakon ubacivanja klijenta: 36 2 -6 5 -2 16 33 19 8 11 35
Niz nakon odlaska klijenta: 2 -6 5 -2 16 33 19 8 11 35
Niz nakon odlaska poslednjeg klijenta: 2 -6 5 -2 16 33 19 8 11
Unesite redni broj klijenta koji je napustio red:
-2
Niz nakon odlaska klijenta: 2 -6 5 16 33 19 8 11

```

[Rešenje 3.1.29]

**Zadatak 3.1.30** Napisati program koji za učitani niz formira i ispisuje niz koji se dobija izbacivanjem svih neparnih elemenata niza. Zadatak rešiti na dva načina: korišćenjem pomoćnog niza i transformacijom polaznog niza. Maksimalan broj elemenata niza je 100. U slučaju neispravnog unosa, ispisati odgovarajuću poruku o grešci.

*Primer 1*

```

INTERAKCIJA SA PROGRAMOM:
Unesite broj elemenata niza: 4
Unesite elemente niza: 8 9 15 12
8 12

```

*Primer 2*

```

INTERAKCIJA SA PROGRAMOM:
Unesite broj elemenata niza: 6
Unesite elemente niza: 21 5 3 22 19 188
22 188

```

*Primer 3*

```

INTERAKCIJA SA PROGRAMOM:
Unesite broj elemenata niza: 4
Unesite elemente niza: 133 129 121 101

```

*Primer 4*

```

INTERAKCIJA SA PROGRAMOM:
Unesite broj elemenata niza: 8
Unesite elemente niza: 15 -22 -23 13 18 46 14 -31
-22 18 46 14

```

[Rešenje 3.1.30]

### 3 Predstavljanje podataka

---

**Zadatak 3.1.31** Napisati program koji učitava dimenziju  $n$  celobrojnog niza  $a$  i njegove elemente, i iz niza  $a$  izbacuje sve elemente koji nisu deljivi svojom poslednjom cifrom. Izuzetak su elementi čija je poslednja cifra nula i koje zbog toga treba zadržati. Program treba da ispiše izmenjeni niz na standardni izlaz. Maksimalan broj elemenata niza je 100. U slučaju neispravnog unosa, ispisati odgovarajuću poruku o grešci.

#### Primer 1

```
INTERAKCIJA SA PROGRAMOM:
Unesite dimenziju niza:
9
Unesite elemente niza a:
173 -25 23 7 17 25 34 61 -4612
Niz a nakon izmene:
-25 7 25 61 -4612
```

#### Primer 2

```
INTERAKCIJA SA PROGRAMOM:
Unesite dimenziju niza:
0
Greska: neispravan unos.
```

[Rešenje 3.1.31]

**Zadatak 3.1.32** Napisati program koji u nizu dužine  $n$  čiji se elementi učitavaju sa ulaza eliminiše sve brojeve koji nisu deljivi svojim indeksom. Niz reorganizovati tako da nema *rupa* koje su nastale eliminacijom elemenata i ispisati na standardni izlaz. Maksimalan broj elemenata niza je 700. Ne razmatrati da li je u novom nizu, nakon brisanja i pomeranja, element deljiv svojim indeksom. U slučaju neispravnog unosa, ispisati odgovarajuću poruku o grešci. NAPOMENA: *Nulti element niza treba zadržati jer nije dozvoljeno deljenje nulom.*

#### Primer 1

```
INTERAKCIJA SA PROGRAMOM:
Unesite broj elemenata niza: 10
Unesite elemente niza:
4 2 1 6 7 8 10 2 16 3
Novi niz:
4 2 6 16
```

#### Primer 2

```
INTERAKCIJA SA PROGRAMOM:
Unesite broj elemenata niza: 10
Unesite elemente niza:
-8 5 10 6 7 10 8 2 16 27
Novi niz:
-8 5 10 6 10 16 27
```

[Rešenje 3.1.32]

**Zadatak 3.1.33** Napisati program koji za učitani niz ispisuje niz koji se dobija izbacivanjem svih elemenata koji su prosti brojevi. Zadatak rešiti na dva načina: korišćenjem pomoćnog niza i transformacijom polaznog niza. Maksimalan broj elemenata niza je 100. NAPOMENA: *Broj 1 nije prost.* U slučaju neispravnog unosa, ispisati odgovarajuću poruku o grešci.



*Primer 1*

```

INTERAKCIJA SA PROGRAMOM:
Unesite broj elemenata niza: 5
Unesite elemente niza: 11 5 6 48 8
6 48 8

```

*Primer 2*

```

INTERAKCIJA SA PROGRAMOM:
Unesite broj elemenata niza: 4
Unesite elemente niza: 11 5 19 21
21

```

*Primer 3*

```

INTERAKCIJA SA PROGRAMOM:
Unesite broj elemenata niza: 5
Unesite elemente niza: 12 18 9 31 7
12 18 9

```

*Primer 4*

```

INTERAKCIJA SA PROGRAMOM:
Unesite broj elemenata niza: 3
Unesite elemente niza: -31 11 -19

```

*Primer 5*

```

INTERAKCIJA SA PROGRAMOM:
Unesite broj elemenata niza: 5
Unesite elemente niza: -2 15 -11 8 7
15 8

```

[Rešenje 3.1.33]

**Zadatak 3.1.34** Napisati funkciju koja proverava da li su elementi celobrojnog niza uređeni neopadajuće. Maksimalan broj elemenata niza je 300. U slučaju neispravnog unosa, ispisati odgovarajuću poruku o grešci.

*Primer 1*

```

INTERAKCIJA SA PROGRAMOM:
Unesite dimenziju niza:
8
Unesite elemente niza:
-40 -8 -8 2 30 30 46 200
Jeste uredjen neopadajuće.

```

*Primer 2*

```

INTERAKCIJA SA PROGRAMOM:
Unesite dimenziju niza:
4
Unesite elemente niza:
4 23 15 30
Nije uredjen neopadajuće.

```

[Rešenje 3.1.34]

**Zadatak 3.1.35** Svaki indeks niza označava jedan dan u mesecu, a elementi celobrojnog niza predstavljaju broj artikala koji se prodao tog dana. Naći koliko najduže je iz dana u dan broj prodatih artikala rastao. U slučaju neispravnog unosa, ispisati odgovarajuću poruku o grešci.

#### Primer 1

```
INTERAKCIJA SA PROGRAMOM:
Unesite dimenziju niza: 30
Unesite broj prodatih artikala:
89 171 112 67 119 36 181 157
49 96 73 116 21 172
140 0 23 71 157 135 11 166 21
56 56 87 103 183 148 174
Duzina najduzeg neopadajuceg
prodavanja je 6.
```

#### Primer 2

```
INTERAKCIJA SA PROGRAMOM:
Unesite dimenziju niza: 31
Unesite broj prodatih artikala:
215 223 262 95 18 116 334 97
146 146 19 314 270 115 21 40
253 27 210 68 96 175 41 242
98 163 8 218 107 102
Duzina najduzeg neopadajuceg
prodavanja je 3.
```

[Rešenje 3.1.35]

**Zadatak 3.1.36** Napisati funkciju koja određuje dužinu najduže serije jednakih uzastopnih elemenata u datom nizu brojeva. Maksimalan broj elemenata niza je 100. U slučaju neispravnog unosa, ispisati odgovarajuću poruku o grešci.

#### Primer 1

```
INTERAKCIJA SA PROGRAMOM:
Unesite dimenziju niza: 8
Unesite elemente niza:
9 -1 2 2 2 2 80 -200
Duzina najduze serije je 4.
```

#### Primer 2

```
INTERAKCIJA SA PROGRAMOM:
Unesite dimenziju niza: 8
Unesite elemente niza:
9 9 0 -3 -3 -3 -3 72
Duzina najduze serije je 4.
```

[Rešenje 3.1.36]

**Zadatak 3.1.37** Napisati funkciju koja određuje da li se jedan niz javlja kao podniz uzastopnih elemenata drugog niza. Zadatak rešiti na dva načina:

- (a) Tako da elementi jednog niza moraju da budu uzastopni u drugom nizu.
- (b) Tako da elementi jednog niza ne moraju da budu uzastopni u drugom nizu, ali je redosled pojavljivanja isti.

Maksimalan broj elemenata niza je 100. U slučaju neispravnog unosa, ispisati odgovarajuću poruku o grešci.

#### Primer 1

```
INTERAKCIJA SA PROGRAMOM:
Unesite dimenziju niza: 8
Unesite elemente niza:
-4 2 7 90 -22 15 14 7
Unesite dimenziju niza: 4
Unesite elemente niza: 90 -22 15 14
Elementi drugog niza cine
uzastopni podniz prvog niza.
Elementi drugog niza cine
podniz prvog niza.
```

#### Primer 2

```
INTERAKCIJA SA PROGRAMOM:
Unesite dimenziju niza: 8
Unesite elemente niza:
-4 2 7 90 -22 15 14 7
Unesite dimenziju niza: 4
Unesite elemente niza: 2 7 15 7
Elementi drugog niza ne cine
uzastopni podniz prvog niza.
Elementi drugog niza cine
podniz prvog niza.
```

## Primer 3

```

INTERAKCIJA SA PROGRAMOM:
Unesite dimenziju niza: 8
Unesite elemente niza:
-4 2 7 90 -22 15 14 7
Unesite dimenziju niza: 4
Unesite elemente niza: 90 -22 200 1
Elementi drugog niza ne cine
uzastopni podniz prvog niza.
Elementi drugog niza ne
cine podniz prvog niza.

```

## Primer 4

```

INTERAKCIJA SA PROGRAMOM:
Unesite dimenziju niza: 8
Unesite elemente niza:
-4 2 7 90 -22 15 14 7
Unesite dimenziju niza: 1
Unesite elemente niza: 90
Elementi drugog niza cine
uzastopni podniz prvog niza.
Elementi drugog niza cine
podniz prvog niza.

```

[Rešenje 3.1.37]

**Zadatak 3.1.38** Za celobrojni niz  $a$  dimenzije  $n$  kažemo da je *permutacija* ako sadrži sve brojeve od 1 do  $n$ .

- (a) Napisati funkciju `void brojanje(int a[], int b[], int n)` koja na osnovu celobrojnog niza  $a$  dimenzije  $n$  formira niz  $b$  tako što  $i$ -ti element niza  $b$  odgovara broju pojavljivanja vrednosti  $i$  u nizu  $a$ .
- (b) Napisati funkciju `int permutacija(int a[], int n)` koja proverava da li je zadati niz permutacija. Funkcija vraća vrednost 1 ako je svojstvo ispunjeno, odnosno 0 ako nije. UPUTSTVO: *Koristiti funkciju brojanje iz tačke (a).*

Napisati program koji sa standardnog ulaza učitava dimenziju niza i elemente niza i ispisuje da li je uneti niz permutacija ili ne. Maksimalan broj elemenata niza je 100. U slučaju neispravnog unosa, ispisati odgovarajuću poruku o grešci.

## Primer 1

```

INTERAKCIJA SA PROGRAMOM:
Unesite broj elemenata niza: 5
Unesite elemente niza:
1 5 4 3 2
Uneti niz je permutacija.

```

## Primer 2

```

INTERAKCIJA SA PROGRAMOM:
Unesite broj elemenata niza: 6
Unesite elemente niza:
2 3 3 1 1 5
Uneti niz nije permutacija.

```

[Rešenje 3.1.38]

**Zadatak 3.1.39** Napisati program koji za dva cela broja  $x$  i  $y$  koja se učitavaju sa standardnog ulaza proverava da li se zapisuju pomoću istih cifara. UPUTSTVO: *Rešiti korišćenjem nizova. Pogledati zadatak 2.5.15.*

#### Primer 1

```
INTERAKCIJA SA PROGRAMOM:  
Unesite dva broja: 251 125  
Brojevi se zapisuju istim ciframa.
```

#### Primer 2

```
INTERAKCIJA SA PROGRAMOM:  
Unesite dva broja: 8898 9988  
Brojevi se ne zapisuju istim ciframa.
```

#### Primer 3

```
INTERAKCIJA SA PROGRAMOM:  
Unesite dva broja: -7391 1397  
Brojevi se zapisuju istim ciframa.
```

[Rešenje 3.1.39]

## 3.2 Rešenja

### Rešenje 3.1.1

```
1 #include <stdio.h>  
2 #include <stdlib.h>  
  
4 /* Predprocesorska direktiva kojom se definise maksimalni broj  
   elemenata niza. */  
6 #define MAKS 100  
  
8 int main()  
{  
10 /* Deklaracija potrebnih promenljivih. */  
   int a[MAKS];  
12   int b[MAKS];  
   int n, i, skalarni_proizvod;  
14  
   /* Ucitava se dimenzija niza i vrsi se provera ispravnosti  
      ulaza. */  
16   printf("Unesite dimenziju vektora: ");  
18   scanf("%d", &n);  
   if (n < 1 || n > 100) {  
20     printf("Greska: neispravan unos.\n");  
     /* Za izlazak iz programa moze da se koristi i funkcija exit.  
        Argument EXIT_FAILURE oznacava da je doslo do neke greske  
        pri izvršavanju programa. */  
22     exit(EXIT_FAILURE);  
24   }  
26  
   /* Ucitavaju se koordinate vektora. */  
28   printf("Unesite koordinate vektora a: ");  
   for (i = 0; i < n; i++)
```

```

30     scanf("%d", &a[i]);

32     printf("Unesite koordinate vektora b: ");
    for (i = 0; i < n; i++)
34         scanf("%d", &b[i]);

36     /* Izracunava se skalarni proizvod po zadatoj formuli. */
    skalarni_proizvod = 0;
38     for (i = 0; i < n; i++)
        skalarni_proizvod = skalarni_proizvod + a[i] * b[i];

40     /* Ispis rezultata. */
42     printf("Skalarni proizvod vektora a i b: %d\n",
        skalarni_proizvod);

44     /* Kada se funkciji exit prosledi EXIT_SUCCESS to znaci da se
46     program uspesno zavrrio. Efekat je isti kao i da je navedeno
        return 0; na ovom mestu. */
48     exit(EXIT_SUCCESS);
}

```

### Rešenje 3.1.2

```

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

4 #define MAKS 100

6 int main()
{
8     /* Deklaracije potrebnih promenljivih. */
    int a[MAKS];
10    int n, i;

12    /* Ucitava se dimenzija niza i vrsi se provera ispravnosti
        ulaza. */
14    printf("Unesite dimenziju niza:\n");
    scanf("%d", &n);
16    if (n < 1 || n > MAKS) {
        printf("Greska: neispravan unos.\n");
18        exit(EXIT_FAILURE);
    }

20    /* Ucitavaju se elementi niza. */
22    printf("Unesi elemente niza:\n");
    for (i = 0; i < n; i++) {
24        scanf("%d", &a[i]);
    }

26    /* Ispisuju se elementi niza na parnim pozicijama. */
28    printf("Elementi niza na parnim pozicijama:\n");

```

### 3 Predstavljanje podataka

---

```
    for (i = 0; i < n; i += 2)
30     printf("%d ", a[i]);
    printf("\n");
32
    /* Ispisuju se parni elementi niza. */
34     printf("Parni elementi niza:\n");
    for (i = 0; i < n; i++)
36         if (a[i] % 2 == 0)
            printf("%d ", a[i]);
38     printf("\n");
40     exit(EXIT_SUCCESS);
}
```

#### Rešenje 3.1.3

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

4 #define MAKS 1000

6 /* Funkcija koja učitava elemente niza. */
void učitaj(int a[], int n)
8 {
    int i;
10     for (i = 0; i < n; i++)
        scanf("%d", &a[i]);
12 }

14 /* Funkcija koja ispisuje elemente niza. */
void stampaj(int a[], int n)
16 {
    int i;
18     for (i = 0; i < n; i++)
        printf("%d ", a[i]);
20     printf("\n");
}

22 /* Funkcija koja racuna sumu elemenata niza. */
24 int suma(int a[], int n)
{
    int i;
26     int suma_elemenata = 0;
28     for (i = 0; i < n; i++)
        suma_elemenata += a[i];
30     return suma_elemenata;
}

32
34 /* Funkcija koja racuna prosečnu vrednost elemenata niza. */
float prosek(int a[], int n)
{
    int i;
```

```
36     int suma_elementata = suma(a, n);
37     return (float) suma_elementata / n;
38 }

40 /* Funkcija koja izracunava maksimum elemenata niza. */
41 int maksimum(int a[], int n)
42 {
43     int najveci, i;
44
45     najveci = a[0];
46     for (i = 1; i < n; i++)
47         if (a[i] > najveci)
48             najveci = a[i];
49
50     return najveci;
51 }

52 /* Funkcija koja izracunava poziciju maksimalnog elementa u nizu. */
53 int pozicija_maksimuma(int a[], int n)
54 {
55     int najveci, pozicija_najveceg;
56     int i;
57
58     najveci = a[0];
59     pozicija_najveceg = 0;
60
61     for (i = 1; i < n; i++)
62         if (a[i] < najveci) {
63             najveci = a[i];
64             pozicija_najveceg = i;
65         }
66
67     return pozicija_najveceg;
68 }

69
70 int main()
71 {
72     /* Deklaracija potrebnih promenljivih. */
73
74     int a[MAKS];
75     int n;
76
77     /* Ucitava se dimenzija niza i vrsi se provera ispravnosti
78        ulaza. */
79     printf("Unesite dimenziju niza:");
80     scanf("%d", &n);
81     if (n < 1 || n > MAKS) {
82         printf("Greska: neispravan unos.\n");
83         exit(EXIT_FAILURE);
84     }
85
86     /* Testira se funkcija kojom se ucitavaju elementi niza. */
87     ucitaj(a, n);
```

### 3 Predstavljanje podataka

---

```
88      /* Testira se funkcija kojom se ispisuju elementi niza. */
89      printf("Vreme trcanja takmicara: ");
90      stampaj(a, n);
91
92      /* Testira se funkcija kojom se izracunava suma elemenata niza. */
93      printf("Ukupno vreme: %d\n", suma(a, n));
94
95      /* Testira se funkcija kojom se racuna prosek elemenata niza. */
96      printf("Prosecno vreme trcanja: %.2f\n", prosek(a, n));
97
98      /* Testira se funkcija kojom se izracunava minimum niza. */
99      printf("Maksimalno vreme trcanja: %d\n", maksimum(a, n));
100
101      /* Testira se funkcija kojom se izracunava pozicija maksimalnog
102         elementa. */
103      printf("Indeks pobednika: %d\n", pozicija_maksimuma(a, n));
104
105      exit(EXIT_SUCCESS);
106  }
```

#### Rešenje 3.1.5

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

#define MAKS 100

/* Funkcija prebrojavanje vraca broj elemenata niza koji su manji
   od poslednjeg elementa. */
int prebrojavanje(int a[], int n)
{
    int i;
    int broj_manjih = 0;

    /* Prebrojavaju se elementi niza za koje vazi da su manji od
       poslednjeg elementa (a[n-1]). Petlja ide od prvog do
       predposlednjeg elementa. */
    for (i = 0; i < n - 1; i++) {
        if (a[i] < a[n - 1])
            broj_manjih++;
    }

    return broj_manjih;
}

int main()
{
    /* Deklaracija potrebnih promenljivih. */
    int a[MAKS];
    int n, i;
```



```

30  /* Ucitava se broj elemenata niza i vrsi se provera ispravnosti
    ulaza. */
32  printf("Unesite broj elemenata niza:");
    scanf("%d", &n);
34  if (n <= 0 || n > MAKS) {
        printf("Greska: neispravan unos.\n");
36      exit(EXIT_FAILURE);
    }

38
    /* Ucitavaju se elementi niza. */
40  printf("Unesite elemente niza:");
    for (i = 0; i < n; i++)
42      scanf("%d", &a[i]);

44  /* Ispis rezultata. */
    printf("%d\n", prebrojavanje(a, n));
46
    exit(EXIT_SUCCESS);
48 }

```

### Rešenje 3.1.6

```

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

4  #define MAKS 100

6  /* Funkcija vraca poziciju najveceg elementa niza. */
    int pozicija_najveceg(int a[], int n)
8  {
        int i, pozicija = 0;
10     /* Prolazi se kroz niz i ako se naidje na element cija je
        vrednost veca od trenutno najveceg (a[pozicija]), vrsi
12     se azuriranje pozicije trenutno najveceg. */
        for(i=1; i<n; i++)
14         if(a[i] > a[pozicija])
            pozicija = i;
16
        return pozicija;
18 }

20 /* Funkcija vraca broj parnih elemenata niza koji prethode
    maksimalnom elementu niza. */
22 int prebrojavanje(int a[], int n)
    {
24         int i;
        int pozicija_maksimuma = pozicija_najveceg(a,n);
26
        int broj_parnih = 0;
28         for (i = 0; i < pozicija_maksimuma; i++) {

```

### 3 Predstavljanje podataka

```
        if (a[i] % 2 == 0) {
30            broj_parnih++;
        }
32    }

34    return broj_parnih;
}

36
37
38    int main()
39    {
40        /* Deklaracija potrebnih promenljivih. */
41        int a[MAKS];
42        int n, i;

43        /* Ucitava se broj elemenata niza i vrsi se provera ispravnosti
44           ulaza. */
45        printf("Unesite broj elemenata niza:");
46        scanf("%d", &n);
47        if (n <= 0 || n > MAKS) {
48            printf("Greska: neispravan unos.\n");
49            exit(EXIT_FAILURE);
50        }

51        /* Ucitavaju se elementi niza. */
52        printf("Unesite elemente niza:");
53        for (i = 0; i < n; i++) {
54            scanf("%d", &a[i]);
55        }
56    }

57    /* Ispis rezultata. */
58    printf("%d\n", prebrojavanje(a, n));
59    exit(EXIT_SUCCESS);
60
61
62 }
```

#### Rešenje 3.1.7

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

4 #define MAKS 100

6 /* Funkcija koja ucitava elemente niza. */
7 void ucitaj(int a[], int n)
8 {
9     int i;
10    for (i = 0; i < n; i++)
11        scanf("%d", &a[i]);
12 }

14 /* Funkcija koja ispisuje elemente niza. */
```

```
void stampaj(int a[], int n)
16 {
    int i;
18     for (i = 0; i < n; i++)
        printf("%d ", a[i]);
20     printf("\n");
}

/* Funkcija koja proverava da li niz sadrzi zadatu vrednost m. */
24 int sadrzi(int a[], int n, int m)
{
26     int i;
    /* Prolazi se kroz sve elemente niza i ukoliko se naidje na element
28     cija je vrednost jednaka m, kao povratna vrednost funkcije se
        vraca 1. */
30     for (i = 0; i < n; i++) {
        if (a[i] == m)
32         return 1;
    }

34     /* Ako se stigne do kraja niza i ne naidje na vrednost koja je
36     jednaka broju m, to znaci da se broj ne nalazi u nizu i da
        funkcija treba da vrati 0. */
38     return 0;
}

/* Funkcija koja vraca vrednost prve pozicije na kojoj se nalazi
42     element koji ima vrednost m, ili -1 ukoliko element nije u nizu.
    */
44 int prvo_pojavljivanje(int a[], int n, int m)
{
46     int i;
    for (i = 0; i < n; i++)
48         if (a[i] == m)
            return i;

50     /* Ako se stigne do kraja niza i ne naidje na vrednost koja je
52     jednaka broju m, to znaci da se broj ne nalazi u nizu i da
        funkcija treba da vrati -1. */
54     return -1;
}

/* Funkcija koja vraca vrednost poslednje pozicije na kojoj se
58     nalazi element koji ima vrednost m, ili -1 ukoliko element nije
        u nizu. */
60 int poslednje_pojavljivanje(int a[], int n, int m)
{
62     int i;

64     /* Polazi se od kraja niza i poredi se element po element sa
        zadatim brojem m. */
66     for (i = n - 1; i >= 0; i--)
```

### 3 Predstavljanje podataka

---

```
        if (a[i] == m)
68             return i;

70     /* Ako se stigne do pocetka niza i ne naidje na vrednost koja je
        jednaka broju m, to znaci da se broj ne nalazi u nizu i da
72     funkcija treba da vrati -1. */
    return -1;
74 }

76 int main()
{
78     /* Deklaracije potrebnih promenljivih. */
    int a[MAKS];
80     int n, m, i;

82     /* Ucitava se dimenzija niza i vrsi se provera ispravnosti
        ulaza. */
84     printf("Unesite dimenziju niza:");
    scanf("%d", &n);
86     if (n < 1 || n > MAKS) {
        printf("Greska: neispravan unos.\n");
88         exit(EXIT_FAILURE);
    }

90     /* Ucitavaju se i ispisuju elementi niza. */
92     ucitaj(a, n);
    printf("Ucitani niz:");
94     stampaj(a, n);

96     /* Ucitava se vrednost za pretragu. */
    printf("Unesi jedan ceo broj:");
98     scanf("%d", &m);

100    /* Testira se rad napisanih funkcija. */
    if (sadrzi(a, n, m))
102        printf("Niz sadrzi element cija je vrednost %d\n", m);
    else
104        printf("Niz ne sadrzi element cija je vrednost %d\n", m);

106    i = prvo_pojavljivanje(a, n, m);
    if (i != -1)
108        printf("Indeks njegovog prvog pojavljivanja u "
                "nizu je %d\n", i);
110

112    i = poslednje_pojavljivanje(a, n, m);
    if (i != -1)
114        printf("Indeks njegovog poslednjeg pojavljivanja u "
                "nizu je %d\n", i);
116    exit(EXIT_SUCCESS);
118 }
```

