

## PROGRAMIRANJE 1



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

# **PROGRAMIRANJE 1**

## **Zbirka zadataka**

**Beograd  
2016.**

Autori:

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

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

*Danijela Simić*, asistent na Matematičkom fakultetu u Beogradu

*Anđelka Zečević*, asistent na Matematičkom fakultetu u Beogradu

PROGRAMIRANJE 1

Zbirka zadataka

# Sadržaj

<b>1</b>	<b>Uvodni zadaci</b>	<b>1</b>
1.1	Naredba izraza . . . . .	1
1.2	Rešenja . . . . .	11
<b>2</b>	<b>Kontrola toka</b>	<b>29</b>
2.1	Naredbe grananja . . . . .	29
2.2	Rešenja . . . . .	41
2.3	Petlje . . . . .	67
2.4	Rešenja . . . . .	94



# 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 ispisuje najpre unete vrednosti, a zatim i njihov zbir, razliku, proizvod, ceo deo pri deljenju prvog broja drugim brojem i ostatak pri deljenju prvog broja drugim brojem.

## 1 Uvodni zadaci

---

NAPOMENA: *Pretpostaviti da je unos korektan, tj. da druga uneta vrednost nije 0.*

### Primer 1

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

### Primer 2

```
INTERAKCIJA SA PROGRAMOM:
Unesi vrednost celobrojne promenljive x: -3
Unesi vrednost celobrojne 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. NAPOMENA: *Pretpostaviti da su cene artikala pozitivni celi brojevi i da je unos korektan.*

### Primer 1

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

### Primer 2

```
INTERAKCIJA SA PROGRAMOM:
Unesi cenu prvog artikla: 384
Unesi 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. NAPOMENA: *Pretpostaviti da je cena jabuka pozitivan ceo broj i da je unos korektan.*

### 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. NAPOMENA: *Pretpostaviti*



da su cene svih artikala pozitivni celi brojevi, kao i da su unete vrednosti ispravne, tj. da se može vratiti kusur.

### 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. Napisati program koji razmenjuje njihove vrednosti.

### Primer 1

```
INTERAKCIJA SA PROGRAMOM:
Unesi dve celobrojne vrednosti: 5 7
pre zamene: x=5, y=7
posle zamene: x=7, y=5
```

### Primer 2

```
INTERAKCIJA SA PROGRAMOM:
Unesi dve celobrojne vrednosti: 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:
Unesi dve celobrojne vrednosti: 5 7
Nove vrednosti su: a=12, b=-2
```

### Primer 2

```
INTERAKCIJA SA PROGRAMOM:
Unesi dve celobrojne vrednosti: 237 -592
Nove vrednosti su: a=-355, b=829
```

## 1 Uvodni zadaci

---

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

### Primer 1

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

### Primer 2

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesi 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. NAPOMENA: *Pretpostaviti da je cena proizvoda pozitivan ceo broj.*

### 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 sa standardnog ulaza i ispisuje broj dobijen obrtanjem njegovih cifara. NAPOMENA: *Pretpostaviti da je unos ispravan.*

### Primer 1

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesi trocifreni broj: 892  
|| Obrnuto: 298
```

### Primer 2

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesi 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 prirodnom broju.

*Primer 1*

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

*Primer 2*

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

**Zadatak 1.1.15** Sa standardnog unosa se unosi pozitivan prirodan broj  $n$  i pozitivan dvocifreni broj  $m$ . Napisati program 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.*

## 1 Uvodni zadaci

---

### Primer 1

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

### Primer 2

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesite pozitivan prirodan 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 decimale. UPUTSTVO: *Jedan inč ima 2.54 centimetra.*

### Primer 1

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

### Primer 2

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesi 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 decimale. UPUTSTVO: *Jedna milja ima 1.609344 kilometara.*

### Primer 1

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

### Primer 2

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesi 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 decimale. UPUTSTVO: *Jedna funta ima 0.45359237 kilograma.*

### Primer 1

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

### Primer 2

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

[Rešenje 1.1.18]

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

*Primer 1*

```
INTERAKCIJA SA PROGRAMOM:
Unesi temperaturu u F: 100.93
100.93 F = 38.29 C
```

*Primer 2*

```
INTERAKCIJA SA PROGRAMOM:
Unesi temperaturu u F: 25.562
25.562 F = -3.58 C
```

[Rešenje 1.1.19]

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

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

Pri ispisu vrednost zaokružiti na 4 decimale.

*Primer 1*

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

*Primer 2*

```
INTERAKCIJA SA PROGRAMOM:
Unesite brojeve: -1 0 0 1
-1.0000
```

*Primer 3*

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

*Primer 4*

```
INTERAKCIJA SA PROGRAMOM:
Unesite brojeve: 0.01 0.01 0.5 7
0.0650
```

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

*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]

## 1 Uvodni zadaci

---

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

*Primer 1*

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesite duzinu poluprecnika kruga: 4.2  
|| Obim: 26.39, površina: 55.42
```

*Primer 2*

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesite duzinu poluprecnika kruga: 14.932  
|| Obim: 93.82, površina: 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 decimalne. NAPOMENA: *Pretpostaviti da je unos ispravan.*