## Rešenje 3.1.8

```
1  #include <stdio.h>
2  #include <stdlib.h>
3
4
5  #define MAKS 100
6
7  /* Funkcija koja vraca broj pojavljivanja broja x u nizu. */
8  int broj_pojavljivanja(int niz[], int n, int x)
9  {
10     int i;
11     int brojac_pojavljivanja = 0;
12
13     for (i = 0; i < n; i++) {
14         if (niz[i] == x) {
15             brojac_pojavljivanja++;
16         }
17     }
18
19     return brojac_pojavljivanja;
20 }
21
22 int main()
23 {
24     /* Deklaracija potrebnih promenljivih. */
25     int a[MAKS], rezultujuci_niz[MAKS];
26     int i, n, j, duzina_rezultujuceg_niza;
27
28     /* Ucitava se dimenzija niza i vrsi se provera ispravnosti
29        ulaza. */
30     printf("Unesite broj n: ");
31     scanf("%d", &n);
32     if (n < 1 || n > MAKS) {
33         printf("Greska: neispravan unos.\n");
34         exit(EXIT_FAILURE);
35     }
36
37     /* Ucitavaju se elementi niza. */
38     printf("Unesite elemente niza a: ");
39     for (i = 0; i < n; i++)
40         scanf("%d", &a[i]);
41
42     /* Parametar j je brojac elemenata rezultujuceg niza. */
43     j = 0;
44
45     /* Obilazi se element po element niza a. */
46     for (i = 0; i < n; i++) {
47         /* Ukoliko se tekuci element pojavljuje vise od dva puta u nizu
48            a i nije upisan u rezultujuci niz koji trenutno ima j
49            elemenata, dodaje se u rezultujuci niz na poziciju j i
50            uvecava se broj elemenata rezultujuceg niza. */
51         if (broj_pojavljivanja(a, n, a[i]) >= 3
```

### 3 Predstavljanje podataka

```
51         && broj_pojavljivanja(rezultujuci_niz, j, a[i]) == 0) {
52             rezultujuci_niz[j] = a[i];
53             j++;
54         }
55     }

57     /* Ispis rezultata. */
58     duzina_rezultujuceg_niza = j;
59     for (i = 0; i < duzina_rezultujuceg_niza; i++)
60         printf("%d ", rezultujuci_niz[i]);
61     printf("\n");

63     exit(EXIT_SUCCESS);
64 }
```

#### Rešenje 3.1.9

```
1  #include <stdio.h>
2  #include <stdlib.h>
3
4  #define MAKS 100
5
6  int main()
7  {
8      /* Deklaracija potrebnih promenljivih. */
9      int brojevi[MAKS];
10     int n, i, k, indikator;
11
12     /* Ucitava se dimenzija niza i vrsi se provera ispravnosti
13        ulaza. */
14     printf("Unesite dimenziju niza: ");
15     scanf("%d", &n);
16     if (n < 1 || n > MAKS) {
17         printf("Greska: neispravan unos.\n");
18         exit(EXIT_FAILURE);
19     }
20
21     /* Ucitavaju se elementi niza. */
22     printf("Unesite elemente niza: ");
23     for (i = 0; i < n; i++)
24         scanf("%d", &brojevi[i]);
25
26     /* Ucitava se broj k i proverava se njegova ispravnost. */
27     printf("Unesite broj k: ");
28     scanf("%d", &k);
29     if (k == 0) {
30         printf("Greska: neispravan unos.\n");
31         exit(EXIT_FAILURE);
32     }
33
34     /* Promenljiva koja cuva informaciju o tome da li je u nizu
```

```

35     postojao element koji je deljiv brojem k. Inicijalna vrednost
        je 0. */
37     indikator = 0;

39     /* Ukoliko je element niza deljiv brojem k, indikator se
        postavlja na 1 i ispisuje se indeks tog elementa. */
41     for (i = 0; i < n; i++) {
42         if (brojevi[i] % k == 0) {
43             indikator = 1;
44             printf("%d ", i);
45         }
46     }

47     /* Ukoliko je indikator jednak nuli to znaci da ne postoji
        element u nizu koji je deljiv brojem k. */
49     if (indikator == 0)
51         printf("U nizu nema elemenata koji su deljivi brojem %d.\n", k);

53     exit(EXIT_SUCCESS);
}

```

### Rešenje 3.1.10

```

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

4 /* Indeksiranje pocinje od 1, pa zato maksimalna dimenzija niza
    mora biti 201, a ne 200. */
6 #define MAKS 201

8 int main()
{
10     /* Deklaracija potrebnih promenljivih. */
    int n, niz[MAKS], i;
12     int k, t, kasnjenje;

14     /* Ucitava se dimenzija niza. */
    printf("Unesite dimenziju niza: ");
16     scanf("%d", &n);

18     /* Vrsi se provera ispravnosti ulaza. */
    if (n <= 0 || n > MAKS) {
20         printf("Greska: neispravan unos.\n");
        exit(EXIT_FAILURE);
22     }

24     /* Ucitavaju se vremena putovanja. */
    printf("Unesite vreme putovanja:\n");
26     for (i = 1; i <= n; i++)
        scanf("%d", &niz[i]);
28

```

### 3 Predstavljanje podataka

---

```
30  /* Ucitavaju se redni brojevi autobusa cije se vreme putovanja
    menja. */
32  printf("Unesite redne brojeve autobusa koji"
    " putuju na potezu Pozega, Uzice:\n");
    scanf("%d%d", &k, &t);

34  /* Vrsi se provera ispravnosti ulaza. */
36  if (k <= 0 || k > n || t <= 0 || t > n) {
    printf("Redni brojevi autobusa nisu u dozvoljenom opsegu.\n");
38    exit(EXIT_FAILURE);
    }

40  /* Ucitava se vrednost kasnjenja. */
42  printf("Unesite novo vreme:\n");
    scanf("%d", &kasnjenje);

44  /* Vrsi se provera ispravnosti ulaza. */
46  if (kasnjenje < 0) {
    printf("Greska: neispravan unos.\n");
48    exit(EXIT_FAILURE);
    }

50  /* Azuriraju se vremena putovanja. */
52  for (i = k; i <= t; i++)
    niz[i] += kasnjenje;

54  /* Ispis rezultata. */
56  printf("Vreme putovanja nakon izmena:");
    for (i = 1; i <= n; i++)
58      printf("%d ", niz[i]);
    printf("\n");

60    exit(EXIT_SUCCESS);
62 }
```

#### Rešenje 3.1.11

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

3  #define MAKS 100

5  /* Funkcija koja racuna zbir elemenata niza od pozicije i do
    pozicije j. */
7  int zbir(int a[], int i, int j)
9  {
    int k;
11    int rezultat = 0;

13    /* Obilaze se elementi niza iz zadatog opsega. */
    for (k = i; k <= j; k++) {
```



```

15     rezultat += a[k];
16 }
17
18 /* Vraca se izracunata vrednost. */
19 return rezultat;
20 }
21
22 int main()
23 {
24     /* Deklaracije potrebnih promenljivih. */
25     int n, i, j;
26     int a[MAKS];
27
28     /* Ucitava se dimenzija niza i vrsi se provera ispravnosti
29        ulaza. */
30     printf("Unesite broj elemenata niza: ");
31     scanf("%d", &n);
32     if (n <= 0 || n > MAKS) {
33         printf("Greska: neispravan unos.\n");
34         exit(EXIT_FAILURE);
35     }
36
37     /* Ucitavaju se elementi niza. */
38     printf("Unesite elemente niza:");
39     for (i = 0; i < n; i++)
40         scanf("%d", &a[i]);
41
42     /* Ucitavaju se vrednosti granica i vrsi se provera
43        ispravnosti ulaza. */
44     printf("Unesite vrednosti za i i j: ");
45     scanf("%d%d", &i, &j);
46     if (i < 0 || j < 0 || i > n - 1 || j > n - 1 || i > j) {
47         printf("Greska: neispravan unos.\n");
48         exit(EXIT_FAILURE);
49     }
50
51     /* Ispis rezultata. */
52     printf("Zbir je: %d", zbir(a, i, j));
53
54     exit(EXIT_SUCCESS);
55 }

```

### Rešenje 3.1.12

```

1  #include <stdio.h>
2  #include <stdlib.h>
3
4  #define MAKS 100
5
6  int main()
7  {

```

### 3 Predstavljanje podataka

```
9  /* Deklaracija potrebnih promenljivih. */
   float brojevi[MAKS];
   int n, i;

11

13  /* Ucitava se dimenzija niza i vrsi se provera ispravnosti
   ulaza. */
   printf("Unesite broj elemenata niza: ");
15   scanf("%d", &n);
   if (n < 1 || n > MAKS) {
17       printf("Greska: neispravan unos.\n");
       exit(EXIT_FAILURE);
19   }

21  /* Ucitavaju se elementi niza. */
   printf("Unesite elemente niza:\n");
23   for (i = 0; i < n; i++) {
       scanf("%f", &brojevi[i]);
25   }

27  /* Ukoliko je i-ti element niza brojevi[i] negativan broj,
   kvadrira se tako sto se pomnozi sa samim sobom. */
29   for (i = 0; i < n; i++) {
       if (brojevi[i] < 0)
31         brojevi[i] *= brojevi[i];
   }

33

35  /* Ispisuje se novodobijeni niz. */
   for (i = 0; i < n; i++) {
       printf("%g ", brojevi[i]);
37   }
   printf("\n");

39   exit(EXIT_SUCCESS);
41 }
```

#### Rešenje 3.1.13

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

3

5  #define MAKS 100

7  /* Funkcija koja kvadrira elemente niza koji se nalaze na parnim
   pozicijama. */
   void kvadriranje(float a[], int n)
9   {
       int i;
11      /* Obilaze se elementi na parnim pozicijama i kvadriraju se:
       a[i] postaje a[i]*a[i]. */
13      for (i = 0; i < n; i += 2)
          a[i] *= a[i];
```

```

15 }
17 int main()
18 {
19     /* Deklaracija potrebnih promenljivih. */
20     int n, i;
21     float a[MAKS];
22
23     /* Ucitava se dimenzija niza i vrsi se provera ispravnosti
24        ulaza. */
25     printf("Unesite broj elemenata niza: ");
26     scanf("%d", &n);
27     if (n <= 0 || n > MAKS) {
28         printf("Greska: neispravan unos.\n");
29         exit(EXIT_FAILURE);
30     }
31
32     /* Ucitavaju se elementi niza. */
33     printf("Unesite elemente niza:");
34     for (i = 0; i < n; i++)
35         scanf("%f", &a[i]);
36
37     /* Poziva se funkcija. */
38     kvadriranje(a, n);
39
40     /* Ispisuju se elementi novodobijenog niza. */
41     for (i = 0; i < n; i++)
42         printf("%g ", a[i]);
43     printf("\n");
44
45     exit(EXIT_SUCCESS);
46 }

```

### Rešenje 3.1.14

```

1 #include <stdio.h>
2 #include <stdlib.h>
3
4 #define MAKS 100
5
6 /* Funkcija racuna zbir prvih k pozitivnih elemenata niza. */
7 float zbir_pozitivnih(float a[], int n, int k)
8 {
9     int i;
10    float zbir = 0;
11
12    /* Obilazi se element po element niza. Postupak se zavrшава
13       ukoliko se dodje do kraja niza ili ukoliko se sabere k
14       pozitivnih elemenata. */
15    for (i = 0; i < n && k > 0; i++) {
16        if (a[i] >= 0) {

```

```

    zbir += a[i];
18    /* Umanjuje se brojac pozitivnih elemenata. */
    k--;
20    }
    }
22    return zbir;
24 }

26 int main()
{
28    /* Deklaracija potrebnih promenljivih. */
    int n, i, k;
30    float a[MAKS];

32    /* Ucitava se dimenzija niza i vrsi se provera ispravnosti
        ulaza. */
34    printf("Unesite broj elemenata niza: ");
    scanf("%d", &n);
36    if (n <= 0 || n > MAKS) {
        printf("Greska: neispravan unos.\n");
38        exit(EXIT_FAILURE);
    }
40

    /* Ucitavaju se elementi niza. */
42    printf("Unesite elemente niza: ");
    for (i = 0; i < n; i++)
44        scanf("%f", &a[i]);

46    /* Ucitava se broj k i vrsi se provera ispravnosti ulaza. */
    printf("Unesite vrednost za k: ");
48    scanf("%d", &k);
    if (k < 0 || k > n) {
50        printf("Greska: neispravan unos.\n");
        exit(EXIT_FAILURE);
52    }

54    /* Ispis rezultata. */
    printf("Zbir je: %.2f\n", zbir_pozitivnih(a, n, k));
56    exit(EXIT_SUCCESS);
58 }
```

#### Rešenje 3.1.15

```

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

5  #define MAKS 200
```

```
7  /* Funkcija koja racuna aritmeticku sredinu onih elemenata niza
   koji su deljivi sa 3. */
9  float artmeticka_sredina_3(int niz[], int n)
{
11     int i, suma_deljivih_sa_tri = 0, broj_deljivih_sa_tri = 0;

13     /* U petlji se proverava svaki element niza i za one koji su
       deljivi sa 3 uvecava se suma i odgovarajuci brojac. */
15     for (i = 0; i < n; i++){
17         if (niz[i] % 3 == 0) {
19             suma_deljivih_sa_tri += niz[i];
21             broj_deljivih_sa_tri++;
23         }
25     }

27     return (float) suma_deljivih_sa_tri / broj_deljivih_sa_tri;
29 }

31 /* Funkcija koja racuna element koji je najblizi po vrednosti
   aritmetickoj sredini elemenata deljivih sa tri. */
33 int blizu_3(int a[], int n)
{
35     float aritmeticka_sredina = artmeticka_sredina_3(a, n);
37     int najblizi_po_vrednosti = a[0];

39     /* Radi brzine izracunavanja pamti se razdaljina izmedju trenutno
       najblizeg elementa i aritmeticke sredine. Izracunava se
       apsolutna vrednost razdaljine jer ona moze biti i negativna. */
41     float razdaljina_najblizeg = fabs(aritmeticka_sredina -
43     najblizi_po_vrednosti);
45     float nova_razdaljina;
47     int i;

49     /* U petlji se ispituje svaki element niza i proverava se da li
       postoji neki koji je po vrednosti blizi aritmetickoj sredini.
       Ako se naidje na takav element, azurira se vrednost najblizeg
       po vrednosti, kao i vrednost njegove razdaljine. */
51     for (i = 1; i < n; i++) {
53         nova_razdaljina = fabs(a[i] - aritmeticka_sredina);
55         if (nova_razdaljina < razdaljina_najblizeg) {
57             razdaljina_najblizeg = nova_razdaljina;
59             najblizi_po_vrednosti = a[i];
61         }
63     }

65     return najblizi_po_vrednosti;
67 }

69 int main()
{
71     /* Deklaracija potrebnih promenljivih. */
```

### 3 Predstavljanje podataka

```
int niz[MAKS], n, i;

59
/* Ucitava se dimenzija niza. */
61 printf("Unesite broj elemenata niza: ");
scanf("%d", &n);

63
/* Vrsi se provera ispravnosti ulaza. */
65 if (n <= 0 || n > MAKS) {
    printf("Greska: neispravan unos.\n");
67     exit(EXIT_FAILURE);
}

69
/* Ucitavaju se elementi niza. */
71 printf("Unesite elemente niza: ");
for (i = 0; i < n; i++)
73     scanf("%d", &niz[i]);

75
/* Ispis rezultata. */
printf("Trazeni broj je %d.\n", blizu_3(niz, n));
77
exit(EXIT_SUCCESS);
79 }
```

#### Rešenje 3.1.16

```
1 #include<stdio.h>
  #include<stdlib.h>
3
  #define BROJ_CIFARA 10
5
int main()
7 {
    /* Deklaracije potrebnih promenljivih. */
9     int x, x_original, cifra, i;
    int brojac[BROJ_CIFARA];

11
    /* Ucitava se ceo broj sa standardnog ulaza. */
13     printf("Unesite ceo broj:\n");
    scanf("%d", &x);

15
    /* Cuva se njegova x_originalna vrednost zbog finalnog ispisa. */
17     x_original = x;
    x = abs(x);

19
    /* Svaki element niza brojac predstavlja brojcu za jednu od
21     cifara: brojac[0] predstavlja broj nula u zapisu broja x
    brojac[1] predstavlja broj jedinica u zapisu broja x ...
23     brojac[9] predstavlja broj devetki u zapisu broja x. */

25
    /* Brojac se na pocetku inicijalizuju nulama. */
    for (i = 0; i < BROJ_CIFARA; i++) {
```

```

27     brojaci[i] = 0;
28 }
29
30 /* Sve dok ima cifara u zapisu broja x */
31 do {
32     /* Izdvaja se krajnja desna cifara. */
33     cifra = x % 10;
34
35     /* Uvecava se njen broj pojavljivanja. */
36     brojaci[cifra]++;
37
38     /* Prelazi se na analiziranje sledece cifre. */
39     x /= 10;
40 } while (x);
41
42 /* Ispisuju se informacije o ciframa koje se nalaze u zapisu
43    broja x. */
44 for (i = 0; i < BROJ_CIFARA; i++) {
45     if (brojaci[i]) {
46         printf("U zapisu broja %d, cifra %d se pojaviljuje %d puta\n",
47             x_original, i, brojaci[i]);
48     }
49 }
50
51 exit(EXIT_SUCCESS);
52 }

```

### Rešenje 3.1.17

```

1  #include <stdio.h>
2  #include <stdlib.h>
3  #include <ctype.h>
4
5  #define MAKS 100
6
7  /* Funkcija prebrojava cifre u datom nizu karaktera. */
8  int cifre(char a[], int n)
9  {
10     int i, broj_cifara = 0;
11
12     for (i = 0; i < n; i++) {
13         if (isdigit(a[i]))
14             broj_cifara++;
15     }
16
17     return broj_cifara;
18 }
19
20 int main()
21 {
22     /* Deklaracija potrebnih promenljivih. */

```

### 3 Predstavljanje podataka

---

```
23  char a[MAKS];
    int n, i;

25

    /* Ucitava se broj elemenata niza i vrsi se provera ispravnosti
    ulaza. */
27  printf("Unesite broj elemenata niza:");
    scanf("%d", &n);
29  if (n <= 0 || n > MAKS) {
    printf("Greska: neispravan unos.\n");
    exit(EXIT_FAILURE);
31  }
33

    /* Ucitavaju se elementi niza. */
35  printf("Unesite elemente niza:");
    for (i = 0; i < n; i++) {
37      /* Preskace se prethodno uneti znak za novi red. */
        getchar();

        /* Ucitava se karakter i smesta se u niz. */
        scanf("%c", &a[i]);
43    }

    /* Ispis rezultata. */
45  printf("Broj cifara je: %d\n", cifre(a, n));

    exit(EXIT_SUCCESS);
47
49 }
```

#### Rešenje 3.1.18

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

4 #define BROJ_CIFARA 10
  #define DUZINA_ALFABETA 26

6
    /* Funkcija za ispis elemenata niza. */
8 void ispis(int niz[], int n, char c)
    {
10     int i;
        for (i = 0; i < n; i++) {
12         if (niz[i] != 0)
            printf("Karakter %c se pojavljuje %d puta\n", c + i, niz[i]);
14     }
    }

16
    int main()
18 {
    /* Niz u kojem ce se cuvati informacije o pojavljivanju cifara. */
20     int cifre[BROJ_CIFARA];
```



```

22  /* Niz u kojem ce se cuvati informacije o pojavljivanju malih
    slova. */
24  int mala_slova[DUZINA_ALFABETA];

26  /* Niz u kojem ce se cuvati informacije o pojavljivanju velikih
    slova. */
28  int velika_slova[DUZINA_ALFABETA];

30  int c, i;

32  /* Brojaci se na pocetku inicijalizuju nulama. */
33  for (i = 0; i < BROJ_CIFARA; i++) {
34      cifre[i] = 0;
35  }
36  for (i = 0; i < DUZINA_ALFABETA; i++) {
37      mala_slova[i] = 0;
38      velika_slova[i] = 0;
39  }

40  /* Ucitavaju se karakteri sve do kraja ulaza. */
41  while ((c = getchar()) != EOF) {
42      /* Ako je procitani karakter veliko slovo uvecava se broj
43      pojavljivanja odgovarajuceg velikog slova. Indeks velikog
44      slova u nizu se odredjuje oduzimanjem slova A.
45      Na taj nacin slovo 'A' ce imati indeks 0, slovo 'B' indeks
46      1, itd.*/
47      if (c >= 'A' && c <= 'Z') {
48          velika_slova[c - 'A']++;
49      }
50      else if (c >= 'a' && c <= 'z') {
51          /* Ako je procitani karakter malo slovo uvecava se broj
52          pojavljivanja odgovarajuceg malog slova. */
53          mala_slova[c - 'a']++;
54      }
55      else if (c >= '0' && c <= '9'){
56          /* Ako je procitani karakter cifra uvecava se broj
57          pojavljivanja odgovarajuce cifre. */
58          cifre[c - '0']++;
59      }
60  }

62  /* Ispisuju se trazene informacije. */
63  ispis(cifre, BROJ_CIFARA, '0');
64  ispis(mala_slova, DUZINA_ALFABETA, 'a');
65  ispis(velika_slova, DUZINA_ALFABETA, 'A');

66  exit(EXIT_SUCCESS);
67  }

```

## Rešenje 3.1.19

```
1 #include <stdio.h>
2 #include <stdlib.h>
3 #include <ctype.h>
4
5 #define DUZINA_ALFABETA 26
6
7 /* Funkcija za ispis elemenata niza. */
8 void ispis(int niz[], int n)
9 {
10     int i;
11     for (i = 0; i < n; i++)
12         printf("%c:%d ", 'a' + i, niz[i]);
13     putchar('\n');
14 }
15
16 int main()
17 {
18     /* Deklaracija potrebnih promenljivih. */
19     int mala_slova[DUZINA_ALFABETA];
20     int c, i;
21
22     /* Inicijalizacija niza brojac na nule. */
23     for (i = 0; i < DUZINA_ALFABETA; i++)
24         mala_slova[i] = 0;
25
26     /* Ucitavaju se karakteri sve do kraja ulaza. */
27     while ((c = getchar()) != EOF) {
28         /* Ako je procitani karakter slovo broj pojavljivanja slova se
29            uvecava. Kako se zanemaruje velicina slova, svako slovo se
30            pretvori u malo i potom se element na odgovarajucoj poziciji
31            u nizu uveca. */
32         if (isalpha(c))
33             mala_slova[tolower(c) - 'a']++;
34     }
35
36     /* Ispis rezultata. */
37     ispis(mala_slova, DUZINA_ALFABETA);
38
39     exit(EXIT_SUCCESS);
40 }
```

#### Rešenje 3.1.20

```
1 #include <stdio.h>
2 #include <stdlib.h>
3
4 #define MAKS 100
5
6 int main()
7 {
```

```

9      /* Deklaracije potrebnih promenljivih. */
10     char karakteri[MAKS];
11     char c;
12     int i, n;
13
14     /* Ucitava se karakter po karakter sa standardnog ulaza sve dok
15        se ne unese * ili se ne prekorači maksimalni broj karaktera. */
16     for (i = 0; i < MAKS; i++) {
17
18         printf("Unesite karakter: ");
19         scanf("%c", &c);
20
21         /* Cita se znak za novi red nakon unesenog karaktera. */
22         getchar();
23
24         /* Ukoliko je unet karakter * izlazi se iz petlje. */
25         if (c == '*')
26             break;
27
28         /* Procitani karakter se smesta u niz. */
29         karakteri[i] = c;
30     }
31
32     /* Broj unetih karaktera nakon izlaska iz petlje je i-1. */
33     n = i - 1;
34
35     /* Ispis karaktera u obrnutom redosledu. */
36     for (i = n; i >= 0; i--) {
37         printf("%c ", karakteri[i]);
38     }
39     printf("\n");
40
41     exit(EXIT_SUCCESS);
42 }

```

### Rešenje 3.1.21

```

1 #include <stdio.h>
2 #include <stdlib.h>
3 #include <ctype.h>
4
5 #define MAKS 200
6
7 int main()
8 {
9     /* Deklaracije potrebnih promenljivih. */
10    int n, i;
11    char niz[MAKS], blanko;
12

```

```
14  /* Ucitava se dimenzija niza i vrsi se provera ispravnosti
    ulaza. */
16  printf("Unesite trenutni broj klijenata: ");
18  scanf("%d", &n);
    if (n <= 0 || n > MAKS) {
18      printf("Greska: neispravan unos.\n");
        exit(EXIT_FAILURE);
20  }

22  /* Nakon unosa dimenzije, korisnik unosi blanko ili znak za novi
    red. Ovaj karakter je potrebno ucitati pre unosa elemenata
24      niza, inace ce ovaj karakter biti prvi element niza, sto nije
    zeljeno ponasanje. */
26  scanf("%c", &blanko);

28  printf("Unesite elemete niza: ");
    for (i = 0; i < n; i++)
30      scanf("%c", &niz[i]);

32  /* Provera da li je niz palindrom. Uporedjuju se prvi i
    poslednji, drugi i preposlednji... odnosno i-ti i
34      ((n-1)-i)-ti element niza. Ako nisu jednaki, niz nije
    palindrom. Kako se istovremeno posmatraju dva elementa niza
36      dovoljno je da brojac u petlji ide do polovine dimenzije niza.
    */
    for (i = 0; i < n / 2; i++){
38        /* Kako se zanemaruje velicina slova, svaki element niza se
            pretvara u malo slovo i potom se vrsi poredjenje. */
40        if (tolower(niz[i]) != tolower(niz[n - 1 - i])) {
            printf("Niz nije palindrom.\n");
42            /* Ukoliko niz nije palindrom, ispitivanje se moze prekinuti
                i izaci iz programa, dalje ispitivanje nije potrebno. */
44            exit(EXIT_SUCCESS);
        }
46    }

48  /* U slucaju kada je petlja zavrшена, a nije izaslo iz programa,
    niz jeste palindrom jer uslov nikada nije bio ispunjen. */
50  printf("Niz jeste palindrom.\n");

52  exit(EXIT_SUCCESS);
}
```

#### Rešenje 3.1.22

```
1  #include <stdio.h>
    #include <stdlib.h>
3
    #define MAKS 100
5
    int main()
```

```
7 {
9     /* Deklaracija potrebnih promenljivih. */
10    int brojevi[MAKS];
11    int n, i;
12    int pozicija_najveceg, pozicija_najmanjeg, najveci, najmanji;
13
14    /* Ucitava se dimenzija niza i vrsi se provera ispravnosti
15       ulaza. */
16    printf("Unesite dimenziju niza: ");
17    scanf("%d", &n);
18    if (n < 1 || n > MAKS) {
19        printf("Greska: neispravan unos.\n");
20        exit(EXIT_FAILURE);
21    }
22
23    /* Ucitavaju se elementi niza. */
24    printf("Unesite elemente niza:\n");
25    for (i = 0; i < n; i++)
26        scanf("%d", &brojevi[i]);
27
28    /* Najvecim, kao i najmanjim elementom niza proglašava se nulti
29       element niza. Pozicije najveceg i najmanjeg elementa se
30       postavljaju na 0. */
31    najveci = brojevi[0];
32    najmanji = brojevi[0];
33    pozicija_najveceg = 0;
34    pozicija_najmanjeg = 0;
35
36    /* U prolazu kroz niz trazi se najveci i najmanji element i
37       pamte se njihove pozicije. */
38    for (i = 1; i < n; i++) {
39        if (brojevi[i] > najveci) {
40            najveci = brojevi[i];
41            pozicija_najveceg = i;
42        }
43
44        if (brojevi[i] < najmanji) {
45            najmanji = brojevi[i];
46            pozicija_najmanjeg = i;
47        }
48    }
49
50    /* Zamenjuju se elementi na pozicijama pozicija_najmanjeg i
51       pozicija_najveceg. */
52    brojevi[pozicija_najveceg] = najmanji;
53    brojevi[pozicija_najmanjeg] = najveci;
54
55    /* Ispisuje se rezultujući niz. */
56    for (i = 0; i < n; i++)
57        printf("%d ", brojevi[i]);
58    printf("\n");
```

```
    exit(EXIT_SUCCESS);
59 }
```

#### Rešenje 3.1.23

```
#include <stdio.h>
2 #include <stdlib.h>
#include <stdlib.h>
4
#define MAKS 500
6
/* Funkcija koja ucitava elemente niza i vrši proveru ispravnosti
8 ulaza. Kao povratnu vrednost vraca dimenziju ucitanog niza. */
int ucitavanje(int niz[])
10 {
    int i, n;
12
    printf("Unesite dimenziju niza: ");
14    scanf("%d", &n);
    if (n <= 0 || n > MAKS) {
16        printf("Nedozvoljena dimenzija niza.\n");
        exit(EXIT_FAILURE);
18    }

    printf("Unesite elemente niza: ");
20    for (i = 0; i < n; i++)
        scanf("%d", &niz[i]);
22

    return n;
24 }

/* Funkcija koja vraca 1 ukoliko broj x postoji u nizu, 0 inace. */
26 int postoji(int niz[], int n, int x)
{
30     int i;

    for (i = 0; i < n; i++)
32         if (niz[i] == x)
            return 1;
34

    return 0;
36 }

38 int main()
{
40     /* Deklaracija potrebnih promenljivih. */
    int a[MAKS], b[MAKS], unija[2 * MAKS], presek[MAKS],
        razlika[MAKS];
42
    int i, n_a, n_b, n_unija, n_presek, n_razlika;
44

    /* Unose se dva niza. */
46
```

```
48     n_a = učitavanje(a);
    n_b = učitavanje(b);

50     /* Brojaci elemenata u nizovima unija, presek i razlika. */
    n_unija = 0;
52     n_presek = 0;
    n_razlika = 0;

54     for (i = 0; i < n_a; i++) {
56         /* Ukoliko se element a[i] ne nalazi u uniji, dodaje se u uniju
           i povećava se brojac elemenata u nizu unija. */
58         if (postoji(unija, n_unija, a[i]) == 0) {
            unija[n_unija] = a[i];
60             n_unija++;
        }

62         /* Ukoliko se element a[i] nalazi u nizu b i ne postoji u nizu
           presek, dodaje se presek i povećava se brojac elemenata u
           nizu presek. */
64         if (postoji(b, n_b, a[i]) == 1
66             && postoji(presek, n_presek, a[i]) == 0) {
            presek[n_presek] = a[i];
68             n_presek++;
        }

70         /* Ukoliko element a[i] ne postoji u nizu b i ne postoji u nizu
           razlika, dodaje se u razliku i povećava se brojac elemenata
           u nizu razlika. */
72         if (postoji(b, n_b, a[i]) == 0
74             && postoji(razlika, n_razlika, a[i]) == 0) {
            razlika[n_razlika] = a[i];
76             n_razlika++;
        }

80     }

82     /* Elemente niza b koji nisu uneti u uniju dodaju se u uniju. */
    for (i = 0; i < n_b; i++)
84         if (postoji(unija, n_unija, b[i])) {
            unija[n_unija] = b[i];
86             n_unija++;
        }

88     /* Ispis rezultata. */
90     printf("Unija: ");
    for (i = 0; i < n_unija; i++)
92         printf("%d ", unija[i]);
    printf("\nPresek: ");
94     for (i = 0; i < n_presek; i++)
        printf("%d ", presek[i]);
    printf("\nRazlika: ");
96     for (i = 0; i < n_razlika; i++)
98         printf("%d ", razlika[i]);
```

```
100     exit(EXIT_SUCCESS);  
    }
```

#### Rešenje 3.1.24

```
#include <stdio.h>  
2 #include <stdlib.h>  
  
4 #define MAKS 100  
  
6 int main()  
{  
8     /* Deklaracija potrebnih promenljivih. */  
    int a[MAKS], b[MAKS], c[2 * MAKS];  
10    int n, i, j;  
  
12    /* Ucitava se dimenzija niza i vrsi se provera ispravnosti  
        ulaza. */  
14    printf("Unesite dimenziju nizova:\n");  
    scanf("%d", &n);  
16    if (n < 1 || n > MAKS) {  
        printf("Greska: neispravan unos.\n");  
18        exit(EXIT_FAILURE);  
    }  
  
20    /* Ucitavaju se elementi prvog niza. */  
22    printf("Unesite elemente niza a:\n");  
    for (i = 0; i < n; i++) {  
24        scanf("%d", &a[i]);  
    }  
  
26    /* Ucitavaju se elementi drugog niza. */  
28    printf("Unesite elemente niza b:\n");  
    for (i = 0; i < n; i++) {  
30        scanf("%d", &b[i]);  
    }  
  
32    /* Formira se treci niz. Koriste se dva indeksa:  
34    - indeks i pomocu kojeg se pristupa elementima nizova a i b i  
        koji treba uvecati za 1 nakon svake iteracije  
36    - indeks j pomocu kojeg se pristupa elementima rezultujuceg  
        niza c; s obzirom da se u svakoj iteraciji u niz c smestaju  
38    dva elementa, jedan iz niza a i jedan iz niza b, indeks j se  
        uvecava za 2 nakon svake iteracije. */  
40    for (i = 0, j = 0; i < n; i++, j += 2) {  
        c[j] = a[i];  
42        c[j + 1] = b[i];  
    }  
  
44    /* Ispisuju se elementi rezultujuceg niza. */
```



```

46 printf("Rezultujuci niz:\n");
   for (i = 0; i < 2 * n; i++)
48     printf("%d ", c[i]);
   printf("\n");

50 exit(EXIT_SUCCESS);
52 }

```

### Rešenje 3.1.25

```

1  #include <stdio.h>
   #include <stdlib.h>
3
   #define MAKS 100
5
   int main()
7 {
   /* Deklaracija potrebnih promenljivih. */
9   int a[MAKS], b[MAKS], c[2 * MAKS];
   int i, n;

11
   /* Ucitava se dimenzija niza i vrsi se provera ispravnosti
13    ulaza. */
   printf("Unesite broj n: ");
15   scanf("%d", &n);
   if (n < 1 || n > MAKS) {
17     printf("Greska: neispravan unos.\n");
     exit(EXIT_FAILURE);
19   }

21
   /* Ucitavaju se elementi nizova */
   printf("Unesite elemente niza a: ");
23   for (i = 0; i < n; i++)
     scanf("%d", &a[i]);

25
   printf("Unesite elemente niza b: ");
27   for (i = 0; i < n; i++)
     scanf("%d", &b[i]);

29
   /* Niz c ima 2*n elemenata: prvih n elemenata su elementi niza b,
31    a narednih n elemenata elementi niza a. Elementi niza b se
     nalaze na pozicijama 0,1,2,...n-1, a elementi niza a na
33    pozicijama n,n+1,...2*n-1. Jednim prolaskom kroz petlju na
     poziciju i u nizu c se postavlja element b[i] niza b, a na
35    poziciju n+i element a[i] niza a. */
   for (i = 0; i < n; i++) {
37     c[i] = b[i];
     c[n + i] = a[i];
39   }

41
   /* Ispisuju se elementi niza c. */

```

### 3 Predstavljanje podataka

---

```
    for (i = 0; i < 2 * n; i++)
43     printf("%d ", c[i]);
    printf("\n");
45
    exit(EXIT_SUCCESS);
47 }
```

#### Rešenje 3.1.26

```
1  #include <stdio.h>
   #include <stdlib.h>
3
   #define MAKS 100
5
   /* Funkcija koja ucitava elemente niza. */
7  void unos(int niz[], int n)
   {
9     int i;
    printf("Uneti elemente sortiranog niza:\n");
11    for (i = 0; i < n; i++)
        scanf("%d", &niz[i]);
13 }

15 /* Funkcija za ispis niza. */
   void ispis(int niz[], int n)
17 {
    int i;
19    for (i = 0; i < n; i++)
        printf("%d ", niz[i]);
21    printf("\n");
   }
23
   int main()
25 {
    /* Deklaracija potrebnih promenljivih. */
27    int a[MAKS], b[MAKS], c[2 * MAKS];
    int n;
29    /* Brojac u petlji za elemente niza a. */
    int i = 0;
31    /* Brojac u petlji za elemente niza b. */
    int j = 0;
33    /* Brojac u petlji za elemente niza c. */
    int k = 0;
35
    /* Ucitava se dimenzija niza i vrsi se provera ispravnosti
37     ulaza. */
    printf("Uneti broj elemenata niza: ");
39    scanf("%d", &n);
    if (n <= 0 || n > MAKS) {
41        printf("Greska: neispravan unos.\n");
        exit(EXIT_FAILURE);
    }
```

```

43     }

44     /* Ucitavaju se elementi nizova. */
45     unos(a, n);
46     unos(b, n);

47     /* Vrsi se spajanje nizova. */
48     while (i < n && j < n) {
49         /* Porede se elementi nizova a i b i u niz c upisuje se samo
50            onaj koji je manji. Ako je upisan element iz niza a, onda se
51            vrsi i uvecavanje brojaca i (prelazak na sledeci element
52            niza a), a ako je upisan element iz niza b, onda se vrsi
53            uvecavanje brojaca j (prelazak na sledeci element niza b). */
54         if (a[i] < b[j]) {
55             c[k] = a[i];
56             i++;
57         } else {
58             c[k] = b[j];
59             j++;
60         }
61         /* U nizu c na poziciju k je upisan ili a[i] ili b[j]. Brojac k
62            se uvecava. */
63         k++;
64     }

65     /* Ukoliko je ostalo elemenata u nizu a, upisuju se u niz c. */
66     while (i < n) {
67         c[k] = a[i];
68         k++;
69         i++;
70     }

71     /* Ukoliko je ostalo elemenata u nizu b, upisuju se u niz c. */
72     while (j < n) {
73         c[k] = b[j];
74         k++;
75         j++;
76     }

77     /* Ispis elemenata niza c cija dimenzija je zbir dimenzija nizova
78        a i b. */
79     ispis(c, 2 * n);

80     exit(EXIT_SUCCESS);
81 }

```

### Rešenje 3.1.27

```

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

```

```
3  #define MAKS 10
5
6  /* Funkcija koja učitava elemente niza. */
7  void unos(int niz[])
8  {
9      int i;
10     printf("Unesite %d brojeva:\n", MAKS);
11     for (i = 0; i < MAKS; i++)
12         scanf("%d", &niz[i]);
13 }
14
15 /* Funkcija koja ispisuje elemente niza. */
16 void ispis(int niz[], int n)
17 {
18     int i;
19     for (i = 0; i < n; i++)
20         printf("%d ", niz[i]);
21     printf("\n");
22 }
23
24 int main()
25 {
26     /* Deklaracija potrebnih promenljivih. */
27     int niz[MAKS];
28     int i = 0, j = MAKS - 1;
29     int pom;
30
31     /* Ucitavanje niza. */
32     unos(niz);
33
34     /* Kreće se sa početka niza (po brojacu i) i sa kraja niza (po
35     brojacu j) i svaki put kada se naidje na elemente koji po
36     parnosti ne odgovaraju delu niza u kome treba da budu, ti
37     elementi se zamene. */
38     while (i < j && i < MAKS && j >= 0) {
39         if (niz[i] % 2 != 0 && niz[j] % 2 == 0) {
40             pom = niz[i];
41             niz[i] = niz[j];
42             niz[j] = pom;
43         }
44
45         /* Ukoliko je element na poziciji i paran, prelazi se na
46         sledeci element niza, brojac i se uvecava. */
47         if (niz[i] % 2 == 0)
48             i++;
49
50         /* Ukoliko je element na poziciji j neparan, prelazi se na
51         sledeci element niza, brojac j se smanjuje. */
52         if (niz[j] % 2 != 0)
53             j--;
54     }
```

```

55      /* Ispis rezultata. */
57      printf("Rezultujući niz:\n");
        ispis(niz, MAKS);
59
        exit(EXIT_SUCCESS);
61    }

```

### Rešenje 3.1.28

```

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

#define MAKS 100

/* Funkcija koja učitava elemente niza. */
void ucitaj(int a[], int n)
{
    int i;
    for (i = 0; i < n; i++)
        scanf("%d", &a[i]);
}

/* Funkcija koja ispisuje elemente niza. */
void stampaaj(int a[], int n)
{
    int i;
    for (i = 0; i < n; i++)
        printf("%d ", a[i]);
    printf("\n");
}

/* Funkcija koja obrće elemente niza. */
void obrni(int a[], int n)
{
    int t, i, j;

    /* Za niz a[0], a[1], ..., a[n-2], a[n-1] obrnuti niz je a[n-1],
       a[n-2], ..., a[1], a[0]. Zato je potrebno razmeniti vrednosti
       elemenata a[0] i a[n-1], a[1] i a[n-2], itd. i zaustaviti se
       kada je vrednost indeksa prvog elementa veća od vrednosti
       drugog elementa. */
    for (i = 0, j = n - 1; i < j; i++, j--) {
        t = a[i];
        a[i] = a[j];
        a[j] = t;
    }
}

/* Funkcija koja rotira niz ciklicno za jedno mesto u levo. */
void rotiraj1(int a[], int n)

```

### 3 Predstavljanje podataka

---

```
42 {
43     int i, tmp;
44
45     /* Izdvaja se prvi element niza. */
46     tmp = a[0];
47
48     /* Pomeraju se preostali elementi niza za jedno mesto u levo. */
49     for (i = 0; i < n - 1; i++) {
50         a[i] = a[i + 1];
51     }
52
53     /* Poslednjem elementu se dodeljuje sacuvana vrednost prvog
54        elementa. */
55     a[n - 1] = tmp;
56 }
57
58 /* Funkcija koja rotira niz ciklicno za k mesta u levo. */
59 void rotirajk(int a[], int n, int k)
60 {
61     int i;
62
63     /* Odredjuje se vrednost broja k koja je u opsegu od 0 do n-1
64        kako bi se izbegla suvisna pomeranja. */
65     k = k % n;
66
67     /* Niz se rotira za jednu poziciju ulevo k puta. */
68     for (i = 0; i < k; i++)
69         rotiraj1(a, n);
70 }
71
72 int main()
73 {
74     /* Deklaracija potrebnih promenljivih. */
75     int a[MAKS];
76     int n, k;
77
78     /* Ucitava se dimenzija niza i vrsi se provera ispravnosti
79        ulaza. */
80     printf("Unesite dimenziju niza:");
81     scanf("%d", &n);
82     if (n < 1 || n > MAKS) {
83         printf("Greska: neispravan unos.\n");
84         exit(EXIT_FAILURE);
85     }
86
87     /* Ucitavaju se elementi niza. */
88     ucitaj(a, n);
89
90     /* Testira se rad napisanih funkcija */
91
92     /* Obrtanje niza. */
93     printf("Elementi niza nakon obrtanja:\n");
```

```

94 obrni(a, n);
   stampaj(a, n);

96
   /* Rotiranje za jedno mesto u levo. */
98   printf("Elementi niza nakon rotiranja za 1 mesto ulevo:\n");
   rotiraj1(a, n);
100   stampaj(a, n);

   /* Rotiranje za k mesta u levo. */
102   printf("Unesite jedan pozitivan ceo broj:");
104   scanf("%d", &k);
   if (k <= 0) {
106       printf("Greska: neispravan unos.\n");
       exit(EXIT_FAILURE);
108   }
   rotirajk(a, n, k);
110   printf("Elementi niza nakon rotiranja za %d mesto ulevo:\n", k);
   stampaj(a, n);
112
   exit(EXIT_SUCCESS);
114 }

```

### Rešenje 3.1.29

```

1  #include <stdio.h>
   #include <stdlib.h>
3
   #define MAKS 2000
5
   /* Funkcija za ispis elemenata niza. */
7   void ispis(int niz[], int n)
   {
9       int i;
       for (i = 0; i < n; i++)
11         printf("%d ", niz[i]);
       printf("\n");
13   }

15   /* Funkcija za ubacivanje na kraj niza. Vraca novu dimenziju niza. */
   int ubaci_na_kraj(int niz[], int n, int x)
17   {
       niz[n] = x;
19       return n + 1;
   }

21
   /* Funkcija za ubacivanje na pocetak niza. Vraca novu dimenziju
23    niza. */
   int ubaci_na_pocetak(int niz[], int n, int x)
25   {
       int i;
27

```

### 3 Predstavljanje podataka

---

```
29  /* Prvo se svi elementi niza pomere za jednu poziciju u desno da
    bi se oslobodio prostor za prvi element niza. Poslednji
    element niza se pomera sa pozicije (n-1) na poziciju (n).
31  Slicno se pomeraju i ostali elementi. */
    for (i = n; i > 0; i--)
33      niz[i] = niz[i - 1];

35  /* Na prvu poziciju se upisuje novi element. Bitan je redosled
    naredbi: ako bi prvo bio upisan novi element, a tek onda
    izvršeno pomeranje, element na poziciji niz[0] bi bio obrisani
37  i ne bi mogao biti upisan na poziciju niz[1]. */
    niz[0] = x;
39

41  return n + 1;
    }
43
45  /* Funkcija za ubacivanje elementa na neku poziciju u nizu. Vraca
    novu dimenziju niza. */
    int ubaci_na_poziciju(int niz[], int n, int x, int pozicija)
47  {
        int i;
49
        /* Prvo se svi elementi niza od pozicije do kraja pomere za jedno
51        mesto u desno da bi se oslobodio prostor za novi element niza.
        */
53        for (i = n; i > pozicija; i--)
            niz[i] = niz[i - 1];
55
        /* Na poziciju se upisuje novi element. */
57        niz[pozicija] = x;

59        return n + 1;
    }
61
63  /* Funkcija za brisanje prvog elementa niza. Vraca novu dimenziju
    niza. */
    int brisi_prvog(int niz[], int n)
65  {
        int i;
67
        /* Svi elementi niza pomeraju se za jedno mesto u levo. */
69        for (i = 0; i < n - 1; i++)
            niz[i] = niz[i + 1];
71
        return n - 1;
73    }

75  /* Funkcija za brisanje poslednjeg elementa niza. Vraca novu
    dimenziju niza. */
    int brisi_poslednjeg(int niz[], int n)
77  {
79      /* Dovoljno je smanjiti dimenziju niza, elemente niza nije
```



```

    potrebno brisati. */
81     return n - 1;
    }
83
/* Funkcija za brisanje elementa niza. Pretpostavlja se da element
85     ima samo jedno pojavljivanje (za vezbu napisati funkciju koja
    brise sva pojavljivanja, ako ih ima vise). Vraca novu dimenziju
87     niza. */
int brisi_element(int niz[], int n, int x)
89 {
    int i, j;
91
    /* Prvo treba pronaci poziciju elementa u nizu. */
93     for (i = 0; i < n; i++)
        if (niz[i] == x)
95         break;
97
    /* Provera da li element postoji u nizu. Ako je brojac stigao do
        kraja niza, onda element ne postoji u nizu. */
99     if (i == n) {
        printf("Klijent sa rednim brojem %d ne postoji u nizu.\n", x);
101        return n;
    }
103
    /* Ukoliko element postoji u nizu, svi elementi niza nakon njega
        se pomeraju za jedno mesto u levo. */
105     for (j = i; j < n - 1; j++)
        niz[j] = niz[j + 1];
107
109     return n - 1;
    }
111
int main()
113 {
    int n, niz[MAKS], i, klijent, pozicija;
115
    /* Ucitava se dimenzija niza i vrsi se provera ispravnosti
        ulaza. */
117     printf("Unesite trenutni broj klijenata: ");
    scanf("%d", &n);
119     if (n <= 0 || n > MAKS) {
        printf("Greska: neispravan unos.\n");
121        exit(EXIT_FAILURE);
    }
123
    /* Ucitavaju se elementi niza. */
125     printf("Unesite niz sa rednim brojevima klijenata: ");
    for (i = 0; i < n; i++)
127         scanf("%d", &niz[i]);
129
    /* Ubacivanje klijenta na kraj. */
131     printf("Unesite klijenta kojeg treba ubaciti u niz: ");

```

```
133     scanf("%d", &klijent);
134     n = ubaci_na_kraj(niz, n, klijent);
135     printf("Niz nakon ubacivanja klijenta:\n");
136     ispis(niz, n);

137     /* Ubacivanje klijenta na pocetak. */
138     printf("Unesite prioritnog klijenta kojeg treba "
139           "ubaciti u niz: ");
140     scanf("%d", &klijent);
141     n = ubaci_na_pocetak(niz, n, klijent);
142     printf("Niz nakon ubacivanja klijenta:\n");
143     ispis(niz, n);

144     /* Ubacivanje klijenta na zadatu poziciju. */
145     printf("Unesite prioritnog klijenta kojeg treba ubaciti "
146           "u niz i njegovu poziciju:");
147     scanf("%d%d", &klijent, &pozicija);
148     if (pozicija < 0 || pozicija > n) {
149         printf("Greska: neispravan unos.\n");
150         exit(EXIT_FAILURE);
151     } else {
152         n = ubaci_na_poziciju(niz, n, klijent, pozicija);
153         printf("Niz nakon ubacivanja klijenta:\n");
154         ispis(niz, n);
155     }

156     /* Brisanje prvog klijenta. */
157     n = brisi_prvog(niz, n);
158     printf("Niz nakon odlaska klijenta:\n");
159     ispis(niz, n);

160     /* Brisanje poslednjeg klijenta. */
161     n = brisi_poslednjeg(niz, n);
162     printf("Niz nakon odlaska klijenta:\n");
163     ispis(niz, n);

164     /* Brisanje klijenta sa datim rednim brojem. */
165     printf("Unesite redni broj klijenta koji je napustio red: ");
166     scanf("%d", &klijent);
167     n = brisi_element(niz, n, klijent);
168     printf("Niz nakon odlaska klijenta:\n");
169     ispis(niz, n);

170     exit(EXIT_SUCCESS);
171 }
```

#### Rešenje 3.1.30

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

```
4 #define MAKS 100
6 int main()
{
8     /* Deklaracija potrebnih promenljivih. */
9     int a[MAKS];
10    int i, j, n;

12    /* Ucitava se dimenzija niza i vrši se provera ispravnosti
13       ulaza. */
14    printf("Unesite broj elemenata niza: ");
15    scanf("%d", &n);
16    if (n < 1 || n > MAKS) {
17        printf("Greska: neispravan unos.\n");
18        exit(EXIT_FAILURE);
19    }

20    /* Ucitavaju se elementi niza. */
21    printf("Unesite elemente niza: ");
22    for (i = 0; i < n; i++)
23        scanf("%d", &a[i]);

24    /* Parametar j predstavlja brojac prve slobodne pozicije na koju
25       se može upisati element niza koji treba da ostane u nizu. Kada
26       se nađe na element koji je paran, on se kopira na mesto a[j]
27       i poveća se vrednost brojača j. Ukoliko se nađe na element
28       koji je neparan, njega treba preskociti. */
29    for (i = 0, j = 0; i < n; i++) {
30        /* Ako je tekuci element niza a paran. */
31        if (a[i] % 2 == 0) {
32            /* Premesta se na poziciju j. */
33            a[j] = a[i];
34
35            /* Vrednost brojača j se priprema za narednu iteraciju. */
36            j++;
37        }

38        /* Ako je tekuci element niza a neparan, sa njim nista ne treba
39           raditi. */
40    }

41    /* U nizu a se sada na pozicijama od 0,...,j-1 nalaze elementi
42       koji su parni, te je njegova nova dimenzija j. */
43    n = j;

44    /* Ispisuju se elementi modifikovanog niza a. */
45    for (i = 0; i < n; i++) {
46        printf("%d ", a[i]);
47    }
48    printf("\n");