*Primer 1*

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

*Primer 2*

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesite duzinu stranice trougla: 2  
|| Obim: 6.00  
|| Površina: 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 decimalne. NAPOMENA: *Pretpostaviti da je unos ispravan.*

*Primer 1*

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesite duzine stranica trougla: 3 4 5  
|| Obim: 12.00  
|| Površina: 6.00
```

*Primer 2*

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesite duzine stranica trougla: 4.3 9.7 8.8  
|| Obim: 22.80  
|| Površina: 18.91
```

[Rešenje 1.1.24]

**Zadatak 1.1.25** Pravougaonik čije su stranice paralelne koordinatnim osama zadat je svojim realnim koordinatama suprotnih temena (gornje levo i donje desno teme). Napisati program koji ispisuje njegov obim i površinu zaokružene na dve decimalne.

*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
```

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

*Primer 1*

```
INTERAKCIJA SA PROGRAMOM:
Unesite tri cela broja: 11 5 4
Aritmeticka sredina unetih brojeva je 6.67
```

*Primer 2*

```
INTERAKCIJA SA PROGRAMOM:
Unesite tri cela broja: 3 -8 13
Aritmeticka sredina unetih brojeva je 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 dimenzije 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 cene usluge po kvadratnom metru program izračuna ukupnu cenu krečenja. Sve realne vrednosti ispisati zaokružene na dve decimalne.

*Primer 1*

```
INTERAKCIJA SA PROGRAMOM:
Unesite dimenzije sobe: 4 4 3
Unesite cenu po m2: 500
Moler treba da okreci 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 okreci 320.80 m2
Cena krecenja je 152380.00
```

[Rešenje 1.1.27]

**Zadatak 1.1.28** Napisati program koji za unete pozitivne prirodne brojeve  $x$ ,  $p$  i  $c$  ispisuje broj koji se dobija ubacivanjem cifre  $c$  u broj  $x$  na poziciju  $p$ .

## 1 Uvodni zadaci

---

NAPOMENA: *Podrazumevati da je unos ispravan, tj. da je broj  $p$  manji od ukupnog broja cifara broja  $x$ . Numeracija cifara počinje od nule, odnosno cifra najmanje težine nalazi se na nultoj poziciji.* UPUTSTVO: *Koristiti funkciju `pow` iz `math.h` biblioteke.*

### 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 2.1.36]

**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

```
2  /* Navedeni program definise funkciju koja se zove main. Program
   * moze da definise vise funkcija, ali obavezno mora da definise
   * funkciju koja se zove main i izvorsavanje programa uvek pocinje
4  * od te funkcije. Pored naziva, zapis svake funkcije cine i
   * povratna vrednost funkcije (u ovom slucaju int), lista
6  * argumenata koje funkcija koristi (u ovom slucaju funkcija nema
   * argumenata pa se navode samo prazne zagrade, ()) i telo
8  * funkcije koje je ograniceno viticastim zagradama ({ i }). 0
   * ovim pojmovima bice vise reci u narednim poglavljima.

10
   *
12  Unutar tela funkcije navode se naredbe. Unutar navedenog
   * programa postoji jedna naredba koja predstavlja poziv funkcije
   * printf. Funkcija printf sluzi za ispis teksta na standardni
```

## 1 Uvodni zadaci

---

```
14      izlaz (obicno ekran). Deklaracija ove funkcije data je u
16      zaglavlju stdio.h koje je potrebno ukljuciti direktivom #include
na pocetku samog programa.
```

```
18      Da bismo pokrenuli program, prvo ga moramo prevesti u izvrsnu
20      datoteku. Na primer, ako je navedeni program sacuvan kao
22      zdravo.c, ako koristimo gcc kompajler koji je sastavni deo
standardnih Linux distribucija, prevodjenje iz komandne linije
se vrsi narednom naredbom: gcc zdravo.c Ukoliko nije bilo
gresaka prilikom prevodjenja, bice generisana izvrsna datoteka
24      pod nazivom a.out koja se pokrece navodjenjem sledece naredbe:
./a.out Ukoliko je bilo gresaka prilikom prevodjenja, one se
26      moraju otkloniti a postupak prevodjenja se mora ponoviti. */
```

```
28      #include<stdio.h>

30      int main()
31      {
32          /* printf: funkcija pomocu koje se vrsi ispis
33             Specijalni karakter \n : prelazak u novi red
34             Svaka naredba zavrшава se karakterom ; */
35          printf("Zdravo svima!\n");
36
37          /* Povratna vrednost 0 se obicno koristi da oznaci da je prilikom
38             izvrsavanja programa sve proslo u redu. */
39          return 0;
40      }
```