49    exit(EXIT_SUCCESS);
}
```

56 | }

#### Rešenje 3.1.31

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

#define MAKS 100

int main()
{
    /* Deklaracija potrebnih promenljivih. */
    int a[MAKS];
    int n, i, j;
    char poslednja_cifra;
    int novo_n;

    /* Ucitava se dimenzija niza i vrsi se provera ispravnosti
       ulaza. */
    printf("Unesite dimenziju niza:\n");
    scanf("%d", &n);
    if (n < 1 || n > MAKS) {
        printf("Greska: neispravan unos.\n");
        exit(EXIT_FAILURE);
    }

    /* Ucitavaju se elementi niza a. */
    printf("Unesite elemente niza a:\n");
    for (i = 0; i < n; i++)
        scanf("%d", &a[i]);

    /* Obilaze se svi elementi niza a. */
    for (i = 0, j = 0; i < n; i++) {
        /* Izdvaja se poslednja cifra tekuceg elementa. */
        poslednja_cifra = a[i] % 10;

        /* Ako je poslednja cifra nula ili je element deljiv svojom
           poslednjom cifrom, taj element se zadržava i smesta na
           poziciju j. */
        if (poslednja_cifra == 0 || a[i] % poslednja_cifra == 0) {
            a[j] = a[i];
            j++;
        }
    }

    /* Dimenzija novog niza odgovara posledjoj vrednosti brojac a j. */
    novo_n = j;

    /* Ispisuje se rezultujući niz. */
    printf("Niz a nakon izmena:\n");
    for (i = 0; i < novo_n; i++)
```

```

48     printf("%d ", a[i]);
    printf("\n");
50
    exit(EXIT_SUCCESS);
52 }

```

### Rešenje 3.1.32

```

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

#define DIM 700

/* Funkcija pomera za jedno mesto u levo elemente niza a pocevsi od
   pozicije j. Element na poziciji j se brise i na njegovo mesto se
   upisuje element na poziciji j+1, a u skladu sa tim svi ostali
   elementi posle njega u nizu se pomeraju. */
void pomeri_za_jedno_mesto(int a[], int n, int j)
{
    int i;
    for (i = j; i < n; i++)
        a[i] = a[i + 1];
}

int main()
{
    /* Deklaracija potrebnih promenljivih. */
    int n, niz[DIM], i;

    /* Ucitava se dimenzija niza i vrsi se provera ispravnosti
       ulaza. */
    printf("Unesite broj elemenata niza: ");
    scanf("%d", &n);
    if (n <= 0 || n > DIM) {
        printf("Greska: neispravan unos. \n");
        exit(EXIT_FAILURE);
    }

    /* Ucitavaju se elementi niza. */
    printf("Unesite elemente niza: ");
    for (i = 0; i < n; i++)
        scanf("%d", &niz[i]);

    /* Potrebno je krenuti od poslednjeg elementa niza i petljom ici
       ka pocetku niza (element na poziciji 0 se ne razmatra).
       Proverava se da li je element potrebno obrisati i ako jeste
       vrsi se pomeranje elemenata niza za jedno mesto u levo.
       Prednost ovog resenja u odnosu na resenje kada se krene od
       pocetka niza je u tome sto element koji se ispituje sigurno
       nije promenio svoju poziciju usled pomeranja zbog brisanja.
       Problem se moze resiti i koriscenjem pomocnog niza (uraditi za

```

### 3 Predstavljanje podataka

---

```
44     vezbu). To resenje je efikasnije, ali trosi vise resursa. */
45     for (i = n - 1; i > 0; i--)
46     {
47         if (niz[i] % i != 0) {
48             pomeri_za_jedno_mesto(niz, n, i);
49             /* Nakon brisanja elementa, smanjuje se i dimenzija niza. */
50             n--;
51         }
52     }
53     /* Ispis novog niza. */
54     printf("Novi niz:\n");
55     for (i = 0; i < n; i++)
56         printf("%d ", niz[i]);
57     printf("\n");
58     exit(EXIT_SUCCESS);
59 }
```

#### Rešenje 3.1.33

```
1  #include <stdio.h>
2  #include <stdlib.h>
3  #include <math.h>
4
5  #define MAKS 100
6
7  /* Funkcija vraća 1 ako je broj prost, a 0 u suprotnom. */
8  int prost(int x){
9      int i;
10
11     /* Brojevi 2 i 3 su prosti. */
12     if (x == 2 || x == 3)
13         return 1;
14
15     /* Parni brojevi nisu prosti. */
16     if (x % 2 == 0)
17         return 0;
18
19     /* Ako se naidje na broj koji deli broj x, onda broj x nije
20     prost. Provera se vrši za sve neparne brojeve izmedju 3 i
21     sqrt(x), jer kada bi x imao parnog delioca, onda bi i broj 2
22     delio x, a taj uslov je vec proveren. */
23     for (i = 3; i <= sqrt(x); i += 2)
24         if (x % i == 0)
25             return 0;
26
27     /* Ako nijedan od prethodnih uslova nije bio ispunjen, to znaci
28     da nijedan broj ne deli x, pa je on prost. */
29     return 1;
30 }
31
32 int main()
```

```

34 {
35     /* Deklaracije potrebnih promenljivih. */
36     int a[MAKS], b[MAKS];
37     int i, j, n_a, n_b;
38
39     /* Ucitava se dimenzija niza i vrsi se provera ispravnosti
40        ulaza. */
41     printf("Unesite broj elemenata niza: ");
42     scanf("%d", &n_a);
43     if (n_a < 1 || n_a > MAKS) {
44         printf("Greska: neispravan unos.\n");
45         exit(EXIT_FAILURE);
46     }
47
48     /* Ucitavaju se elementi niza a. */
49     printf("Unesite elemente niza: ");
50     for (i = 0; i < n_a; i++)
51         scanf("%d", &a[i]);
52
53     /* Kada se u nizu a naidje na prost element, on se upisuje u niz
54        b i uvecava se brojac za niz b. */
55     for (i = 0, j = 0; i < n_a; i++) {
56         if (prost(a[i]) == 0) {
57             b[j] = a[i];
58             j++;
59         }
60     }
61
62     /* Broj elemenata novodobijenog niza b je j. */
63     n_b = j;
64
65     /* Ispisuju se elementi niza b. */
66     for (i = 0; i < n_b; i++)
67         printf("%d ", b[i]);
68     printf("\n");
69
70     exit(EXIT_SUCCESS);
71 }

```

### Rešenje 3.1.34

```

1 #include <stdio.h>
2 #include <stdlib.h>
3
4 #define MAKS 300
5
6 int main()
7 {
8     /* Deklaracija potrebnih promenljivih. */
9     int n, niz[MAKS], i;
10

```

### 3 Predstavljanje podataka

---

```
12  /* Ucitava se dimenzija niza i vrsi se provera ispravnosti
    ulaza. */
13  printf("Unesite dimenziju niza: ");
14  scanf("%d", &n);
15  if (n <= 0 || n > MAKS) {
16      printf("Greska: neispravan unos.\n");
17      exit(EXIT_FAILURE);
18  }

19  /* Ucitavaju se elementi niza. */
20  printf("Unesite elemente niza: ");
21  for (i = 0; i < n; i++)
22      scanf("%d", &niz[i]);
23
24
25  /* Provera da li su elementi niza uredjeni neopadajuće. */
26  for (i = 0; i < n - 1; i++)
27      /* Porede se svaka dva uzastopna elementa niza. Ukoliko za bilo
28       koja takva dva elementa uslov nije ispunjen, prekida se
29       dalje ispitivanje i vraća se poruka da niz nije uredjen
30       neopadajuće. U suprotnom se nakon izlaska iz petlje ispisuje
31       poruka da je niz uredjen neopadajuće. */
32      if (niz[i] > niz[i + 1]) {
33          printf("Nije uredjen neopadajuće.\n");
34          exit(EXIT_SUCCESS);
35      }
36
37  printf("Jeste uredjen neopadajuće.\n");
38  exit(EXIT_SUCCESS);
39
40 }
```

#### Rešenje 3.1.35

```
1  #include <stdio.h>
2  #include <stdlib.h>
3  #include <stdlib.h>

4
5  /* Maksimalan broj dana u mesecu je 31, ali dani pocinju od 1, pa
6   je potrebno odvojiti 32 mesta u nizu jer se nulti ne koristi. */
7  #define MAKS_DANA 32

8
9  /* Funkcija ucitava dimenziju i elemente niza. */
10 int ucitavanje(int niz[])
11 {
12     int i, n;
13
14     printf("Unesite dimenziju niza: ");
15     scanf("%d", &n);
16
17     if (n <= 0 || n > MAKS_DANA) {
18         printf("Greska: neispravan unos.\n");
19     }
20 }
```



```
19     exit(EXIT_FAILURE);
20 }
21
22 printf("Unesite broj prodatih artikala: ");
23 for (i = 0; i < n; i++)
24     scanf("%d", &niz[i]);
25
26 return n;
27 }
28
29 int najduzi_neopadajuci(int a[], int n)
30 {
31     int i;
32     /* Na pocetku i duzina trenutne serije i duzina maksimalne serije
33        se inicijalizuju na 1. */
34     int duzina_trenutne_serije = 1;
35     int duzina_najduze_serije = 1;
36
37     for (i = 1; i < n; i++) {
38         /* Proverava se da li uzastopni elementi ispunjavaju
39            neopadajuci uslov. Ako je to slucaj uvecava se duzina
40            serije, a ako nije, duzina trenutne serije se vraca na 1,
41            kako bi se ispravno racunala duzina sledece serije. */
42         if (a[i] >= a[i - 1])
43             duzina_trenutne_serije++;
44         else
45             duzina_trenutne_serije = 1;
46
47         /* Ukoliko je trenutna duzina serije veca od duzine do sada
48            najduze serije, parametar za duzinu najduze serije se
49            postavlja na novu, vecu vrednost. */
50         if (duzina_trenutne_serije > duzina_najduze_serije)
51             duzina_najduze_serije = duzina_trenutne_serije;
52     }
53
54     return duzina_najduze_serije;
55 }
56
57 int main()
58 {
59     /* Deklaracija potrebnih promenljivih. */
60     int a[MAKS_DANA], n;
61
62     /* Ucitava se ulaz. */
63     n = ucitavanje(a);
64
65     /* Ispis rezultata. */
66     printf("Duzina najduzeg neopadajuceg prodavanja je %d.\n",
67           najduzi_neopadajuci(a, n));
68
69     exit(EXIT_SUCCESS);
70 }
```

#### Rešenje 3.1.36

```
1 #include <stdio.h>
2 #include <stdlib.h>
3 #include <stdlib.h>
4
5 #define MAKS 100
6
7 /* Funkcija koja ucitava elemente niza i vrsi proveru ispravnosti
8    ulaza. Kao povratnu vrednost vraca dimenziju ucitanog niza. */
9 int ucitavanje(int niz[])
10 {
11     int i, n;
12
13     printf("Unesite dimenziju niza: ");
14     scanf("%d", &n);
15     if (n <= 0 || n > MAKS) {
16         printf("Nedozvoljena dimenzija niza.\n");
17         exit(EXIT_FAILURE);
18     }
19
20     printf("Unesite elemente niza: ");
21     for (i = 0; i < n; i++)
22         scanf("%d", &niz[i]);
23
24     return n;
25 }
26
27 /* Funkcija koja vraca duzinu najduze serije jednakih elemenata
28    niza. */
29 int najduza_serija(int a[], int n)
30 {
31     int i;
32     /* Na pocetku i duzina trenutne serije i duzina maksimalne serije
33        se inicijalizuju na 1. */
34     int trenutna_serija = 1;
35     int najduza_serija = 1;
36
37     for (i = 1; i < n; i++) {
38         /* Proverava se da li su uzastopni elementi jednaki. Ako je to
39            slucaj uvecava se duzina serije. Ako uzastopni elementi nisu
40            jednaki serija je prekinuta i parametar za duzinu serije se
41            postavlja ponovo na 1 da bi mogla da se racuna duzina
42            sledece serije. */
43         if (a[i] == a[i - 1])
44             trenutna_serija++;
45         else
46             trenutna_serija = 1;
47
48         /* Ukoliko je trenutna duzina serije veca od duzine do sada
49            najduze serije, parametar za duzinu najduze serije se
50            postavlja na novu, vecu vrednost. */
51     }
```

```

51     if (trenutna_serija > najduza_serija)
52         najduza_serija = trenutna_serija;
53     }
54
55     return najduza_serija;
56 }
57
58 int main()
59 {
60     /* Deklaracija potrebnih promenljivih. */
61     int n, a[MAKS];
62
63     /* Ucitavanje niza. */
64     n = ucitavanje(a);
65
66     /* Ispis rezultata. */
67     printf("Duzina najduze serije je %d.\n", najduza_serija(a, n));
68
69     exit(EXIT_SUCCESS);
70 }

```

### Rešenje 3.1.37

```

1 #include <stdio.h>
2 #include <stdlib.h>
3 #include <stdlib.h>
4
5 #define DIM 100
6
7 /* Funkcija koja ucitava niz i vrši proveru ispravnosti ulaza.
8    Kao povratnu vrednost vraća dimenziju niza. */
9 int ucitavanje(int niz[])
10 {
11     int i, n;
12
13     printf("Unesite dimenziju niza: ");
14     scanf("%d", &n);
15     if (n <= 0 || n > DIM) {
16         printf("Nedozvoljena dimenzija niza.\n");
17         exit(EXIT_FAILURE);
18     }
19
20     printf("Unesite elemente niza: ");
21     for (i = 0; i < n; i++)
22         scanf("%d", &niz[i]);
23
24     return n;
25 }
26
27 /* Rešenje pod a. */
28 int podniz_uzastopnih(int a[], int n, int b[], int m)

```

```
{
30  int i, j;

32  /* Prolaze se elementi prvog niza. Svaki element prvog niza moze
    biti pocetak podniza, odnosno pocetak drugog niza. */
34  for (i = 0; i + m - 1 < n; i++) {
    /* Prolaze se elementi drugog niza. Za svaki element niza b
36     proverava se da li je jednak odgovarajucem elementu niza a.
    Za niz a razmatra se da li podniz pocinje od pozicije i. Tako
38     0-ti element niza b je na poziciji i, 1. element je na
    poziciji i+1, 2. na poziciji i+2, ..., j-ti na poziciji
40     i+j. Ako uslov nije ispunjen, petlja se prekida i proverava
    se da li na sledecoj poziciji u nizu a pocinje podniz. */
42     for (j = 0; j < m; j++)
        if (a[i + j] != b[j])
44         break;
    /* Ako petlja nije prekinuta nakon ispitivanja, brojac za niz b
46     je jednak dimenziji niza b, odnosno svi elementi niza b se
    uzastopno nalaze u nizu a. */
48     if (j == m)
        return 1;
50 }

52 /* Ukoliko niz b jeste uzastopni podniz uslov u petlji ce u nekom
    trenutku biti ispunjen i iz petlje i funkcije ce se izaci sa
54     return naredbom. Ipak, ako se to nije desilo i dalje se
    izvrsava funkcija, onda niz b nije uzastopni podniz. */
56 return 0;
}

58 /* Resenje pod b. */
60 int podniz(int a[], int n, int b[], int m)
{
62     int i, j;

64     /* Petljom se prolaze elementi niza a. */
    for (i = 0, j = 0; i < n && j < m; i++)
66         /* Svaki put kada se naidje na element niza b, brojac za niz b
        se uvecava i proverava se da li se sledeci element niza b
68         nalazi u nizu a. */
        if (a[i] == b[j])
70             j++;

72     /* Ukoliko se pronadju svi elementi niza b u nizu a, onda je
        brojac za niz b jednak dimenziji niza b. U tom slucaju se
74         vraca vrednost 1, odnosno da niz jeste podniz. */
    return j == m;
76 }

78 int main()
{
80     /* Deklaracija potrebnih promenljivih. */
```

```

82  int n, a[DIM];
    int m, b[DIM];

84  /* Ucitavanje nizova. */
    n = ucitavanje(a);
86  m = ucitavanje(b);

88  /* a) */
    if (podniz_uzastopnih(a, n, b, m))
90      printf("Elementi drugog niza cine uzastopni podniz "
              "prvog niza.\n");
92  else
    printf("Elementi drugog niza ne cine uzastopni podniz "
          "prvog niza.\n");

96  /* b) */
    if (podniz(a, n, b, m))
98      printf("Elementi drugog niza cine podniz prvog niza.\n");
    else
100     printf("Elementi drugog niza ne cine podniz prvog niza.\n");
102  exit(EXIT_SUCCESS);
}

```

### Rešenje 3.1.38

```

1  #include <stdio.h>
   #include <stdlib.h>
3
   #define MAKS 100
5
   /* Funkcija koja prebrojava koliko puta se pojavljuje svaki element
   niza a. */
7  void brojanje(int a[], int b[], int n)
   {
9      int i;

11
   /* Niz b se inicijalizuje nulama jer se za svaki element postavi
   da se pojavljuje 0 puta u nizu a. */
13  for (i = 1; i <= n; i++)
15      b[i] = 0;

17  /* Peljom se prolazi kroz niz a i za svaki element a[i] uvecava
   se broj njegovog pojavljivanja u nizu b. Na primer, ako je
19  a[3] = 7, onda treba uvecati broj pojavljivanja broja 7, a to
   je b[7]++, sto se krace moze zapisati kao b[a[3]]++.
21  Pretpostavlja se da je niz a dobro zadat, odnosno da su sve
   njegove vrednosti u intervalu od 1 do n. */
23  for (i = 0; i < n; i++)
       b[a[i]]++;
25 }

```

### 3 Predstavljanje podataka

---

```
27 int main()
28 {
29     int a[MAKS];
30     /* Niz b moze imati index MAKS (jer niz b se posmatra od 1 do
31        MAKS), pa zato njegova dimenzija mora biti za jedan veca. */
32     int b[MAKS + 1];
33     int i, n;
34
35     /* Ucitava se dimenzija niza i vrsi se provera ispravnosti
36        ulaza. */
37     printf("Unesite broj elemenata niza: ");
38     scanf("%d", &n);
39     if (n <= 0 || n > MAKS) {
40         printf("Neispravna dimenzija niza.\n");
41         exit(EXIT_FAILURE);
42     }
43
44     /* Ucitavaju se elementi niza. */
45     printf("Unesite elemente niza:\n");
46     for (i = 0; i < n; i++) {
47         scanf("%d", &a[i]);
48         /* Niz a moze sadrzati elemente koji nisu u opsegu od 1 do n. U
49            tom slucaju taj niz nije permutacija. */
50         if (a[i] <= 0 || a[i] > n) {
51             printf("Uneti niz nije permutacija.\n");
52             exit(EXIT_SUCCESS);
53         }
54     }
55
56     brojanje(a, b, n);
57
58     /* Ukoliko se svaki element niza a javlja tacno jednom u nizu a,
59        onda niz a jeste permutacija. Ovo svojstvo se proverava
60        koriscenjem dobijenog niza b. */
61     for (i = 1; i <= n; i++)
62         if (b[i] != 1) {
63             printf("Uneti niz nije permutacija.\n");
64             exit(EXIT_SUCCESS);
65         }
66
67     /* Ukoliko prethodna petlja nije ranije zavrшена (i nije se
68        izaslo iz programa), onda svaki element niza b je jednak 1,
69        odnosno svaki element se pojavljuje tacno jednom, pa niz a
70        jeste permutacija. */
71     printf("Uneti niz je permutacija.\n");
72
73     exit(EXIT_SUCCESS);
74 }
```

#### Rešenje 3.1.39

```
1  #include <stdio.h>
2  #include <stdlib.h>
3
4  #define BROJ_CIFARA 10
5
6  /* Funkcija izdvaja cifru po cifru broja i uvecava odgovarajuci
7     element niza koji odgovara brojacu za tu cifru. Na primer,
8     za broj=1123, po zavrsetku ove funkcije niz[1] ce imati vrednost
9     2 jer se cifra 1 pojavljuje 2 puta, niz[2] i niz[3] ce imati
10    vrednost 1, a svi ostali elementi niza ce imati vrednost 0. */
11 void analiza_cifara(int broj, int niz[])
12 {
13     int c, i;
14
15     /* Inicijalizacija svih brojaca na nule. */
16     for (i = 0; i < BROJ_CIFARA; i++)
17         niz[i] = 0;
18
19     do {
20         c = broj % 10;
21         niz[c]++;
22         broj /= 10;
23     }
24     while (broj);
25 }
26
27 int main()
28 {
29     /* Niz cifrex predstavlja brojace za cifre broja x. Niz cifrey
30        predstavlja brojace za cifre broja y. */
31     int cifrex[BROJ_CIFARA], cifrey[BROJ_CIFARA];
32     int x, y, i, indikator;
33
34     /* Ucitavaju se brojevi x i y. */
35     printf("Unesite dva broja: ");
36     scanf("%d%d", &x, &y);
37
38     /* Za slucaj da su unete vrednosti negativne, posmatra se njihova
39        apsolutna vrednost. Ovo je opravdano iz razloga sto se brojevi
40        x i -x zapisuju istim ciframa. */
41     x = abs(x);
42     y = abs(y);
43
44     /* Popunjavaju se nizovi sa brojacima cifara. */
45     analiza_cifara(x, cifrex);
46     analiza_cifara(y, cifrey);
47
48     /* Promenljiva indikator sluzi za pracenje da li su oba broja
49        sastavljena od istih cifara. */
50     indikator = 1;
51 }
```

```
53     for (i = 0; i < BROJ_CIFARA; i++) {
54         /* Ako se broj pojavljivanja cifre i u zapisu broja x razlikuje
55            od broja pojavljivanja cifre i u zapisu broja y, brojevi se
56            ne zapisuju istim ciframa. Zato se vrednost indikatora moze
57            postaviti na 0 i prekinuti dalje upoređivanje broja
58            pojavljivanja. */
59         if (cifrey[i] != cifrex[i]) {
60             indikator = 0;
61             break;
62         }
63     }
64
65     /* Ako je vrednost promenljive indikator ostala 1, to znaci da u
66        petlji nije pronadjena cifra koja se ne pojavljuje isti broj
67        puta u zapisima brojeva x i y. Zato se moze zakljuciti da se
68        brojevi zapisuju istim ciframa. */
69     if (indikator)
70         printf("Brojevi se zapisuju istim ciframa.\n");
71     else
72         printf("Brojevi se ne zapisuju istim ciframa.\n");
73
74     exit(EXIT_SUCCESS);
75 }
```

## 3.3 Pokazivači

**Zadatak 3.3.1** Napisati funkciju koja uređuje svoja dva celobrojna argumenta tako da se u prvom nalazi manja vrednost, a u drugom veća. Napisati program koji učitava dva cela broja i ispisuje uređene brojeve.

#### Primer 1

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite dva broja: 2 5
|| Uredjene promenljive: 2, 5
```

#### Primer 2

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite dva broja: 11 -4
|| Uredjene promenljive: -4, 11
```

[Rešenje 3.3.1]

**Zadatak 3.3.2** Napisati funkciju koja datu boju u *rgb* formatu konvertuje u boju u *cmv* formatu po sledećim formulama:

$$c = 1 - r/255$$

$$m = 1 - g/255$$

$$y = 1 - b/255$$



Napisati program koji učitava boju u *rgb* formatu i ispisuje vrednosti unete boje u *cmy* formatu. U slučaju neispravnog unosa, ispisati odgovarajuću poruku o grešci. NAPOMENA: *Vrednosti boja u rgb formatu su u opsegu [0, 255]*.

*Primer 1*

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite boju u rgb formatu: 56 111 24
|| cmy: (0.78, 0.56, 0.91)
```

*Primer 2*

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite boju u rgb formatu: 156 -90 5
|| Greska: neispravan unos.
```

*Primer 3*

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite boju u rgb formatu: 9 0 237
|| cmy: (0.96, 1.00, 0.07)
```

*Primer 4*

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite boju u rgb formatu: 300 11 27
|| Greska: neispravan unos.
```

[Rešenje 3.3.2]

**Zadatak 3.3.3** Napisati funkciju koja za dve prave date svojim koeficijentima pravca i slobodnim članovima određuje njihovu tačku preseka. Funkcija treba da vrati jedinicu ako se prave seku i nulu ako nemaju tačku preseka (ako su paralelne). Napisati program koji učitava podatke o pravama i ukoliko prave imaju presek, ispisuje koordinate tačke preseka, a ako nemaju, ispisuje odgovarajuću poruku.

*Primer 1*

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite k i n za prvu pravu: 4 5
|| Unesite k i n za drugu pravu: 11 -4
|| Prave se seku u tacki (1.29,10.14).
```

*Primer 2*

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite k i n za prvu pravu: 0.5 -4.7
|| Unesite k i n za drugu pravu: 0.5 9.1
|| Prave su paralelne.
```

[Rešenje 3.3.3]

**Zadatak 3.3.4** Napisati funkciju koja za dva cela broja izračunava njihov količnik i ostatak pri deljenju. Funkcija treba da vrati vrednost jedinicu ukoliko je uspešno izračunala vrednosti, a nulu ukoliko deljenje nije moguće. Napisati program koji učitava dva cela broja i ispisuje njihov količnik i ostatak pri deljenju. U slučaju neispravnog unosa, ispisati odgovarajuću poruku o grešci.

*Primer 1*

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite brojeve: 4 5
|| Kolicnik: 0
|| Ostatak: 4
```

*Primer 2*

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite brojeve: 4 0
|| Greska: neispravan unos.
```

*Primer 3*

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite brojeve: -123 11
|| Kolicnik: -11
|| Ostatak: -2
```

[Rešenje 3.3.4]

**Zadatak 3.3.5** Napisati funkciju koja za dužinu trajanja filma koje je dato u sekundama, određuje ukupno trajanje filma u satima, minutama i sekundama. Napisati program koji učitava trajanje filma u sekundama i ispisuje odgovarajuće vreme trajanja u formatu *broj\_sati:broj\_minuta:broj\_sekundi*. U slučaju neispravnog unosa, ispisati odgovarajuću poruku o grešci.

*Primer 1*

```
|| INTERAKCIJA SA PROGRAMOM:  
| Trajanje fima u sekundama: 5000  
| 1h:23m:20s
```

*Primer 2*

```
|| INTERAKCIJA SA PROGRAMOM:  
| Trajanje fima u sekundama: -300  
| Greska: neispravan unos.
```

*Primer 3*

```
|| INTERAKCIJA SA PROGRAMOM:  
| Trajanje fima u sekundama: 2500  
| 0h:41m:40s
```

*Primer 4*

```
|| INTERAKCIJA SA PROGRAMOM:  
| Trajanje fima u sekundama: 7824  
| 2h:10m:24s
```

[Rešenje 3.3.5]

**Zadatak 3.3.6** Napisati funkciju `void interpunkcija(int* broj_tacaka, int* broj_zareza)` koja sa ulaza učitava karakter po karakter sve do kraja ulaza i prebrojava sva pojavljivanja karaktera tačka i sva pojavljivanja karaktera zarez. Napisati program koji za uneti tekst ispisuje koliko puta se pojavila tačka, a koliko puta se pojavio zarez.

*Primer 1*

```
|| INTERAKCIJA SA PROGRAMOM:  
| Unesite tekst:  
| a.b.c.d  
| a,b,,c,d,e  
| Broj tacaka: 3  
| Broj zareza: 5
```

*Primer 2*

```
|| INTERAKCIJA SA PROGRAMOM:  
| Unesite tekst:  
| .....789.....  
| Broj tacaka: 10  
| Broj zareza: 0
```

*Primer 3*

```
|| INTERAKCIJA SA PROGRAMOM:  
| Unesite tekst:  
| sunce  
| Broj tacaka: 0  
| Broj zareza: 0
```

[Rešenje 3.3.6]

**Zadatak 3.3.7** Napisati funkciju `void par_nepar(int a[], int n, int parni[], int* np, int neparni[], int* nn)` koja razbija niz *a* na niz parnih i niz neparnih brojeva. Pokazivači *np* i *nn* redom treba da sadrže broj elemenata niza parnih tj. niza neparnih elemenata. Maksimalan broj elemenata niza je 50.

Napisati program koji učitava dimenziju niza, a zatim i elemente niza i ispisuje odgovarajuće nizove parnih, odnosno neparnih elemenata unetog niza. U slučaju neispravnog unosa, ispisati odgovarajuću poruku o grešci.

*Primer 1*

```
INTERAKCIJA SA PROGRAMOM:
Unesite broj elemenata niza: 8
Unesite elemente niza:
1 8 9 -7 -16 24 77 4
Niz parnih brojeva: 8 -16 24 4
Niz neparnih brojeva: 1 9 -7 77
```

*Primer 2*

```
INTERAKCIJA SA PROGRAMOM:
Unesite broj elemenata niza: 5
Unesite elemente niza:
2 4 6 8 -11
Niz parnih brojeva: 2 4 6 8
Niz neparnih brojeva: -11
```

*Primer 3*

```
INTERAKCIJA SA PROGRAMOM:
Unesite broj elemenata niza: 2
Unesite elemente niza:
-15 15
Niz parnih brojeva:
Niz neparnih brojeva: -15 15
```

*Primer 4*

```
INTERAKCIJA SA PROGRAMOM:
Unesite broj elemenata niza: 0
Greska: neispravan unos.
```

[Rešenje 3.3.7]

**Zadatak 3.3.8** Napisati funkciju `void min_max(float a[], int n, float* najmanji, float* najveći)` koja izračunava najmanji i najveći element niza *a* dužine *n*. Napisati program koji učitava niz realnih brojeva maksimalne dužine 50 i ispisuje vrednosti namanjeg i najvećeg elementa niza, zaokružene na tri decimala. U slučaju neispravnog unosa, ispisati odgovarajuću poruku o grešci.

*Primer 1*

```
INTERAKCIJA SA PROGRAMOM:
Unesite broj elemenata niza: 5
Unesite elemente niza:
24.16 -32.11 999.25 14.25 11
Najmanji: -32.110
Najveci: 999.250
```

*Primer 2*

```
INTERAKCIJA SA PROGRAMOM:
Unesite broj elemenata niza: 4
Unesite elemente niza:
-5.126 -18.29 44 29.268
Najmanji: -18.290
Najveci: 44.000
```

*Primer 3*

```
INTERAKCIJA SA PROGRAMOM:
Unesite broj elemenata niza: 1
Unesite elemente niza:
4.16
Najmanji: 4.160
Najveci: 4.160
```

*Primer 4*

```
INTERAKCIJA SA PROGRAMOM:
Unesite broj elemenata niza: -3
Greska: neispravan unos.
```

[Rešenje 3.3.8]

## 3.4 Rešenja

### Rešenje 3.3.1

```
1  #include <stdio.h>
2  #include <stdlib.h>
3
4  /* Argumenti funkcije uredi_pogresno, promenljive a i b,
5     predstavljaju lokalne promenljive za ovu funkciju i prestaju
6     da postoje po zavrsetku funkcije. Zbog toga se efekti razmene
7     vrednosti promenljivih a i b u slucaju da je a>b ne vide u
8     glavnom programu. */
9  void uredi_pogresno(int a, int b)
10 {
11     int pom;
12     if (a > b) {
13         pom = a;
14         a = b;
15         b = pom;
16     }
17 }
18
19 /* Argumenti funkcije uredi_tacno, promenljive pa i pb, takodje su
20     lokalne promenljive za ovu funkciju i prestaju da postoje kada
21     se funkcija zavrshi. Njima prosledjujemo adrese promenljivih a i
22     b koje zelimo da razmenimo u slucaju da je a>b.
23
24     Promenljivoj a pristupamo preko pokazivacke promenljive pa sa
25     *pa i slicno, promenljivoj b pristupamo sa *pb.
26
27     Vrednosti promenljivih *pa i *pb razmenjujemo kao i vrednosti
28     bilo koje dve celobrojne promenljive. */
29 void uredi_tacno(int *pa, int *pb)
30 {
31     int pom;
32     if (*pa > *pb) {
33         pom = *pa;
34         *pa = *pb;
35         *pb = pom;
36     }
37 }
38
39 int main()
40 {
41     /* Deklaracija potrebnih promenljivih. */
42     int a, b;
43
44     /* Ucitavaju se vrednosti dva cela broja. */
45     printf("Unesite dva broja:");
46     scanf("%d%d", &a, &b);
```

```

47  /* Neispravan nacin:
49  uredi_pogresno(a, b);
    printf("Uredjene promenljive: %d, %d\n", a, b); */

51

53  /* Funkcija uredi_tacno kao argumente prima dve pokazivacke
    promenljive (int*,int*). Zbog toga joj je u pozivu funkcije
    neophodno proslediti adrese promenljivih koje zelimo da
55  uredimo rastuce, &a i &b. */
    uredi_tacno(&a, &b);
57  printf("Uredjene promenljive: %d, %d\n", a, b);

59  exit(EXIT_SUCCESS);
}

```

### Rešenje 3.3.2

```

1  #include <stdio.h>
    #include <stdlib.h>
3
    #define MIN_RGB 0
5  #define MAKS_RGB 255

7  /* Funkcija koja vrsi konverziju boje iz rgb formata u cmy format.
    Kako pomocu naredbe return ne mozemo vratiti vise od jedne
9  vrednosti, neophodno je da promenljive ciju vrednost zelimo
    da promenimo prenesemo preko pokazivaca.*/
11 void rgb_u_cmy(float *a, float *b, float *c)
    {
13     *a = 1 - *a / 255;
        *b = 1 - *b / 255;
15     *c = 1 - *c / 255;
    }

17

19  /* Funkcija koja proverava da li je vrednost boje u ispravnom
    opsegu. */
    int ispravna_rgb_vrednost(float a)
21  {
        if (a < MIN_RGB || a > MAKS_RGB)
23         return 0;
        return 1;
25  }

27  int main()
    {
29     /* Deklaracija potrebnih promenljivih. */
        float a, b, c;

31

        /* Ucitava se vrednost boje u rgb formatu. */
33     printf("Unesite boju u rgb formatu: ");
        scanf("%f%f%f", &a, &b, &c);

```

### 3 Predstavljanje podataka

```
35  /* Vrsi se provera ispravnosti ulaza. */
37  if (!ispravna_rgb_vrednost(a) || !ispravna_rgb_vrednost(b) ||
    !ispravna_rgb_vrednost(c))
39  {
    printf("Greska: neispravan unos.\n");
41    exit(EXIT_FAILURE);
  }
43
45  /* Konverzija boje i ispis rezultata. */
46  rgb_u_cmy(&a, &b, &c);
47  printf("cmy: (%.2f,%.2f,%.2f)\n", a, b, c);
48
49  exit(EXIT_SUCCESS);
}
```

#### Rešenje 3.3.3

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

/* Funkcija koja racuna presek pravih  $y = k_1 * x + n_1$  i
    $y = k_2 * x + n_2$ . Koordinate preseka (ako postoji) se upisuju
   na adrese px i py. Kao povratna vrednost funkcije se vraca
   jedinica ukoliko presek postoji, a nula inace. */
8  int presek(float k1, float n1, float k2, float n2, float *px,
             float *py)
10 {
    /* Ako je koeficijent pravca jednak, prave su paralelne.
       Povratna vrednost funkcije je 0, kao indikator da
       nema presečne tačke. */
12    if (k1 == k2)
14        return 0;

    /* Koordinate preseka se upisuju na adrese (px, py). */
16    *px = -(n1 - n2) / (k1 - k2);
18    *py = k1 * (*px) + n1;

    /* Funkcija vraca 1 kao indikator da je presek uspesno
       izracunat. */
20    return 1;
22 }

24

26 int main()
27 {
    /* Deklaracije potrebnih promenljivih. */
28    float k1, k2, n1, n2;
30    float x, y;

    /* Ucitavaju se parametri za dve prave. */
32    printf("Unesite k i n za prvu pravu:");
```

```

34 scanf("%f%f", &k1, &n1);
printf("Unesite k i n za drugu pravu:");
36 scanf("%f%f", &k2, &n2);

38 /* Ispis rezultata. */
if (preseka(k1, n1, k2, n2, &x, &y))
40 printf("Prave se seku u tacki (%.2f,%.2f).\n", x, y);
else
42 printf("Prave su paralelne.\n");

44 exit(EXIT_SUCCESS);
}

```

### Rešenje 3.3.4

```

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

4 int kolicnik_ostatak(int a, int b, int* pk, int* po)
{
6     if(b == 0)
        return 0;

8     *pk = a/b;
10    *po = a%b;
    return 1;
12 }

14 int main()
{
16     /* Deklaracija potrebnih promenljivih. */
    int a, b, kolicnik, ostatak;

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

22     /* Ispis rezultata. */
24     if(kolicnik_ostatak(a, b, &kolicnik, &ostatak)){
        printf("Kolicnik: %d\n", kolicnik);
26         printf("Ostatak: %d\n", ostatak);
    }
28     else
    {
30         printf("Greska: neispravan unos.\n");
        exit(EXIT_FAILURE);
32     }

34     exit(EXIT_SUCCESS);
}

```

#### Rešenje 3.3.5

```
1 #include <stdio.h>
2 #include <stdlib.h>
3
4 void konverzija(int trajanje, int* psati, int* pminuti, int* psekunde
5 )
6 {
7     *psati = trajanje / 3600;
8     trajanje -= *psati * 3600;
9
10    *pminuti = trajanje/60;
11    trajanje -= *pminuti * 60;
12
13    *psekunde = trajanje;
14 }
15
16 int main()
17 {
18     /* Deklaracija potrebnih promenljivih. */
19     int trajanje, sati, minuti, sekunde;
20
21     /* Ucitava se trajanje u sekundama. */
22     printf("Trajanje filma u sekundama: ");
23     scanf("%d", &trajanje);
24
25     /* Vrsi se provera ispravnosti ulaza. */
26     if(trajanje < 0)
27     {
28         printf("Greska: neispravan unos.\n");
29         exit(EXIT_FAILURE);
30     }
31
32     /* Racunanje rezultata. */
33     konverzija(trajanje, &sati, &minuti, &sekunde);
34
35     /* Ispis rezultata. */
36     printf("%dh:%dm:%ds\n", sati, minuti, sekunde);
37     exit(EXIT_FAILURE);
38 }
```

#### Rešenje 3.3.6

```
1 #include <stdio.h>
2 #include <stdlib.h>
3
4 /* Funkcija koja ucitava karakter po karakter sa ulaza i prebrojava
5    koliko puta se pojavio karakter '.' i koliko puta se pojavio
6    karakter ','. Ucitavanje se zaustavlja kada se dodje do kraja
7    ulaza (EOF-a). */
```



```

9 void interpunkcija(int *br_tacaka, int *br_zareza)
10 {
11     /* Deklaracije i inicijalizacije pomocnih promenljivih. */
12     int tacke = 0, zarezi = 0;
13     char c;
14
15     /* Ucitavanje i prebrojavanje trazениh karaktera. */
16     while ((c = getchar()) != EOF) {
17         if (c == '.')
18             tacke++;
19
20         if (c == ',')
21             zarezi++;
22     }
23
24     /* Smestanje rezultata na prosledjene adrese. */
25     *br_tacaka = tacke;
26     *br_zareza = zarezi;
27 }
28
29 int main()
30 {
31     /* Deklaracije potrebnih promenljivih. */
32     int br_tacaka, br_zareza;
33
34     /* Ucitavanje i obrada teksta. */
35     printf("Unesite tekst: \n");
36     interpunkcija(&br_tacaka, &br_zareza);
37
38     /* Ispis rezultata. */
39     printf("Broj tacaka: %d\n", br_tacaka);
40     printf("Broj zareza: %d\n", br_zareza);
41
42     exit(EXIT_SUCCESS);
43 }

```

### Rešenje 3.3.7

```

1  #include <stdio.h>
2  #include <stdlib.h>
3
4  #define MAKS 50
5
6  /* Funkcija koja od niza a formira dva niza: niz parnih elemenata
7   niza a i niz neparnih elemenata niza a. Duzine rezultujucih
8   nizova se upisuju na adrese np i nn. */
9  void par_nepar(int a[], int n, int parni[], int *np,
10                int neparni[], int *nn)
11 {
12     int i, j, k;
13

```

```
15  /* Promenljiva i je brojac u originalnom nizu i on se uvecava u
    svako iteraciji.
    Promenljiva j je brojac za niz parnih brojeva i on treba da se
17  uveca svaki put kada se naidje na novi element ovog niza.
    Promenljiva k je brojac za niz neparnih brojeva i on treba da
19  se uveca sveki put kada se naidje na novi element ovog niza. */
for (i = 0, j = 0, k = 0; i < n; i++) {
21  if (a[i] % 2 == 0) {
    parni[j] = a[i];
23  j++;
  }
25  else {
    neparni[k] = a[i];
27  k++;
  }
29 }

31 /* Na kraju petlje, u promenljivoj j se nalazi podatak o broju
    elementa niza parni[], a u promenljivoj k podatak o broju
33  elementa niza neparni[]. Ove vrednosti se upisuju na adrese
    np i nn. */
35  *np = j;
    *nn = k;
37 }

39 void ispisi(int niz[], int n)
{
41  int i;
    for (i = 0; i < n; i++)
43  printf("%d ", niz[i]);
    printf("\n");
45 }

47 int main()
{
49  /* Deklaracije potrebnih promenljivih. */
    int n, n1, n2;
51  int a[MAKS], parni[MAKS], neparni[MAKS];
    int i;
53
    /* Ucitava se dimenzija niza. */
55  printf("Unesite broj elemenata niza: ");
    scanf("%d", &n);
57
    /* Vrsi se provera ispravnosti ulaza. */
59  if (n < 0 || n > MAKS) {
    printf("Greska: neispravan unos.\n");
61  exit(EXIT_FAILURE);
  }
63
    /* Ucitavaju se elementi niza. */
65  printf("Unesite elemente niza: ");
```

```

67     for (i = 0; i < n; i++)
        scanf("%d", &a[i]);

69     /* Nizovi parni[] i neparni[] se popunjavaju odgovarajucim
        vrednostima. */
71     par_nepar(a, n, parni, &n1, neparni, &n2);

73     /* Ispis niza parni[] koji ima n1 elemenata. */
    printf("Niz parnih brojeva: ");
75     ispisi(parni, n1);

77     /* Ispis niza neparni[] koji ima n2 elemenata. */
    printf("Niz neparnih brojeva: ");
79     ispisi(neparni, n2);

81     exit(EXIT_SUCCESS);
}

```

### Rešenje 3.3.8

```

1  #include <stdio.h>
   #include <stdlib.h>
3
   #define MAKS 50
5
   /* Funkcija koja racuna najmanji i najveći element niza a. */
7  void min_max(float a[], int n, float *najmanji, float *najveci)
   {
9     int i;

11     /* Vrednosti minimuma i maksimuma se inicijalizuju na vrednost
        prvog clana niza. */
13     *najmanji = a[0];
        *najveci = a[0];
15
17     /* U petlji se prolazi kroz ostale clanove niza i po potrebi se
        vrsi azuriranje najmanje i najveće vrednosti. */
        for (i = 1; i < n; i++) {
19         if (a[i] > *najveci)
            *najveci = a[i];

21         if (a[i] < *najmanji)
23             *najmanji = a[i];
        }
25
27     /* Na kraju petlje, na adresama najmanji i najveći se nalaze
        trazene vrednosti. */
    }
29
31 int main()
   {

```

### 3 Predstavljanje podataka

```
/* Deklaracija potrebnih promenljivih. */
33 int i, n;
float a[MAKS], min, max;

35 /* Ucitava se dimenzija niza. */
37 printf("Unesite broj elemenata niza: ");
scanf("%d", &n);

39 /* Vrsi se provera ispravnosti ulaza. */
41 if (n < 0 || n > MAKS) {
    printf("Greska: neispravan unos.\n");
43     exit(EXIT_FAILURE);
}

45 /* Ucitavaju se elementi niza. */
47 printf("Unesite elemente niza:\n");
for (i = 0; i < n; i++)
49     scanf("%f", &a[i]);

51 /* Racunaju se vrednosti najmanjeg i najveceg elementa. */
min_max(a, n, &min, &max);

53 /* Ispis rezultata. */
55 printf("Najmanji: %.3f\n", min);
printf("Najveci: %.3f\n", max);

57 exit(EXIT_SUCCESS);
59 }
```

## 3.5 Niske

**Zadatak 3.5.1** Napisati funkciju `void konvertuj(char s[])` koja menja nisku `s` tako što mala slova zamenjuje odgovarajućim velikim slovima, a velika slova zamenjuje odgovarajućim malim slovima. Napisati program koji učitava nisku maksimalne dužine 10 karaktera i ispisuje konvertovanu nisku.

#### Primer 1

```
INTERAKCIJA SA PROGRAMOM:
Unesite nisku: BeoGrad
bEOgRAD
```

#### Primer 2

```
INTERAKCIJA SA PROGRAMOM:
Unesite nisku: A+B+C
a+b+c
```

#### Primer 3

```
INTERAKCIJA SA PROGRAMOM:
Unesite nisku: 12345
12345
```

[Rešenje 3.5.1]

**Zadatak 3.5.2** Napisati funkciju `void ubaci_zvezdice(char s[])` koja menja nisku `s` tako što u njoj svaki drugi karakter zameni zvezdicom. Napisati

program koji učitava nisku maksimalne dužine 20 karaktera i ispisuje izmenjenu nisku.

*Primer 1*

```
|| INTERAKCIJA SA PROGRAMOM:
   Unesite nisku: *a*b*c*
   Izmenjena niska: *****
```

*Primer 2*

```
|| INTERAKCIJA SA PROGRAMOM:
   Unesite nisku: zima
   Izmenjena niska: z*m*
```

*Primer 3*

```
|| INTERAKCIJA SA PROGRAMOM:
   Unesite nisku: 123abc789
   Izmenjena niska: 1*3*b*7*9
```

[Rešenje 3.5.2]

**Zadatak 3.5.3** Napisati program koji vrši poređenje niski. Napisati funkcije:

- (a) `int poredjenje(char s1[], char s2[])` koja vraća jedinicu ako su  $s_1$  i  $s_2$  jednake niske, a nulu inace.
- (b) `void u_velika_slova(char s[])` koja pretvara sva slova niske  $s$  u velika slova, a ostale karaktere ne menja.

Program učitava dve reči maksimalne dužine 20 karaktera i ispituje da li su unete reči jednake. Pri poređenju treba zanemariti razliku između malih i velikih slova.

*Primer 1*

```
|| INTERAKCIJA SA PROGRAMOM:
   Unesite niske:
   isPit2010
   IsPit2010
   Niske su jednake.
```

*Primer 2*

```
|| INTERAKCIJA SA PROGRAMOM:
   Unesite niske:
   Prog1
   prog2
   Niske nisu jednake.
```

*Primer 3*

```
|| INTERAKCIJA SA PROGRAMOM:
   Unesite niske:
   jun
   JUNSKI
   Niske nisu jednake.
```

[Rešenje 3.5.3]

**Zadatak 3.5.4** Napisati program koji proverava da li se uneta niska završava samoglasnikom. Napisati funkcije:

- (a) `int samoglasnik(char c)` — ispituje da li je karakter  $c$  samoglasnik;
- (b) `int samoglasnik_na_kraju(char s[])` — ispituje da li se niska  $s$  završava samoglasnikom.

Pretpostaviti da je uneta niska maksimalne dužine 20 karaktera.

*Primer 1*

```
|| INTERAKCIJA SA PROGRAMOM:
   Unesite nisku: kestenje
   Niska se završava samoglasnikom.
```

*Primer 2*

```
|| INTERAKCIJA SA PROGRAMOM:
   Unesite nisku: vetar
   Niska se ne završava samoglasnikom.
```

### 3 Predstavljanje podataka

---

#### Primer 3

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesite nisku: OLUJA  
|| Niska se završava samoglasnikom.
```

#### Primer 4

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesite nisku: Programiranje1  
|| Niska se ne završava samoglasnikom.
```

[Rešenje 3.5.4]

**Zadatak 3.5.5** Napisati funkciju `int sadrzi_veliko(char s[])` koja proverava da li niska *s* sadrži veliko slovo. Napisati program koji za učitano nisku maksimalne dužine 20 karaktera proverava da li sadrži veliko slovo i ispisuje odgovarajuću poruku.

#### Primer 1

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesite nisku:  
|| naocare  
|| Ne sadrzi veliko slovo.
```

#### Primer 2

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesite nisku:  
|| DiopTrija0.75  
|| Sadrzi veliko slovo.
```

#### Primer 3

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesite nisku:  
|| 21.06.2017.  
|| Ne sadrzi veliko slovo.
```

[Rešenje 3.5.5]

**Zadatak 3.5.6** Napisati program koji za učitano nisku *s* i karakter *c* ispituje da li se *c* pojavljuje u niski *s*. Ako se pojavljuje, program treba da ispiše indeks prvog pojavljivanja karaktera *c* u niski *s*, a u suprotnom -1. Pretpostaviti da niska može da ima najviše 20 karaktera.

#### Primer 1

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesite nisku:  
|| bazen  
|| Unesite karakter:  
|| z  
|| 2
```

#### Primer 2

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesite nisku:  
|| lezaljka  
|| Unesite karakter:  
|| a  
|| 3
```

#### Primer 3

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesite nisku:  
|| limunada  
|| Unesite karakter:  
|| b  
|| -1
```

[Rešenje 3.5.6]

**Zadatak 3.5.7** Napisati funkciju `int podniska(char s[], char t[])` koja proverava da li je niska *t* podniska niske *s*. Napisati program koji učitava dve niske maksimalne dužine 10 karaktera i testira rad napisane funkcije.

*Primer 1*

```

INTERAKCIJA SA PROGRAMOM:
Unesite nisku s: abcde
Unesite nisku t: bcd
t je podniska niske s.

```

*Primer 2*

```

INTERAKCIJA SA PROGRAMOM:
Unesite nisku s: abcde
Unesite nisku t: bCd
t nije podniska niske s.

```

*Primer 3*

```

INTERAKCIJA SA PROGRAMOM:
Unesite nisku s: abcde
Unesite nisku t: def
t nije podniska niske s.

```

[Rešenje 3.5.7]

**Zadatak 3.5.8** Napisati funkciju `void skрати(char s[])` koja uklanja beline sa kraja date niske. Napisati program koji učitava liniju maksimalne dužine 100 karaktera i ispisuje učitanu i izmenjenu nisku između zvezdica.

*Primer 1*

```

INTERAKCIJA SA PROGRAMOM:
Unesite nisku: rep belina
Ucitana niska: *rep belina
Izmenjena niska: *rep belina*

```

*Primer 2*

```

INTERAKCIJA SA PROGRAMOM:
Unesite nisku: tri tabulatora na kraju
Ucitana niska: *tri tabulatora na kraju
Izmenjena niska: *tri tabulatora na kraju*

```

[Rešenje 3.5.8]

**Zadatak 3.5.9** Napisati funkciju `void ukloni_slova(char s[])` koja iz niske `s` uklanja sva mala i sva velika slova. Napisati program koji za učitanu nisku maksimalne dužine 20 karaktera ispisuje odgovarajuću izmenjenu nisku.

*Primer 1*

```

INTERAKCIJA SA PROGRAMOM:
Unesite nisku:
a1b2c3def
123

```

*Primer 2*

```

INTERAKCIJA SA PROGRAMOM:
Unesite nisku:
1+2=3
1+2=3

```

*Primer 3*

```

INTERAKCIJA SA PROGRAMOM:
Unesite nisku:
maLaVELIKA

```

[Rešenje 3.5.9]

**Zadatak 3.5.10** Napisati funkciju `void ukloni(char *s)` koja iz niske uklanja sva slova iza kojih neposredno sledi slovo koje je u abecedi nakon njih,

### 3 Predstavljanje podataka

---

pri čemu se veličina slova zanemaruje. Testirati funkciju u programu za učitano liniju od najviše 100 karaktera.

#### Primer 1

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

#### Primer 2

```
|| INTERAKCIJA SA PROGRAMOM:
|| Danas je 10 stepeni.
|| Dns j 10 tpni.
```

#### Primer 3

```
|| INTERAKCIJA SA PROGRAMOM:
|| Ima vetra, kise i hladnoce.
|| ma vtra, se i loe.
```

[Rešenje 3.5.10]

**Zadatak 3.5.11** Napisati program koji učitava nisku  $s$  i formira nisku  $t$  trostrukim nadovezivanjem niske  $s$ . Možemo pretpostaviti da niska  $s$  sadrži najviše 30 karaktera.

#### Primer 1

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite nisku: dan
|| dandandan
```

#### Primer 2

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite nisku: 3sesira
|| 3sesira3sesira3sesira
```

#### Primer 3

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite nisku: a-b=5
|| a-b=5a-b=5a-b=5
```

[Rešenje 3.5.11]

**Zadatak 3.5.12** Napisati program koji za unetu reč maksimalne dužine 20 karaktera i pozitivan broj  $n$  manji od 10, formira rezultujuću reč tako što unetu reč kopira  $n$  puta, pri čemu se između svaka dva kopiranja umeće crtica. U slučaju neispravnog unosa, ispisati odgovarajuću poruku o grešci.

#### Primer 1

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite nisku: ana
|| Unesite broj n: 4
|| ana-ana-ana-ana
```

#### Primer 2

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite nisku: 123
|| Unesite broj n: 1
|| 123
```

#### Primer 3

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite nisku: x*y
|| Unesite broj n: 3
|| x*y-x*y-x*y
```

[Rešenje 3.5.12]

**Zadatak 3.5.13** Napisati funkciju `void kopiraj_n(char t[], char s[], int n)` koja kopira najviše  $n$  karaktera niske  $s$  u nisku  $t$ . Napisati program koji testira rad napisane funkcije. Pretpostaviti da je maksimalna dužina niske  $s$  20 karaktera. U slučaju neispravnog unosa, ispisati odgovarajuću poruku o grešci.



*Primer 1*

```

INTERAKCIJA SA PROGRAMOM:
Unesite nisku: petar
Unesite broj n: 3
Rezultujuca niska: pet

```

*Primer 2*

```

INTERAKCIJA SA PROGRAMOM:
Unesite nisku: gromobran
Unesite broj n: 4
Rezultujuca niska: grom

```

*Primer 3*

```

INTERAKCIJA SA PROGRAMOM:
Unesite nisku: abc
Unesite broj n: 15
Rezultujuca niska: abc

```

[Rešenje 3.5.13]

**Zadatak 3.5.14** Napisati funkciju `void dupliranje(char t[], char s[])` koja na osnovu niske *s* formira nisku *t* tako što duplira svaki karakter niske *s*. Napisati program koji učitava nisku maksimalne dužine 20 karaktera i testira rad napisane funkcije.

*Primer 1*

```

INTERAKCIJA SA PROGRAMOM:
Unesite nisku: zima
zziimmaa

```

*Primer 2*

```

INTERAKCIJA SA PROGRAMOM:
Unesite nisku: A+B+C
AA++BB++CC

```

*Primer 3*

```

INTERAKCIJA SA PROGRAMOM:
Unesite nisku: C
CC

```

[Rešenje 3.5.14]

**Zadatak 3.5.15** Napisati program koji učitava nisku cifara sa eventualnim vodećim znakom i pretvara je u ceo broj.

*Primer 1*

```

INTERAKCIJA SA PROGRAMOM:
Unesite nisku:
-1238
-1238

```

*Primer 2*

```

INTERAKCIJA SA PROGRAMOM:
Unesite nisku:
73
73

```

*Primer 3*

```

INTERAKCIJA SA PROGRAMOM:
Unesite nisku:
+1
1

```

[Rešenje 3.5.15]

**Zadatak 3.5.16** Napisati program koji učitava ceo broj, pretvara ga u nisku i ispisuje dobijenu nisku.

*Primer 1*

```

INTERAKCIJA SA PROGRAMOM:
Unesite ceo broj:
-6543
-6543

```

*Primer 2*

```

INTERAKCIJA SA PROGRAMOM:
Unesite ceo broj:
84
84

```

*Primer 3*

```

INTERAKCIJA SA PROGRAMOM:
Unesite ceo broj:
5
5

```

[Rešenje 3.5.16]

**Zadatak 3.5.17** Napisati funkciju `int heksadekadni_broj(char s[])` koja proverava da li je niskom *s* zadat korektan heksadekadni broj. Funkcija treba da vrati vrednost 1 ukoliko je uslov ispunjen, odnosno 0 ako nije. Napisati program koji za učitane nisku maksimalne dužine 7 karaktera ispisuje da li je korektan heksadekadni broj. UPUTSTVO: *Heksadekadni broj je korektno zadat ako počinje prefiksom 0x ili 0X i ako sadrži samo cifre i mala ili velika slova A, B, C, D, E i F.*

*Primer 1*

```
|| INTERAKCIJA SA PROGRAMOM: ||  
|| Unesite nisku: 0x12EF      ||  
|| Korektan heksadekadni broj. ||
```

*Primer 2*

```
|| INTERAKCIJA SA PROGRAMOM: ||  
|| Unesite nisku: 0X22af     ||  
|| Korektan heksadekadni broj. ||
```

*Primer 3*

```
|| INTERAKCIJA SA PROGRAMOM: ||  
|| Unesite nisku: 0xErA9      ||  
|| Nekorektan heksadekadni broj. ||
```

[Rešenje 3.5.17]

**Zadatak 3.5.18** Napisati funkciju `int dekadna_vrednost(char s[])` koja izračunava dekadnu vrednost heksadekadnog broja zadatog niskom *s*. Napisati program koji za učitane nisku maksimalne dužine 7 karaktera ispisuje odgovarajuću dekadnu vrednost. Pretpostaviti da je uneta niska korektan heksadekadni broj.

*Primer 1*

```
|| INTERAKCIJA SA PROGRAMOM: ||  
|| Unesite nisku: 0x2A34      ||  
|| 10804                      ||
```

*Primer 2*

```
|| INTERAKCIJA SA PROGRAMOM: ||  
|| Unesite nisku: 0Xff2       ||  
|| 4082                       ||
```

*Primer 3*

```
|| INTERAKCIJA SA PROGRAMOM: ||  
|| Unesite nisku: 0xE1A9      ||  
|| 57769                      ||
```

[Rešenje 3.5.18]

**Zadatak 3.5.19** Napisati funkciju `int ucitaj_liniju(char s[], int n)` koja učitava liniju maksimalne dužine *n* u nisku *s* i vraća dužinu učitane linije. Napisati program koji učitava linije do EOF i ispisuje najdužu liniju i njenu dužinu. Ukoliko ima više linija maksimalne dužine, ispisati prvu. Pretpostaviti da svaka linija sadrži najviše 80 karaktera. NAPOMENA: *Linija može da sadrži blanko znakove, ali ne sadrži znak za novi red ili EOF.*

*Primer 1*

```

INTERAKCIJA SA PROGRAMOM:
Unesite linije:
Dobar dan!
Kako ste, sta ima novo?
Ja sam dobro.
Kako ste, sta ima novo?
23

```

*Primer 2*

```

INTERAKCIJA SA PROGRAMOM:
Unesite linije:
Prva linija
Druga linija
Trecia linija
Druga linija
12

```

*Primer 3*

```

INTERAKCIJA SA PROGRAMOM:
Unesite linije:
Danas je lep dan.
Danas je lep dan.
17

```

[Rešenje 3.5.19]

\* **Zadatak 3.5.20** Napisati funkcije za rad sa rečenicama:

- (a) `int procitaj_recenicu(char s[], int n)` koja učitava rečenicu i smešta je u nisku `s`. Funkcija vraća dužinu učitane rečenice. Učitavanje se završava nakon učitanoog karaktera `.`, nakon `n` učitanih karaktera ili ako se dodje do kraja ulaza.
- (b) `void prebroj(char s[], int *broj_malih, int *broj_velikih)` koja prebrojava mala i velika slova u niski `s`.

Napisati program koji učitava rečenice do kraja ulaza i ispisuje onu rečenicu kod koje je apsolutna razlika broja malih i velikih slova najveća. Pri učitavanju rečenica zanemariti sve beline koje se nalaze između dve rečenice.

*Primer 1*

```

INTERAKCIJA SA PROGRAMOM:
Unesite tekst:
U ovom poglavlju se govori o niskama. Niske su nizovi karaktera ciji je
poslednji element terminalna nula.
U ovom zadatku je potrebno ucitati recenice. Svaka recenica pocinje sa bilo
kojim karakterom koji nije belina. Na kraju recenice se nalazi tacka.
Niske su nizovi karaktera ciji je poslednji element terminalna nula.

```

*Primer 2*

```

INTERAKCIJA SA PROGRAMOM:
Unesite tekst:
bbbbAAA. AbAbAb. AbAbAbab.
AbAbAbab.

```

*Primer 3*

```

INTERAKCIJA SA PROGRAMOM:
Unesite tekst:
Nije uneta nijedna recenica.

```

*Primer 4*

```

INTERAKCIJA SA PROGRAMOM:
Unesite tekst:
AAAbbb. bbbAAA. AbAbAb.
AAAbbb

```

[Rešenje 3.5.20]

### 3 Predstavljanje podataka

---

**Zadatak 3.5.21** Napisati funkciju `char* strchr_klon(char s[], char c)` koja vraća pokazivač na prvo pojavljivanje karaktera `c` u niski `s` ili `NULL` ukoliko se karakter `c` ne pojavljuje u niski `s`. Napisati program koji za učitane nisku maksimalne dužine 20 karaktera i karakter `c` ispisuje indeks prvog pojavljivanja karaktera `c` u okviru učitane niske ili `-1` ukoliko učitana niska ne sadrži uneti karakter.

#### Primer 1

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite nisku s: programiranje
|| Unesite karakter c: a
|| Pozicija: 5
```

#### Primer 2

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite nisku s: 123456789
|| Unesite karakter c: y
|| Pozicija: -1
```

#### Primer 3

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite nisku s: leto2017
|| Unesite karakter c: o
|| Pozicija: 5
```

#### Primer 4

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite nisku s: jedrilica
|| Unesite karakter c: i
|| Pozicija: -1
```

[Rešenje 3.5.21]

**Zadatak 3.5.22** Napisati funkciju `int strspn_klon(char t[], char s[])` koja izračunava dužinu prefiksa niske `t` sastavljenog od karaktera niske `s`. Napisati program koji za učitane dve niske maksimalne dužine 20 karaktera ispisuje rezultat poziva napisane funkcije.

#### Primer 1

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite nisku t: program
|| Unesite nisku s: pero
|| 3
```

#### Primer 2

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite nisku t: Barcelona
|| Unesite nisku s: Brazil
|| 3
```

#### Primer 3

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite nisku t: 24.10.2017.
|| Unesite nisku s: 0123456789
|| 2
```

#### Primer 4

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite nisku t: 12345
|| Unesite nisku s: 9876543210
|| 5
```

[Rešenje 3.5.22]

**Zadatak 3.5.23** Napisati funkciju `int strcspn_klon(char t[], char s[])` koja izračunava dužinu prefiksa niske `t` sastavljenog isključivo od karaktera

koji se ne nalaze u niski *s*. Napisati program koji testira ovu funkciju za dve unete niske maksimalne dužine 100 karaktera.

*Primer 1*

```
INTERAKCIJA SA PROGRAMOM:
Unesite nisku t:
programiranje
Unesite nisku s:
pero
0
```

*Primer 2*

```
INTERAKCIJA SA PROGRAMOM:
Unesite nisku t:
programiranje
Unesite nisku s:
analiza
5
```

*Primer 3*

```
INTERAKCIJA SA PROGRAMOM:
Unesite nisku t:
programiranje
Unesite nisku s:
1.10.
13
```

[Rešenje 3.5.23]

**Zadatak 3.5.24** Napisati funkciju `char* strstr_klon(char s[], char t[])` koja vraća pokazivač na prvo pojavljivanje niske *t* u niski *s* ili *NULL* ukoliko se niska *t* ne pojavljuje u niski *s*. Napisati program koji testira napisanu funkciju tako što učitava pet linija i ispisuje sve redne brojeve linija koje sadrže nisku *program*. Ukoliko ne postoji linija sa niskom *program*, ispisati odgovarajuću poruku. Pretpostaviti da je svaka linija maksimalne dužine 100 karaktera kao i da se linije numerišu od broja 1.

*Primer 1*

```
INTERAKCIJA SA PROGRAMOM:
Unesite pet linija:
tu program
c prog. jezik
c++ programskih jezik
Programski odbor
<b>program</b>
1 3 5
```

*Primer 2*

```
INTERAKCIJA SA PROGRAMOM:
Unesite pet linija:
Programske paradigme
su predmet na
trecoj godini
programerskih
smerova.
4
```

*Primer 3*

```
INTERAKCIJA SA PROGRAMOM:
Unesite pet linija:
U narednim
linijama
necemo navoditi
nisku koja se
trazi.
Nijedna linija ne sadrzi
nisku program.
```

[Rešenje 3.5.24]

**Zadatak 3.5.25** Napisati funkciju `int strcmp_klon(char s[], char t[])` koja vraća 0 ako su niske *s* i *t* jednake, neku pozitivnu vrednost ako je *s* leksikografski iza *t*, a neku negativnu vrednost inače. Napisati program koji učitava dve niske maksimalne dužine 20 karaktera i ako su različite, ispisuje učitane niske u rastućem leksikografskom poretku, a ako su jednake, ispisuje samo jednu nisku.

### 3 Predstavljanje podataka

---

#### Primer 1

```
INTERAKCIJA SA PROGRAMOM:
Unesite nisku s:
Beograd
Unesite nisku t:
Amsterdam
Amsterdam
Beograd
```

#### Primer 2

```
INTERAKCIJA SA PROGRAMOM:
Unesite nisku s:
Beograd
Unesite nisku t:
Beograd
Beograd
```

#### Primer 3

```
INTERAKCIJA SA PROGRAMOM:
Unesite nisku s:
beogradjanin
Unesite nisku t:
beogradjanka
beogradjanin
beogradjanka
```

[Rešenje 3.5.25]

**Zadatak 3.5.26** Napisati funkciju `void obrni(char s[])` koja obrće nisku `s`. Napisati program koji obrće učitane niske maksimalne dužine 20 karaktera i ispisuje obrnutu nisku.

#### Primer 1

```
INTERAKCIJA SA PROGRAMOM:
kisobran
narbosik
```

#### Primer 2

```
INTERAKCIJA SA PROGRAMOM:
Aleksandar
radnaskelA
```

#### Primer 3

```
INTERAKCIJA SA PROGRAMOM:
kajak
kajak
```

[Rešenje 3.5.26]

**Zadatak 3.5.27** Napisati funkciju `void rotiraj(char s[], int k)` koja rotira nisku `s` za `k` mesta ulevo. Napisati program koji rotira učitane niske maksimalne dužine 20 karaktera i ispisuje rotiranu nisku.

#### Primer 1

```
INTERAKCIJA SA PROGRAMOM:
sveska
2
eskasv
```

#### Primer 2

```
INTERAKCIJA SA PROGRAMOM:
olovka
6
olovka
```

#### Primer 3

```
INTERAKCIJA SA PROGRAMOM:
rezac
8
acrez
```

[Rešenje 3.5.27]

**Zadatak 3.5.28** Napisati program koji šifrira unetu nisku tako što svako slovo zamenjuje sledećim slovom abecede (slova 'z' i 'Z' zamenjuje redom sa 'a' i 'A'), a ostale karaktere ostavlja nepromenjene. Pretpostaviti da uneta niska nije duža od 20 karaktera.

*Primer 1*

```

INTERAKCIJA SA PROGRAMOM:
Unesite nisku:
bundeua
cvoefwb

```

*Primer 2*

```

INTERAKCIJA SA PROGRAMOM:
Unesite nisku:
zimzelen
ajnafmfo

```

*Primer 3*

```

INTERAKCIJA SA PROGRAMOM:
Unesite nisku:
Oktobar17
Plupcbs17

```

[Rešenje 3.5.28]

**Zadatak 3.5.29** Napisati funkciju `void sifruj(char rec[], char sifra[])` koja na osnovu date reči formira šifru tako što se svako slovo u reči zameni sa naredna tri slova u abecedi. Napisati program koji testira napisanu funkciju za reč maksimalne dužine 20 karaktera.

*Primer 1*

```

INTERAKCIJA SA PROGRAMOM:
Unesite nisku:
tamo
uvwbcdnopppqr

```

*Primer 2*

```

INTERAKCIJA SA PROGRAMOM:
Unesite nisku:
Zec
ABCfghdef

```

*Primer 3*

```

INTERAKCIJA SA PROGRAMOM:
Unesite nisku:
a+b=c
bcd+cde=def

```

[Rešenje 3.5.29]

**Zadatak 3.5.30** Napisati funkciju `void formiraj(char s1[], char s2[], char c1, char c2)` koja na osnovu niske  $s_1$  formira nisku  $s_2$  udvajanjem svih karaktera  $c_1$  u niski  $s_1$  i izbacivanjem svih karaktera  $c_2$  iz niske  $s_1$ , dok ostali karakteri ostaju nepromenjeni. Napisati program koji testira ovu funkciju za unetu nisku i dva uneta karaktera. Pretpostaviti da uneta niska nije duža od 20 karaktera.

*Primer 1*

```

INTERAKCIJA SA PROGRAMOM:
Unesite nisku:
flomaster
Unesite prvi karakter:
m
Unesite drugi karakter:
s
flosaster

```

*Primer 2*

```

INTERAKCIJA SA PROGRAMOM:
Unesite nisku:
bojica
Unesite prvi karakter:
b
Unesite drugi karakter:
a
bbojic

```

*Primer 3*

```

INTERAKCIJA SA PROGRAMOM:
Unesite nisku:
patentara
Unesite prvi karakter:
t
Unesite drugi karakter:
a
pttenttr

```

[Rešenje 3.5.30]

### 3 Predstavljanje podataka

\* **Zadatak 3.5.31** Napisati program za rad sa brojevima zapisanim u različitim brojevnim sistemima.

- (a) Napisati funkciju `unsigned int u_dekadni_sistem(char broj[], unsigned int osnova)` koja određuje dekadnu vrednost zapisa datog neoznačenog broja *broj* u datoj osnovi.
- (b) Napisati funkciju `void iz_dekadnog_sistema(unsigned int broj, unsigned int osnova, char rezultat[])` koja datu dekadnu vrednost *broj* zapisuje u datoj osnovi *osnovi* i smešta rezultat u nisku *rezultat*. Pretpostaviti da je  $0 < b \leq 16$ .

Napisati program koji učitava broj  $n$  i osnove  $o_1$  i  $o_2$  i ispisuje dekadnu vrednost broja  $n$  u osnovi  $o_1$ , kao i vrednost koja se dobije kada se ta dekadna vrednost zapise u osnovi  $o_2$ . Pretpostaviti da je ulaz ispravan i da će svi brojevi biti u opsegu tipa `unsigned`.

#### Primer 1

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite n, o1 i o2: 10101011 2 16
|| Dekadna vrednost broja 10101011: 171
|| Vrednost broja 171 u osnovi 16: AB
```

#### Primer 2

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite n, o1 i o2: 1067 8 3
|| Dekadna vrednost broja 1067: 567
|| Vrednost broja 567 u osnovi 3: 210000
```

#### Primer 3

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite n, o1 i o2: 1010111001010 2 3
|| Dekadna vrednost broja 1010111001010: 5578
|| Vrednost broja 5578 u osnovi 3: 21122121
```

#### Primer 4

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite n, o1 i o2: 111 3 5
|| Dekadna vrednost broja 111: 13
|| Vrednost broja 13 u osnovi 5: 23
```

[Rešenje 3.5.31]

## 3.6 Rešenja

### Rešenje 3.5.1

```
1 #include <stdio.h>
2 #include <stdlib.h>
3 #include <ctype.h>
4
5 #define MAKS_NISKA 11
6
7 /* Funkcija koja svako malo slovo niske pretvara u odgovarajuće
   veliko i obrnuto. Ostali karakteri ostaju nepromenjeni. */
```



```

9 void konvertuj(char s[])
{
11     int i;

13     /* Prolazi se kroz nisku, karakter po karakter, sve dok se ne
        dodje do terminalne nule koja služi kao oznaka da se doslo
15     do kraja niske. */
    for (i = 0; s[i] != '\0'; i++)
17     {
        /* Svako malo slovo se pretvara u veliko i obrnuto. */
19         if (islower(s[i]))
            s[i] = toupper(s[i]);
21         else if (isupper(s[i]))
            s[i] = tolower(s[i]);
23     }
25 }

27 int main()
{
    /* Poslednji karakter svake niske je terminirajuća nula '\0',
29     specijalni karakter čiji je ASCII kod 0.

    Ukoliko je pretpostavka da niska sadrži najviše 10 karaktera,
31     neophodno je deklarirati niz od 11 karaktera, pri čemu se
    dodatni izdvaja za terminirajuću nulu. */
33     char s[MAKS_NISKA];
35     printf("Unesite nisku: ");

37     /* Za razliku od nizova koji se učitavaju i štampaju element po
        element, niske se mogu učitati i odštampati pomoću jedne
39     scanf/printf naredbe korišćenjem specifikatora %s. */
    scanf("%s", s);
41
    /* Izmena niske. */
43     konvertuj(s);

45     /* Ispis rezultata. */
    printf("%s\n", s);
47
    exit(EXIT_SUCCESS);
49 }

```

### Rešenje 3.5.2

```

1  #include <stdio.h>
   #include <stdlib.h>
3
   #define MAKS_NISKA 31
5
   /* Funkcija koja ubacuje zvezdice na svako drugo mesto
7   niske s. */

```

### 3 Predstavljanje podataka

---

```
void ubaci_zvezdice(char s[]){
9   int i;

11  for(i=0; s[i] != '\0' && s[i+1] != '\0'; i+=2)
    s[i+1] = '*';

13

15  /* Uslov u petlji moze krace da se zapise na sledeci nacin:
    for(i=0; s[i] && s[i+1]; i+=2)
        s[i+1] = '*';
17  */

19  /* III nacin:
    for(i=0; s[i]; i++)
21      if(i%2 == 1)
        s[i] = '*';
23  */
}

25
27 int main()
28 {
29     /* Deklaracija potrebnih promenljivih. */
    char s[MAKS_NISKA];

31     /* Ucitavanje niske. */
    printf("Unesite nisku: ");
33     scanf("%s", s);

35     /* Izmena niske. */
    ubaci_zvezdice(s);

37

39     /* Ispis rezultata. */
    printf("Izmenjena niska: %s\n", s);

41     exit(EXIT_SUCCESS);
}
```

#### Rešenje 3.5.3

```
1  #include <stdio.h>
   #include <stdlib.h>
3  #include <ctype.h>

5  #define MAKS_NISKA 21

7  /* Funkcija koja sva slova niske s pretvara u velika slova. */
void u_velika_slova(char s[]){
9   int i;

11  for(i=0; s[i]; i++)
    s[i] = toupper(s[i]);
13 }
```

```

15 /* Funkcija koja vraca 1 ako su niske s1 i s2 jednake, a nulu
    inace. */
17 int poredjenje(char s1[], char s2[]){
    int i;
19
    /* Prolazi se kroz obe niske dok god ima neobradjenih karaktera
    u bilo kojoj od njih. Ukoliko se naidje na karaktere koji
    su razliciti, kao povratna vrednost se vraca 0 jer u tom
    slucaju niske nisu jednake. */
23 for(i=0; s1[i] || s2[i]; i++)
25     if(s1[i] != s2[i])
        return 0;
27
    /* Ako se doslo do kraja petlje znaci da su se svi karakteri
    poklopili, a da se pri tom doslo do kraja obe niske, tako da
    se kao povratna vrednost funkcije vraca 1 jer su niske
    s1 i s2 jednake. */
31     return 1;
33 }

35 int main(){
    /* Deklaracija potrebnih promenljivih. */
37     char s1[MAKS_NISKA], s2[MAKS_NISKA];

39     /* Ucitavaju se niske s1 i s2. */
    printf("Unesite niske:\n");
41     scanf("%s%s", s1, s2);

43     /* Kako bi se pri poredjenju zanemarila razlika izmedju malih i
    velikih slova, sva slova obe niske se pretvaraju u velika. */
45     u_velika_slova(s1);
    u_velika_slova(s2);
47

    /* Ispis rezultata. */
49     if(poredjenje(s1, s2))
        printf("Niske su jednake.\n");
51     else
        printf("Niske nisu jednake.\n");
53
    exit(EXIT_FAILURE);
55 }

```

### Rešenje 3.5.4

```

1 #include <stdio.h>
  #include <stdlib.h>
3 #include <ctype.h>
  #include <string.h>
5
  #define MAKS_NISKA 21

```

```
7  /* Funkcija koja proverava da li je karakter c samoglasnik. */
9  int samoglasnik(char c)
10 {
11     /* Karakter se pretvara u veliko slovo kako bi se izbegle posebne
12        provere za mala i velika slova. */
13     c = toupper(c);
14
15     /* Samoglasnici su slova a, e, i, o i u */
16     if (c == 'A' || c == 'E' || c == 'I' || c == 'O' || c == 'U')
17         return 1;
18
19     return 0;
20 }
21
22 /* Funkcija koja proverava da li se niska s završava samoglasnikom.
23    */
24 int samoglasnik_na_kraju(char s[])
25 {
26     /* Funkcija strlen racuna duzinu niske. Njena deklaracija se
27        nalazi u zaglavlju string.h. */
28     int duzina = strlen(s);
29
30     /* Ako je niska prazna, ne završava se samoglasnikom. */
31     if (duzina == 0)
32         return 0;
33
34     /* Proverava se da li je poslednji karakter niske samoglasnik. */
35     return samoglasnik(s[duzina - 1]);
36 }
37
38 int main()
39 {
40     /* Deklaracija niske. */
41     char s[MAKS_NISKA];
42
43     /* Ucitava se niska. */
44     printf("Unesite nisku: ");
45     scanf("%s", s);
46
47     /* Ispis rezultata. */
48     if (samoglasnik_na_kraju(s))
49         printf("Niska se završava samoglasnikom.\n");
50     else
51         printf("Niska se ne završava samoglasnikom.\n");
52
53     exit(EXIT_SUCCESS);
54 }
```

#### Rešenje 3.5.5

```
1  #include <stdio.h>
2  #include <stdlib.h>
3  #include <ctype.h>
4
5  #define MAKS_NISKA 21
6
7  /* Funkcija koja proverava da li niska s sadrzi bas jedno
8     veliko slovo. */
9  int sadrzi_veliko(char s[])
10 {
11     int i;
12
13     for(i=0; s[i]; i++)
14         if(isupper(s[i]))
15             return 1;
16
17     return 0;
18 }
19
20 int main()
21 {
22     /* Deklaracija niske. */
23     char s[MAKS_NISKA];
24
25     /* Ucitava se niska. */
26     printf("Unesite nisku:");
27     scanf("%s", s);
28
29     /* Ispis rezultata. */
30     if(sadrzi_veliko(s))
31         printf("Sadrzi veliko slovo.\n");
32     else
33         printf("Ne sadrzi veliko slovo.\n");
34
35     exit(EXIT_SUCCESS);
36 }
```

### Rešenje 3.5.6

```
1  #include <stdio.h>
2  #include <stdlib.h>
3  #include <string.h>
4
5  #define MAKS_NISKA 21
6
7  /* Funkcija koja vraca indeks prvog pojavljivanja karaktera c u
8     okviru niske s. Ukoliko se ne pojavljuje, funkcija vraca -1. */
9  int pozicija(char s[], char c)
10 {
11     int i;
```

```
12     for(i=0; s[i]; i++)
14         if(s[i] == c)
15             return i;
16
17     return -1;
18 }
19
20 int main()
21 {
22     /* Deklaracija potrebnih promenljivih. */
23     char s[MAKS_NISKA];
24     char c;
25
26     /* Ucitavaju se niska i karakter. */
27     printf("Unesite nisku:");
28     scanf("%s", s);
29     getchar();
30     printf("Unesite karakter:");
31     c = getchar();
32
33     /* I nacin: */
34     printf("%d\n", pozicija(s,c));
35
36     /* II nacin:
37      Funkcija strchr(s,c) je funkcija koja vraca adresu prvog
38      pojavljivanja karaktera c u niski s, ako se c pojavljuje u s,
39      a NULL inace.
40
41      Vrednost promenljive s je zapravo vrednost adrese prvog
42      karaktera niske s.
43
44      Ako treba da se ispise indeks prvog pojavljivanja, to moze
45      da se uradi tako sto se od adrese koji je vratila funkcija
46      strchr oduzme adresa prvog karaktera.
47
48      Na primer:
49      s = "koliba"      ==> s je adresa karaktera 'k'
50      p = strchr(s, 'l') ==> p je adresa karaktera 'l'
51      |k|o|l|i|b|a|
52      ^   ^
53      |   |
54      s   p
55      Izraz p-s ima vrednost 2 (jer je rastojanje izmedju
56      ove dve adrese 2).
57
58      Tip promenljive p je char* jer predstavlja adresu
59      jednog karaktera.
60
61      char *p = strchr(s, c);
62      if (p != NULL)
63          printf("%d\n", p - s);
```

```

64     else
65         printf("-1\n");
66     */
67
68     exit(EXIT_SUCCESS);
69 }

```

### Rešenje 3.5.7

```

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

#define MAKS_NISKA 11

/* Funkcija koja proverava da li je niska t podniska niske s. */
int podniska(char s[], char t[])
{
    int i, j;

    /* Spoljasnja petlja ide redom po niski s. */
    for (i = 0; s[i] != '\0'; i++) {

        /* Unutrasnja petlja ide redom po niski t pomocu brojaca j
           i proverava da li se cela niska t poklapa sa delom niske
           s koji pocinje na poziciji i.

           Cim se naidje na situaciju da se karakteri ne poklapaju,
           izlazi se iz unutrasnje petlje. */
        for (j = 0; t[j] != '\0'; j++)
            if (s[i+j] != t[j])
                break;

        /* Ako je unutrasnja petlja dosla do kraja niske t, to
           znaci da su se svi karakteri iz t poklopili sa karakterima
           iz s i t je podniska od s. */
        if (t[j] == '\0')
            return 1;
    }

    return 0;
}

int main()
{
    /* Deklaracija niski s i t. */
    char s[MAKS_NISKA], t[MAKS_NISKA];

    /* Ucitavaju se niske s i t. */
    printf("Unesite nisku s: ");
    scanf("%s", s);

```

### 3 Predstavljanje podataka

```
printf("Unesite nisku t: ");
scanf("%s", t);

/* Ispis rezultata. */
if (podniska(s, t))
    printf("t je podniska niske s.\n");
else
    printf("t nije podniska niske s.\n");

/* II nacin:
   Funkcija strstr(t, s) proverava da li je t podniska od s
   i kao povratnu vrednost vraća adresu prvog pojavljivanja t u s
   ili NULL ukoliko se t ne pojavljuje u s.
   Deklaracija ove funkcije se nalazi u zaglavlju string.h

   char* p = strstr(t, s);
   if(p == NULL)
       printf("t nije podniska od s.\n");
   else
       printf("t je podniska od s.\n");
*/

exit(EXIT_SUCCESS);
}
```

#### Rešenje 3.5.8

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <ctype.h>

#define MAKS_LINIJA 101

/* Funkcija koja učitava liniju maksimalne dužine n i upisuje
   je u s. Funkcija ne smesta znak za novi red na kraj linije. */
void ucitaj_liniju(char s[], int n)
{
    int i = 0;
    int c;

    /* Učitava se karakter po karakter dok se ne unese novi red
       ili oznaka za kraj ulaza ili dok se ne dostigne maksimalan
       broj karaktera. */
    while ((c = getchar()) != '\n' && i < n-1 && c != EOF) {
        s[i] = c;
        i++;
    }

    /* Maksimalan broj karaktera za liniju je n-1 jer na kraju
       treba ostaviti i jedno mesto za terminalnu nulu. */
}
```



```

    s[i] = '\0';
26 }

28 /* Funkcija koja uklanja beline sa kraja niske s. */
void skрати(char s[])
30 {
    int i;
32     /* Vrsi se prolazak kroz nisku sa desna na levo i trazi se pozicija
        prvog karaktera koji nije belina.

34     Funkcija isspace proverava da li je dati karakter neka od belina
        (blanko, tab ili novi red) i njena deklaracija se nalazi u
        zaglavlju ctype.h. */
36     for (i = strlen(s) - 1; i >= 0; i--)
        if (!isspace(s[i]))
40         break;

42     /* Nakon izlaska iz petlje, brojac i se nalazi na poziciji prvog
        karaktera sa desne strane koji nije belina. Iz tog razloga se
        na poziciju i+1 upisuje terminalna nula kao oznaka da se sada
        tu nalazi kraj niske. */
44     s[i + 1] = '\0';
46 }

48 int main()
50 {
    /* Deklaracija niske. */
52     char s[MAKS_LINIJA];

54     /* Ucitava se cela linija sa ulaza. */
    printf("Unesite nisku:\n");
56     ucitaj_liniju(s, MAKS_LINIJA);

58     /* Ispis rezultata. */
    printf("Ucitana niska: %s\n", s);
60     skрати(s);
    printf("Izmenjena niska: %s\n", s);
62     exit(EXIT_SUCCESS);
64 }

```

### Rešenje 3.5.9

```

#include <stdio.h>
2 #include <stdlib.h>
#include <string.h>
4 #include <ctype.h>

6 #define MAKS_LINIJA 21

8 /* Funkcija koja iz niske s uklanja sva slova. */

```

```
void ukloni_slova(char s[])
{
    int i, j;

    /* Prolazi se kroz nisku s karakter po karakter i vrsi se provera
    da li trenutni karakter treba da se zadrzi. Karakter treba da
    se zadrzi ukoliko nije ni malo ni veliko slovo.

    Brojac j služi da pamti gde se upisuje sledeći karakter koji
    treba da se zadrži i svaki put kada se nađe na takav karakter,
    on se upisuje na poziciju j, a brojac j se uvecava. */
    for(i=0, j=0; s[i]; i++){
        if(!islower(s[i]) && !isupper(s[i]))
        {
            s[j] = s[i];
            j++;
        }
    }

    /* Na kraju se na poziciji j upisuje i terminalna nula, kako bi se
    naznacilo da se kraj niske nalazi nakon poslednjeg zadržanog
    karaktera. */
    s[j] = '\0';
}

int main()
{
    /* Deklaracija niske. */
    char s[MAKS_LINIJA];

    /* Ucitava se niska s. */
    printf("Unesite nisku:\n");
    scanf("%s", s);

    /* Ispis rezultata. */
    ukloni_slova(s);
    printf("%s\n", s);

    exit(EXIT_SUCCESS);
}
```

#### Rešenje 3.5.10

```
1 #include <stdio.h>
2 #include <stdlib.h>
3 #include <string.h>
4 #include <ctype.h>
5
6 #define MAKS_LINIJA 101
7
8 /* Funkcija koja ucitava liniju maksimalne duzine n i upisuje
```

```

9      je u s. Funkcija ne smesta znak za novi red na kraj linije. */
void ucitaj_liniju(char s[], int n)
11 {
13     int i = 0, c;

15     while ((c = getchar()) != '\n' && i < n-1 && c != EOF) {
17         s[i] = c;
19         i++;
21     }
22     s[i] = '\0';
23 }

24 /* Pomocna funkcija koja proverava da li karakter c1 treba zadržati
25    ako vazi da se iza njega nalazi c2. */
26 int treba_zadržati(char c1, char c2){
28     /* Ako neki od karaktera nije slovo, c1 se ne izbacuje. */
29     if(!isalpha(c1) || !isalpha(c2))
30         return 1;

31     /* Oba karaktera se pretvaraju u veliko slovo kako bi se smanjio
32        broj poredjenja. */
33     c1 = toupper(c1);
34     c2 = toupper(c2);

35     /* c1 se zadržava ako se c2 ne nalazi iza njega u abecedi. */
36     return c2 <= c1;
37 }

38 /* Funkcija koja uklanja sva slova za koja vazi da se neposredno
39    nakon njih nalazi slovo koje je u abecedi iza njih. */
40 void ukloni(char s[])
41 {
42     int i, j;
43     for(i=0, j=0; s[i]; i++){
44         if(traba_zadržati(s[i], s[i+1]))
45             {
46                 s[j] = s[i];
47                 j++;
48             }
49     }
50     s[j] = '\0';
51 }

52 int main()
53 {
54     /* Deklaracija niske. */
55     char s[MAKS_LINIJA];

56     /* Ucitava se cela linija sa ulaza. */
57     printf("Unesite nisku:\n");
58     ucitaj_liniju(s, MAKS_LINIJA);

```

```
61      /* Ispis rezultata. */
63      ukloni(s);
        printf("%s\n", s);
65
        exit(EXIT_SUCCESS);
67    }
```

#### Rešenje 3.5.11

```
1  #include <stdio.h>
2  #include <stdlib.h>
3  #include <string.h>
4
5  #define MAKS_NISKA 31
6  #define MAKS_REZULTAT 91
7
8  /* Niske se ne kopiraju naredbom dodele. Ukoliko je potrebno da neka
9   niska ima isti sadrzaj kao i neka druga niska, moze se koristiti
10  funkcija strcpy(t, s) koja kopira karaktere niske s u nisku t
11  zajedno za terminirajucom nulom. Deklaracija ove funkcije se
12  nalazi u zaglavlju string.h.
13
14  Funkcija strcpy_klon predstavlja jednu implementaciju funkcije
15  strcpy. */
16  void strcpy_klon(char kopija[], char original[])
17  {
18      int i;
19      for (i = 0; original[i]; i++)
20          kopija[i] = original[i];
21
22      kopija[i] = '\0';
23  }
24
25  int main()
26  {
27      /* Deklaracija niski. */
28      char s[MAKS_NISKA];
29      char t[MAKS_REZULTAT];
30
31      /* Ucitava se niska. */
32      printf("Unesite nisku: ");
33      scanf("%s", s);
34
35      /* Niska s se kopira u nisku t. */
36      strcpy_klon(t, s);
37
38      /* Funkcija strcat(s,t) nadovezuje karaktere niske s na kraj
39       niske t i novu nisku terminira karakterom '\0'. Deklaracija
40       ove funkcije se nalazi u zaglavlju string.h. */
41  }
```

```

43  /* Niska s se jos dva puta nadovezuje na nisku t. */
    strcat(t, s);
    strcat(t, s);
45
    /* Ispis rezultata. */
47    printf("%s\n", t);
49    exit(EXIT_SUCCESS);
}

```

### Rešenje 3.5.12

```

#include <stdio.h>
2  #include <stdlib.h>
#include <string.h>
4
#define MAKS_NISKA 21
6  #define MAKS_N 10
#define MAKS_REZULTAT (MAKS_NISKA*MAKS_N + 1)
8
int main()
10 {
    /* Deklaracija niski. */
12    char s[MAKS_NISKA];
    char t[MAKS_REZULTAT];
14    int i, n;

16    /* Ucitava se niska. */
    printf("Unesite nisku: ");
18    scanf("%s", s);

20    /* Ucitava se broj ponavljanja i vrsi se provera ispravnosti
        ulaza. */
22    printf("Unesite broj n: ");
    scanf("%d", &n);
24    if(n <= 0 || n > MAKS_N)
    {
26        printf("Greska: neispravan unos.\n");
        exit(EXIT_FAILURE);
28    }

30    /* Formira se rezultat. Prvi karakter rezultujuce niske se
        postavlja na terminalnu nulu. Ovo se radi jer strcat
32    funkcioniše tako sto krene od pocetka niske, ide do
        terminalne nule i zatim pocevši od tog mesta nadovezuje
34    nisku koja je prosledjena kao drugi argument. Na ovaj nacin
        je obezbedjeno da ce prvi poziv funkcije strcat krenuti da
36    nadovezuje od pocetka niske t.
        U petlji se na t nadovezuje prvo niska s, a zatim niska "-".
38    Ovo se ponavlja n-1 puta jer nakon poslednjeg nadovezivanja
        niske s ne treba da se nadje "-". Iz tog razloga se po

```

### 3 Predstavljanje podataka

```
40     završetku petlje vrši još jedno nadovezivanje niske s, ali ne
      i niske "-". */
42     t[0] = '\0';
      for(i=0; i<n-1; i++) {
44         strcat(t, s);
         strcat(t, "-");
46     }
      strcat(t, s);
48
      /* Ispis rezultata. */
50     printf("%s\n", t);
52     exit(EXIT_SUCCESS);
}
```

#### Rešenje 3.5.13

```
#include <stdio.h>
2 #include <stdlib.h>
#include <string.h>
4
#define MAKS_NISKA 21
6
void kopiraj_n(char t[], char s[], int n)
8 {
    int i;
10     /* Kopiranje se vrši ili dok se ne dodje do terminalne nule u s
       ili dok se ne prekopira n karaktera. */
12     for (i = 0; i < n && s[i] != '\0'; i++) {
        t[i] = s[i];
14     }

16     /* Na kraju rezultujuće niske se upisuje terminalna nula. */
    t[i] = '\0';
18 }

20 int main()
{
22     /* Deklaracije potrebnih promenljivih. */
    int n;
24     char s[MAKS_NISKA], t[MAKS_NISKA];

26     /* Učitava se niska. */
    printf("Unesite nisku: ");
28     scanf("%s", s);

30     /* Učitava se broj n i vrši se provera ispravnosti
       ulaza. */
32     printf("Unesite broj n: ");
    scanf("%d", &n);
34     if (n < 0 || n > MAKS_NISKA-1) {
```

```

36     printf("Greska: neispravan unos.\n");
    exit(EXIT_FAILURE);
}

38     /* Formira se rezultat. */
40     kopiraj_n(t, s, n);

42     /* II nacin:
    Koriscenjem funkcije strncpy(t, s, n), cija se deklaracija
44     nalazi u zaglavlju string.h, kopira najvise n karaktera niske
    s u nisku t.

46     strncpy(t,s,n);
48     */

50     /* Ispis rezultata. */
    printf("%s\n", t);
52
    exit(EXIT_SUCCESS);
54 }

```

### Rešenje 3.5.14

```

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

3
    /* Duzina niske koja se ucitava, bez terminalne nule. */
5  #define MAKS_DUZINA 20

7  /* Duzine originalne i rezultujuce niske. */
    #define MAKS_NISKA (MAKS_DUZINA + 1)
9    #define MAKS_REZULTAT (2 * MAKS_DUZINA + 1)

11 /* Funkcija koja od niske s formira nisku t dupliranjem svakog
    karaktera. Npr. abc postaje aabbcc. */
13 void dupliranje(char t[], char s[])
    {
15     int i, j;

17     /* Brojac i oznacava tekucu poziciju u niski s, a
    brojac j oznacava tekucu poziciju u niski t. */
19     for (i = 0, j = 0; s[i] != '\0'; i++, j += 2) {
        t[j] = s[i];
21         t[j + 1] = s[i];
    }

23
    /* Upisuje se terminalna nula na kraj rezultujuce niske. */
25     t[j] = '\0';
}

27 int main()

```

### 3 Predstavljanje podataka

---

```
29 {
30     /* Deklaracija niski. */
31     char s[MAKS_NISKA], t[MAKS_REZULTAT];
32
33     /* Ucitava se niska. */
34     printf("Unesite nisku: ");
35     scanf("%s", s);
36
37     /* Poziva se funkcija za dupliranje. */
38     dupliranje(t, s);
39
40     /* Ispis rezultata. */
41     printf("%s\n", t);
42
43     exit(EXIT_SUCCESS);
44 }
```

#### Rešenje 3.5.15

```
#include <stdio.h>
#include <stdlib.h>
#include <ctype.h>

#define MAKS_NISKA 10

/* Funkcija formiraj_broj na osnovu niske b formira ceo broj
   ciji je to zapis.

   Ako su cifre broja a, b, c i d, tada broj mozemo kreirati kao:
   a*10^3 + b*10^2 + c*10^1 + d*10^0.
   Medjutim, efikasnije je koristiti Hornerovu semu:
   10*(10*(10*(10*0 + a)+b)+c)+d. */
int formiraj_broj(char b[])
{
    int i;
    int broj = 0, znak;

    /* Odredjivanje znaka broja i pozicije prve cifre. */
    if(b[0] == '-'){
        znak = -1;
        i = 1;
    }else if(b[0] == '+'){
        znak = 1;
        i = 1;
    }else{
        i = 0;
        znak = 1;
    }

    /* Prolazak kroz cifre broja i racunanje vrednosti broja
       koriscenjem Hornerove seme. Vrednost trenutne cifre se
```



```

34     dobija kada se od trenutnog karaktera (b[i]) oduzme
    karakter '0'.
36     Ako se naidje na karakter koji nije cifra, petlja se
    prekida. Na primer za b="123abc", rezultat treba da
    bude 123. */
38     for (; b[i] != '\0'; i++)
    {
40         if(isdigit(b[i]))
            broj = broj * 10 + (b[i] - '0');
42         else
            break;
44     }

46     return broj*znak;
    }
48
49 int main()
50 {
51     /* Deklaracija niske. */
52     char s[MAKS_NISKA];

53
54     /* Broj se ucitava kao niska. */
55     scanf("%s", s);

56
57     /* Ispis rezultata. */
58     printf("%d\n", formiraj_broj(s));

59
60     /* II nacin:
        Koriscenjem funkcije atoi. Deklaracija ove funkcije se nalazi
        u zaglavlju stdlib.h.
61
62     printf("%d\n", atoi(s)); */
63
64     exit(EXIT_SUCCESS);
65 }

```

### Rešenje 3.5.16

```

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

4 #define MAKS_NISKA 10

6 /* Funkcija koja racuna broj cifara broja n. */
7 int broj_cifara(int n)
8 {
9     int i = 0;
10    do{
11        i++;
12        n/=10;
    }while(n);

```

```
14     return i;
15 }
16
17 /* Funkcija koja od prosledjenog broja formira nisku. */
18 void broj_u_nisku(int broj, char s[])
19 {
20     int n, cifra, i;
21
22     /* Promenljiva n cuva informaciju o duzini niske. Duzina niske
23      odgovara broju cifara prosledjenog broja. Ukoliko je broj
24      negativan, onda se duzina uvecava za 1 i na prvo mesto se
25      upisuje znak '-'. */
26     n = broj_cifara(broj);
27     if(broj < 0)
28     {
29         s[0] = '-';
30         n++;
31     }
32
33     /* U nastavku se radi sa apsolutnom vrednoscu broja. */
34     broj = abs(broj);
35
36     /* Cifre broja se upisuju u nisku s sa desna na levo. */
37     s[n] = '\0';
38     i = n-1;
39     do{
40         /* Karakter koji odgovara trenutnoj cifri se dobija izrazom
41          '0' + cifra. Na primer, '0' + 5 je '5' jer se karakter '5'
42          nalazi 5 mesta nakon karaktera '0' u ASCII tablici. */
43         cifra = broj % 10;
44         broj = broj / 10;
45         s[i] = '0' + cifra;
46         i--;
47     } while(broj);
48 }
49
50 int main()
51 {
52     /* Deklaracija broja i niske. */
53     int n;
54     char s[MAKS_NISKA];
55
56     /* Ucitava se broj. */
57     printf("Unesite ceo broj: ");
58     scanf("%d", &n);
59
60     /* Formira se niska. */
61     broj_u_nisku(n, s);
62
63     /* Ispis rezultata. */
64     printf("%s\n", s);
```

```

66     exit(EXIT_SUCCESS);
68 }

```

### Rešenje 3.5.17

```

1  #include <stdio.h>
2  #include <stdlib.h>
3  #include <ctype.h>
4
5  #define MAKS_NISKA 8
6
7  /* Funkcija koja proverava da li je prosledjeni karakter ispravna
   *   heksadekadna cifra. */
9  int heksa_cifra(char c){
10     c = toupper(c);
11
12     /* Cifra je ispravna ako je cifra ili ako je neko od slova:
13        A, B, C, D, E ili F. */
14     return isdigit(c) || (c >= 'A' && c <= 'F');
15 }
16
17 /* Funkcija koja proverava da li prosledjena niska s predstavlja
   *   ispravan heksadekadni broj. */
19 int heksadekadni_broj(char s[]){
20     int i;
21
22     /* Svaki heksadekasni broj pocinje sa 0x ili 0X. */
23     if (s[0] != '0' || toupper(s[1]) != 'X')
24         return 0;
25
26     /* Za svaki karakter niske s se proverava da li predstavlja
27        ispravnu heksadekadnu cifru. Ako se naidje na neku cifru koja
28        ne zadovoljava taj uslov, onda se kao povratna vrednost
29        vraca nula. */
30     for (i = 2; s[i]; i++)
31         if (!heksa_cifra(s[i]))
32             return 0;
33
34     /* Ako su sve cifre ispravne heksadekadne cifre, onda je i s
35        ispravan heksadekadni broj i funkcija vraca jedinicu. */
36     return 1;
37 }
38
39 int main()
40 {
41     /* Deklaracija niske. */
42     char s[MAKS_NISKA];
43
44     /* Ucitava se niska. */
45     printf("Unesite nisku: ");

```

### 3 Predstavljanje podataka

```
scanf("%s", s);

47
/* Ispis rezultata. */
49 if (heksadekadni_broj(s))
    printf("Korektan heksadekadni broj.\n");
51 else
    printf("Nekorektan heksadekadni broj.\n");
53
    exit(EXIT_SUCCESS);
55 }
```

#### Rešenje 3.5.18

```
1 #include <stdio.h>
2 #include <stdlib.h>
3 #include <string.h>
4 #include <ctype.h>
5
6 #define MAKS_NISKA 8
7
8 /* Funkcija koja racuna dekadnu vrednost jedne heksadekadne cifre.
9    Ako je c broj, vrednost se dobija oduzimanjem '0'.
10    Ako je c slovo, vrednost se dobija oduzimanjem 'A' i dodavanjem
11    10 (npr. vrednost karaktera 'B' je 10 + 'B' - 'A' = 11). */
12 int vrednost_heksa_cifre(char c)
13 {
14     if(isdigit(c))
15         return c - '0';
16     else
17         return 10 + toupper(c) - 'A';
18 }
19
20 /* Funkcija koja racuna dekadnu vrednost heksadekadnog broja */
21 int dekadna_vrednost(char s[])
22 {
23     int i, tezina_pozicije = 1, rezultat = 0;
24     int n = strlen(s);
25
26     /* Vrsi se prolazak kroz nisku sa desna na levo. Heksadekadna
27        cifra najvece tezine se nalazi na poziciji n-1, a ona najvece
28        tezine se nalazi na poziciji 2 (jer su prva dva karaktera 0x).
29
30        U svakoj iteraciji, na rezultat se dodaje vrednost tekuce
31        cifre, pomnozena sa vrednoscu tezine njene pozicije.
32        Na primer, za s = "0x1a8e", n=6
33        i = 5, rezultat += vrednost('e')*1 => rezultat += 11*1
34        i = 4, rezultat += vrednost('8')*16 => rezultat += 8*16
35        i = 3, rezultat += vrednost('a')*256 => rezultat += 10*256
36        i = 2, rezultat += vrednost('1')*4096 => rezultat += 1*4096 */
37     for (i = n - 1; i >= 2; i--) {
        rezultat += tezina_pozicije * vrednost_heksa_cifre(s[i]);
    }
```

```

39     tezina_pozicije *= 16;
40 }
41
42     return rezultat;
43 }
44
45 int main()
46 {
47     /* Deklaracija niske. */
48     char s[MAKS_NISKA];
49
50     /* Ucitava se niska. */
51     printf("Unesite nisku: ");
52     scanf("%s", s);
53
54     /* Ispis rezultata. */
55     printf("%d\n", dekadna_vrednost(s));
56
57     exit(EXIT_SUCCESS);
58 }

```

### Rešenje 3.5.19

```

1  #include <stdio.h>
2  #include <stdlib.h>
3  #include <string.h>
4
5  #define MAKS_LINIJA 81
6
7  /* Funkcija koja ucitava liniju maksimalne duzine n i upisuje
8     je u s. Funkcija ne smesta znak za novi red na kraj linije. */
9  int ucitaj_liniju(char s[], int n)
10 {
11     int i = 0;
12     int c;
13
14     /* Ucitava se karakter po karakter dok se ne unese novi red
15        ili oznaka za kraj ulaza ili dok se ne dostigne maksimalan
16        broj karaktera. */
17     while ((c = getchar()) != '\n' && i < n-1 && c != EOF) {
18         s[i] = c;
19         i++;
20     }
21
22     /* Maksimalan broj karaktera za liniju je n-1 jer na kraju
23        treba ostaviti i jedno mesto za terminalnu nulu. */
24     s[i] = '\0';
25
26     return i;
27 }

```

### 3 Predstavljanje podataka

```
29 int main()
{
31     /* Deklaracija potrebnih promenljivih. */
    char linija[MAKS_LINIJA], najduza_linija[MAKS_LINIJA];
33     int duzina_najduze = 0, duzina;

35     /* U petlji se ucitavaju linije sve dok se ne unese prazna linija.
        Ukoliko se unese linija koja je duza od trenutno najduze,
37     vrsi se azuriranje duzine najduze linije, kao i same linije. */
    while ((duzina = ucitaj_liniju(linija, MAKS_LINIJA)) > 0) {
39         if (duzina_najduze < duzina) {
            duzina_najduze = duzina;
41             strcpy(najduza_linija, linija);
        }
43     }

45     /* Ispis rezultata. */
    if(duzina_najduze == 0)
47         printf("Nije uneta nijedna linija.\n");
    else
49         printf("%s\n%d\n", najduza_linija, duzina_najduze);

51     exit(EXIT_SUCCESS);
}
```

#### Rešenje 3.5.20

```
#include <stdio.h>
2 #include <stdlib.h>
#include <string.h>
4 #include <ctype.h>

6 #define MAKS_RECENICA 81

8 /* Funkcija koja ucitava recenicu maksimalne duzine n. */
int ucitaj_recenicu(char s[], int n)
10 {
    int i = 0, c;
12

14     /* Preskacu se beline sa pocetka ako ih ima. Po zavrsetku ove
        petlje u c se nalazi prvi sledeci karakter koji nije belina. */
    do{
16         c = getchar();
    } while(isspace(c));
18

20     /* Ako je taj karakter EOF, zavrшава se ucitavanje. */
    if(c == EOF)
        return 0;
22

24     /* U nisku se smesta karakter, prelazi se na sledeci karakter i
        postupak se ponavlja sve dok se ne unese tacka, EOF ili dok
```

```
26     se ne popuni maksimalan broj karaktera koje recenica moze
    da sadrzi. */
27 do{
28     s[i] = c;
    i++;
30     c = getchar();
} while (c != '.' && i < n-2 && c != EOF);
32
/* Ako je poslednji uneti karakter EOF, završava se učitavanje. */
34 if(c == EOF)
    return 0;
36
/* Na kraju svake recenice stoji tacka za kojom sledi '\0'. */
38 s[i] = '.';
s[i+1] = '\0';
40
    return i+1;
42 }

44 /* Funkcija koja prebrojava mala i velika slova. */
void prebroj(char s[] , int* broj_malih, int* broj_velikih)
46 {
    int mala = 0, velika = 0, i;
48
    for(i=0; s[i]; i++)
    {
50         if(islower(s[i]))
            mala++;
52         else if(isupper(s[i]))
            velika++;
54     }
56
    *broj_malih = mala;
58     *broj_velikih = velika;
}

60
int main()
62 {
    /* Deklaracija potrebnih promenljivih. */
64     char recenica[MAKS_RECENICA];
    char rezultujuca_recenica[MAKS_RECENICA];
66     int najveca_razlika = -1, trenutna_razlika;
    int mala, velika;
68     int ucitana_bar_jedna = 0;

70     /* U petlji se ucitavaju recenice sve dok se ne unese EOF. */
    while (ucitaj_recenicu(recenica, MAKS_RECENICA) > 0) {
72
        /* Prebrojavaju se mala i velika slova. */
74         prebroj(recenica, &mala, &velika);

76         /* Racuna se njihova apsolutna razlika. */
```

### 3 Predstavljanje podataka

---

```
    trenutna_razlika = abs(mala-velika);
78
    /* Ako je razlika veca od trenutno najvece, azurira se vrednost
80     najvece razlike i pamti se trenutna recenica. */
    if(trenutna_razlika > najveca_razlika){
82         najveca_razlika = trenutna_razlika;
        strcpy(rezultujuca_recenica, recenica);
84     }

    /* Indikator koji oznacava da se petlja bar jednom izvorsila, tj.
86     da korisnik nije odmah zadao EOF. */
    ucitana_bar_jedna = 1;
88 }

90
    /* Ispis rezultata. */
    if(ucitana_bar_jedna)
92         printf("%s\n", rezultujuca_recenica);
    else
94         printf("Nije uneta nijedna recenica. ");

96     exit(EXIT_SUCCESS);
98 }
```

#### Rešenje 3.5.21

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

4 #define MAKS_NISKA 21

6 /* Funkcija vraca adresu prvog pojavljivanja karaktera c u niski s
   ili NULL ukoliko se c ne pojavljuje u s. */
8 char* strchr_klon(char s[], char c)
{
10     int i;

12     /* Za svaki karakter se proverava da li je jednak karakteru c.
       Ako se naidje na takav karakter, kao povratna vrednost funkcije
14     se vraca njegova adresa (&s[i]). */
    for(i=0; s[i]; i++)
16         if(s[i] == c)
            return &s[i];

18     /* Ako je petlja zavrшена, znaci da nije pronadjen karakter koji
       je jednak karakteru c i kao povratna vrednost funkcije se vraca
20     NULL pokazivac kao oznaka da se c ne nalazi u s. */
22     return NULL;
}

24
26 int main()
{
```



```

28  /* Deklaracije potrebnih promenljivih. */
    char s[MAKS_NISKA];
    char c;

30

    /* Ucitava se niska s. */
32  printf("Unesite nisku s: ");
    scanf("%s", s);

34

    /* Preskace se novi red koji je unet nakon niske s i
36     ucitava se karakter c. */
    getchar();
    printf("Unesite karakter c: ");
    scanf("%c", &c);

40

    /* Racunanje i ispis rezultata. */
42  char* p = strchr_klon(s, c);
    if(p == NULL)
44      printf("Pozicija: -1\n");
    else
46      printf("Pozicija: %ld\n", p-s);

48  exit(EXIT_SUCCESS);
}

```

### Rešenje 3.5.22

```

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

4

#define MAKS_NISKA 21

6

/* Funkcija koja racuna duzinu prefiksa niske t koji se moze
8  zapisati pomocu karaktera niske s.
   Na primer, t="programiranje", s="grupacija", rezultat je 2 jer
10  niska s sadrzi prva dva karaktera niske t, ali ne i treci. */
int strspn_klon(char t[], char s[])
12 {
    int i, brojac = 0;

14

    /* Ide se redom po karakterima niske t i za svaki karakter se
16     vrsi provera da li se on nalazi u zapisu niske s. Za ovo se
       koristi funkcija strchr. Ako se nalazi, uvecava se brojac,
18     a ako se ne nalazi, prekida se petlja. */
    for(i=0; t[i]; i++){
20        if(strchr(s, t[i]) != NULL)
            brojac++;
22        else
            break;
24    }
}

```

### 3 Predstavljanje podataka

---

```
26     return brojac;
27 }
28
29 int main()
30 {
31     /* Deklaracije potrebnih promenljivih. */
32     char s[MAKS_NISKA];
33     char t[MAKS_NISKA];
34
35     /* Ucitavaju se niske. */
36     printf("Unesite nisku t: ");
37     scanf("%s", t);
38     printf("Unesite nisku s: ");
39     scanf("%s", s);
40
41     /* Racunanje i ispis rezultata. */
42     printf("%d\n", strspn_klon(t,s));
43
44     exit(EXIT_SUCCESS);
45 }
```

#### Rešenje 3.5.23

Rešenje ovog zadatka se svodi na rešenje zadatka 3.5.22, uz razliku da se ovde prebrojavaju karakteri koji se ne nalaze u zapisu niske s.

#### Rešenje 3.5.24

```
1  #include <stdio.h>
2  #include <stdlib.h>
3  #include <string.h>
4
5  #define MAKS_NISKA 101
6
7  /* Funkcija koja ucitava liniju maksimalne duzine n i upisuje
8   * je u s. Funkcija ne smesta znak za novi red na kraj linije. */
9  void ucitaj_liniju(char s[], int n)
10 {
11     int i = 0, c;
12
13     while ((c = getchar()) != '\n' && i < n-1 && c != EOF) {
14         s[i] = c;
15         i++;
16     }
17
18     s[i] = '\0';
19 }
20
21 /* Funkcija koja vraca pokazivac na prvo pojavljivanje niske
22  * t u okviru niske s ili NULL ukoliko se t ne nalazi u s. */
```

```
23 char* strstr_klon(char s[], char t[])
24 {
25     int i, j;
26
27     /* Spoljasnja petlja ide redom po niski s. */
28     for (i = 0; s[i] != '\0'; i++) {
29         /* Unutrasnja petlja ide redom po niski t pomocu brojaca j
30            i proverava da li se cela niska t poklapa sa delom niske
31            s koji pocinje na poziciji i.
32
33            Cim se naidje na situaciju da se karakteri ne poklapaju,
34            izlazi se iz unutrasnje petlje. */
35         for (j = 0; t[j] != '\0'; j++)
36             if (s[i+j] != t[j])
37                 break;
38
39         /* Ako je unutrasnja petlja dosla do kraja niske t, to
40            znaci da su se svi karakteri iz t poklopili sa karakterima
41            iz s i t je podniska od s.
42            Kao povratna vrednost se vraca mesto gde t pocinje u s. */
43         if (t[j] == '\0')
44             return &s[i];
45     }
46
47     return NULL;
48 }
49
50 int main()
51 {
52     /* Deklaracije potrebnih promenljivih. */
53     char linija[MAKS_NISKA];
54     int i, bar_jedna = 0;
55
56     /* Ucitavanje linija i ispis rednih brojeva linija
57        koje sadrze rec "program". */
58     for(i=1; i<=5; i++)
59     {
60         ucitaj_liniju(linija, MAKS_NISKA);
61         if(strstr_klon(linija, "program") != NULL){
62             printf("%d ", i);
63             bar_jedna = 1;
64         }
65     }
66     printf("\n");
67
68     /* Ako indikator bar_jedna i dalje ima vrednost 0, znaci da
69        nije uneta nijedna linija koja sadrzi rec "program". */
70     if(!bar_jedna)
71     {
72         printf("Nijedna linija ne sadrzi nisku program.\n");
73     }
```

```
75     exit(EXIT_SUCCESS);  
    }
```

#### Rešenje 3.5.25

```
1  #include <stdio.h>  
2  #include <stdlib.h>  
3  
4  #define MAKS_NISKA 21  
5  
6  /* Funkcija koja poredi dve niske. Niske je potrebno porediti  
7     karakter po karakter. */  
8  int strcmp_klon(char s1[], char s2[])  
9  {  
10     int i;  
11  
12     /* Prolazi se kroz obe niske dok god se odgovarajuci karakteri  
13        poklapaju. Ako se u ovom prolasku desi da je petlja dosla  
14        do kraja obe niske, onda su one jednake i kao povratna vrednost  
15        funkcije se vraca 0. */  
16     for (i = 0; s1[i] == s2[i]; i++)  
17         if (s1[i] == '\0')  
18             return 0;  
19  
20     /* Ako niske nisu jednake, znaci da je brojac i stao na prvom  
21        mestu gde se niske s1 i s2 razlikuju. Posto funkcija treba da  
22        vrati pozitivnu vrednost ako je niska s1 leksikografski iza  
23        s2, a negativnu u suprotnom, ovo moze biti realizovano vraćanjem  
24        razlike ASCII kodova.  
25        Na primer: s1 = "pero", s2 = "program"  
26        Nakon petlje, brojac i ima vrednost 1 (jer je tu prva razlika).  
27        Kao povratna vrednost se vraca s1[1] - s2[1] = 'e' - 'r' = -13  
28        sto kao negativna vrednost govori da se s1 nalazi  
29        leksikografski ispred s2. */  
30     return s1[i] - s2[i];  
31 }  
32  
33 int main()  
34 {  
35     /* Deklaracije potrebnih promenljivih. */  
36     char s[MAKS_NISKA];  
37     char t[MAKS_NISKA];  
38     int rezultat;  
39  
40     /* Ucitavaju se niske s i t. */  
41     printf("Unesite nisku s: ");  
42     scanf("%s", s);  
43     printf("Unesite nisku t: ");  
44     scanf("%s", t);  
45  
46     /* Vrsi se poredjenje niski i ispisuje se rezultat. */
```

```

47 rezultat = strcmp_klon(s,t);
   if(rezultat == 0)
49     printf("%s\n", s);
   else if(rezultat < 0)
51     printf("%s\n%s\n", s, t);
   else
53     printf("%s\n%s\n", t, s);

55 exit(EXIT_SUCCESS);
}

```

### Rešenje 3.5.26

```

1  #include <stdio.h>
   #include <stdlib.h>
3  #include <string.h>

5  #define MAKS_NISKA 21

7  /* Funkcija koja obrce nisku s. */
void obrni(char s[]){
9     int i, j;
    int n = strlen(s);
11    char c;

13    /* Brojac i ide od prvog karaktera niske s, a brojac j
       od poslednjeg i dok god se ne sretnu, vrsi se zamena
15    karaktera koji se nalaze na njihovim pozicijama. */
    for(i=0, j=n-1; i<j; i++,j--){
17        c = s[i];
        s[i] = s[j];
19        s[j] = c;
    }
21 }

23 int main()
{
25     /* Deklaracija niske. */
    char s[MAKS_NISKA];

27     /* Ucitava se niska. */
29     printf("Unesite nisku: ");
    scanf("%s", s);

31     /* Racunanje i ispis rezultata. */
33     obrni(s);
    printf("%s\n", s);

35     exit(EXIT_SUCCESS);
37 }

```

#### Rešenje 3.5.27

```
1  #include <stdio.h>
2  #include <stdlib.h>
3  #include <string.h>
4
5  #define MAKS_NISKA 21
6
7  /* Funkcija koja rotira nisku za jedno mesto ulevo. */
8  void rotiraj1(char s[], int n)
9  {
10     int i;
11     /* Pamti se prvi karakter. */
12     char prvi = s[0];
13
14     /* Svaki sledeci karakter se pomera za jedno mesto ulevo. */
15     for(i=0; i<n-1; i++)
16         s[i] = s[i+1];
17
18     /* Prvi karakter se upisuje na kraj niske. */
19     s[n-1] = prvi;
20 }
21
22 /* Funkcija koja rotira nisku s za k mesta ulevo. */
23 void rotiraj(char s[], int k){
24     int i;
25     int n = strlen(s);
26
27     for(i=0; i<k; i++)
28         rotiraj1(s, n);
29 }
30
31 int main()
32 {
33     /* Deklaracija potrebnih promenljivih. */
34     char s[MAKS_NISKA];
35     int k;
36
37     /* Ucitavaju se niska i vrednost k. */
38     printf("Unesite nisku: ");
39     scanf("%s", s);
40     printf("Unesite k: ");
41     scanf("%d", &k);
42
43     /* Vrsi se provera ispravnosti ulaza. */
44     if(k < 0){
45         printf("Greska: neispravan unos.\n");
46         exit(EXIT_FAILURE);
47     }
48
49     /* Racunanje i ispis rezultata. */
50     rotiraj(s, k);
```

```

printf("%s\n", s);
52 exit(EXIT_SUCCESS);
54 }

```

### Rešenje 3.5.28

```

#include <stdio.h>
2 #include <stdlib.h>
#include <string.h>
4 #include <ctype.h>

6 #define MAKS_NISKA 21

8 /* Funkcija koja svako slovo niske s menja sa slovom
   koje se u ASCII tablici nalazi neposredno iza njega.
10 Specijalan slucaj je slovo z koje treba da se zameni
   sa slovom a. Ostali karakteri ostaju nepromenjeni. */
12 void sifruj(char s[]){
   int i;

14   for(i=0; s[i]; i++){
       if(isalpha(s[i])){
16         if(s[i] == 'z'){
             s[i] = 'a';
18         else if(s[i] == 'Z'){
             s[i] = 'A';
20         else
22             s[i] = s[i] + 1;
       }
24   }
}

26 int main()
28 {
   /* Deklaracija niske. */
30   char s[MAKS_NISKA];

   /* Ucitava se niska. */
32   printf("Unesite nisku: ");
34   scanf("%s", s);

   /* Racunanje i ispis rezultata. */
36   sifruj(s);
38   printf("%s\n", s);

40   exit(EXIT_SUCCESS);
}

```

#### Rešenje 3.5.29

```
1 #include <stdio.h>
2 #include <stdlib.h>
3 #include <ctype.h>
4
5 #define MAKS_DUZINA 20
6 #define MAKS_NISKA (MAKS_DUZINA + 1)
7 #define MAKS_REZULTAT (3*MAKS_DUZINA + 1)
8
9 /* Pomocna funkcija koja za prosledjeno slovo
10 vraca slovo koje ide posle njega. */
11 char sledeci(char c)
12 {
13     if(c == 'z')
14         return 'a';
15
16     if(c == 'Z')
17         return 'A';
18
19     return c+1;
20 }
21
22 /* Funkcija koja svako slovo niske s menja sa slovom
23 koje se u ASCII tablici nalazi neposredno iza njega.
24 Specijalan slucaj je slovo z koje treba da se zameni
25 sa slovom a. Ostali karakteri ostaju nepromenjeni. */
26 void sifruj(char s[], char rezultat[]){
27     int i, j;
28
29     /* Brojac i se koristi za nisku s, a brojac j za
30 rezultujucu nisku. */
31     for(i=0, j=0; s[i]; i++){
32         if(isalpha(s[i]))
33         {
34             /* Ako je s[i] slovo, onda se u rezultat upisuju 3 slova koja
35 slede nakon njega. */
36             rezultat[j] = sledeci(s[i]);
37             rezultat[j+1] = sledeci(rezultat[j]);
38             rezultat[j+2] = sledeci(rezultat[j+1]);
39             j += 3;
40         }
41         else
42         {
43             /* Ako s[i] nije slovo, onda se samo prepisuje u rezultat. */
44             rezultat[j] = s[i];
45             j++;
46         }
47     }
48
49     /* Na kraj rezultata se dopisuje terminalna nula. */
50     rezultat[j] = '\0';
```



```

}
52
int main()
54 {
    /* Deklaracija niske. */
56     char s[MAKS_NISKA];
    char rezultat[MAKS_REZULTAT];
58
    /* Ucitava se niska. */
60     printf("Unesite nisku: ");
    scanf("%s", s);
62
    /* Racunanje i ispis rezultata. */
64     sifruj(s, rezultat);
    printf("%s\n", rezultat);
66
    exit(EXIT_SUCCESS);
68 }

```

### Rešenje 3.5.30

```

#include <stdio.h>
2  #include <stdlib.h>
#include <ctype.h>
4
#define MAKS_DUZINA 20
6  #define MAKS_NISKA (MAKS_DUZINA + 1)
#define MAKS_REZULTAT (2*MAKS_DUZINA + 1)
8
/* Funkcija koja svako slovo niske s menja sa slovom
10 koje se u ASCII tablici nalazi neposredno iza njega.
Specijalan slucaj je slovo z koje treba da se zameni
12 sa slovom a. Ostali karakteri ostaju nepromenjeni. */
void formiraj(char s[], char rezultat[], char c1, char c2){
14     int i, j;

16     /* Brojac i se koristi za nisku s, a brojac j za
        rezultujucu nisku. */
18     for(i=0, j=0; s[i]; i++){
        if(s[i] == c1){
20             /* Ako je s[i] jednako c1, duplira se u rezultatu. */
            rezultat[j] = s[i];
22             rezultat[j+1] = s[i];
            j += 2;
24         }
        else if(s[i] != c2) {
26             /* Ako s[i] razlicito od c2, upisuje se u rezultat. */
            rezultat[j] = s[i];
28             j++;
        }
30     }
}

```

```
32  /* Na kraj rezultata se dopisuje terminalna nula. */
    rezultat[j] = '\0';
34  }

36  int main()
  {
38      /* Deklaracija potrebnih promenljivih. */
      char s[MAKS_NISKA];
40      char rezultat[MAKS_REZULTAT];
      char c1, c2;

42      /* Ucitavaju se niska i karakteri. */
44      printf("Unesite nisku: ");
      scanf("%s", s);
46      getchar();
      printf("Unesite prvi karakter: ");
48      scanf("%c", &c1);
      getchar();
50      printf("Unesite drugi karakter: ");
      scanf("%c", &c2);

52      /* Vrsi se provera ispravnosti ulaza. */
54      if(c1 == c2)
      {
56          printf("Greska: neispravan unos.\n");
          exit(EXIT_FAILURE);
58      }

60      /* Racunanje i ispis rezultata. */
      formiraj(s, rezultat, c1, c2);
62      printf("%s\n", rezultat);

64      exit(EXIT_SUCCESS);
  }
```

#### Rešenje 3.5.31

```
#include <stdio.h>
2 #include <stdlib.h>
  #include <string.h>
4 #include <ctype.h>

6 #define MAKS_NISKA 20

8 /* Pomocna funkcija koja racuna dekadnu vrednost prosledjenog
   karaktera ('1' ima vrednost 1, 'C' ima vrednost 12). */
10 unsigned vrednost_cifre(char c){
    c = toupper(c);
12    if(isdigit(c))
        return c - '0';
```

```

14     else
15         return c - 'A' + 10;
16 }

18 /* Funkcija koja racuna dekadnu vrednost neonacenog broja
   zapisanog u datoj osnovi. */
20 unsigned int u_dekadni_sistem(char broj[], unsigned int osnova){
21     int i, n = strlen(broj);
22     int rezultat = 0;
23     int tezina_pozicije = 1;
24
25     for(i=n-1; i>=0; i--){
26         rezultat += vrednost_cifre(broj[i])*tezina_pozicije;
27         tezina_pozicije *= osnova;
28     }
29
30     return rezultat;
31 }

32 /* Pomocna funkcija koja obrce nisku. */
34 void obrni(char s[]){
35     int i, j;
36     int n = strlen(s);
37     char c;
38
39     for(i=0, j=n-1; i<j; i++,j--){
40         c = s[i];
41         s[i] = s[j];
42         s[j] = c;
43     }
44 }

46 /* Pomocna funkcija koja dekadnu vrednost cifre pretvara u
   odgovarajuci karakter (12 u 'C', 5 u '5', itd.). */
48 char ostatak_u_char(int ostatak){
49     if(ostatak < 10)
50         return '0' + ostatak;
51     else
52         return 'A' + ostatak - 10;
53 }

54 /* Funkcija koja datu dekadnu vrednost broja prebacuje u broj u
   datoj osnovi. */
56 void iz_dekadnog_sistema(unsigned int broj, unsigned int osnova,
57                          char rezultat[]){
58     int i = 0;
59     int ostatak;
60
61     do{
62         ostatak = broj % osnova;
63         broj = broj / osnova;
64         rezultat[i] = ostatak_u_char(ostatak);

```

### 3 Predstavljanje podataka

---

```
66     i++;
67     }while(broj);
68
69     rezultat[i] = '\\0';
70
71     obrni(rezultat);
72 }
73
74 int main()
75 {
76     char broj[MAKS_NISKA];
77     char broj2[MAKS_NISKA];
78     unsigned int osnova1, osnova2;
79
80     printf("Unesite n, o1 i o2: ");
81     scanf("%s%u%u", broj, &osnova1, &osnova2);
82
83     unsigned dekadna_vrednost = u_dekadni_sistem(broj, osnova1);
84     printf("Dekadna vrednost broja %s: %u\\n", broj, dekadna_vrednost);
85
86     iz_dekadnog_sistema(dekadna_vrednost, osnova2, broj2);
87     printf("Vrednost broja %u u osnovi %u: %s\\n", dekadna_vrednost,
88           osnova2, broj2);
89
90     exit(EXIT_SUCCESS);
91 }
```