### Rešenje 1.1.2

```
#include <stdio.h>

2
int main()
3
4      {
5          /* Svaka promenljiva u programu mora biti deklarirana na pocetku
6             main funkcije. Deklaracija se sastoji iz naziva promenljive
7             (u ovom slucaju n) ispred kog se navodi tip promenljive (u
8             ovom slucaju celobrojni tip, int). */

9
10         int n;

11
12         /* Vrednost promenljive se ucitava pomocu funkcije scanf koja
13            je, kao i funkcija printf, sastavni deo standardne biblioteke.
14            Argumenti funkcije scanf koji se navode u zagradama ( i ) i
15            razdvajaju zarezima, oznacavaju sledece: "%d" - format za tip
16            podatka koji ce biti ucitan (%d za int, svaki tip ima svoj
17            format) &n - adresa promenljive x (o adresama ce biti vise
18            reci u narednim zadacima).

19
20            Ucitavanje se vrsi sa standardnog ulaza (obicno tastatura). */
21         printf("Unesite ceo broj: ");
```

```

22     scanf("%d", &n);

24     /* Funkcija printf ispisuje tekst "Uneti broj: ", a nakon toga,
        umesto formata %d, ispisuje vrednost promenljive n. */
26     printf("Uneti broj: %d\n", n);
        /* Umesto formata %d, ispisuje vrednost izraza n*n. */
28     printf("Kvadrat: %d\n", n * n);
        /* Umesto formata %d, ispisuje vrednost izraza n*n*n. */
30     printf("Kub: %d\n", n * n * n);

32     return 0;
}

```

### Rešenje 1.1.3

```

#include<stdio.h>

2
int main()
4
{
    /* Promenljive istog tipa mogu se deklarirati jedna za drugom. */
6
    int x, y, rezultat;

8
    printf("Unesi vrednost celobrojne promenljive x: ");
    scanf("%d", &x);

10
    printf("Unesi vrednost celobrojne promenljive y: ");
12
    scanf("%d", &y);

14
    /* Dodeljujemo vrednost promenljivoj rezultat. */
    rezultat = x + y;
16
    printf("%d + %d = %d\n", x, y, rezultat);

18
    /* Mozemo ispisivati direktno vrednost izraza x-y i bez njegovog
        dodeljivanja posebnoj promenljivoj */
20
    printf("%d - %d = %d\n", x, y, x - y);
    printf("%d * %d = %d\n", x, y, x * y);

22
    /* Kada bilo koju aritmeticku operaciju primenimo na dve
        promenljive istog tipa (u ovom slucaju dva celobrojne
        promenljive), rezultat ce biti tog istog tipa. Specijalno, za
        operaciju deljenja: kada operator / primenimo na dva
        celobrojna argumenta x i y, kao rezultat dobijemo 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. Na primer, 7%2 ima vrednost 1 (jer je
        7=3*2+1). Da bismo odstampali karakter %, u naredbi printf
        pisemo %% */
34
    printf("%d %% %d = %d\n", x, y, x % y);

36

```

```
38     return 0;
}
```

### 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

```
1  #include <stdio.h>
2
3  int main()
4  {
5      /* S obzirom da su sve promenljive pozitivini celi brojevi,
6         koristimo tip unsigned int (skraceno unsigned) */
7      unsigned cena;
8      unsigned kolicina;
9      unsigned iznos;
10     unsigned kusur;
11
12     /* Ucitavamo potrebne podatke. Unutar jednog scanf-a mozemo
13        ucitati vise podataka odjednom. Za svaki treba navesti
14        odgovarajuci format za tip podataka koji se unosi (%u za
15        unsigned). */
16     printf("Unesite cenu, kolicinu i iznos: ");
17     scanf("%u%u%u", &cena, &kolicina, &iznos);
18
19     /* Izracunavamo kusur: */
20     kusur = iznos - kolicina * cena;
21
22     /* I ispisujemo trazenu vrednost: */
23     printf("Kusur je %u dinara.\n", kusur);
24
25     return 0;
26 }
```

### Rešenje 1.1.7

```
1  #include <stdio.h>
2
3  int main()
4  {
5
6      unsigned poletanje, poletanje_sat, poletanje_minut;
7      unsigned sletanje, sletanje_sat, sletanje_minut;
8      unsigned duzina, duzina_sat, duzina_minut;
9
10     printf("Unesite vreme poletanja: ");
11     scanf("%u%u", &poletanje_sat, &poletanje_minut);
12
13
14     printf("Unesite vreme sletanja: ");
15     scanf("%u%u", &sletanje_sat, &sletanje_minut);
16
17     /* Pretvoricemo i vreme poletanja i vreme sletanja u sekunde */
18     poletanje = poletanje_sat * 3600 + poletanje_minut * 60;
19     sletanje = sletanje_sat * 3600 + sletanje_minut * 60;
20
21     /* I izracunati razliku u sekundama */
22     duzina = sletanje - poletanje;
23
24     /* Izdvajamo broj sati i broj minuta. */
25     duzina_sat = duzina / 3600;
26     duzina_minut = (duzina % 3600) / 60;
27
28     /* I ispisujemo rezultat */
29     printf("Duzina trajanja leta je %u h i %u min\n", duzina_sat,
30           duzina_minut);
31
32     return 0;
33 }
```

### Rešenje 1.1.8

```
1  #include<stdio.h>
2
3  int main()
4  {
5      int x, y;
6      int p;
7
8      printf("Unesi dve celobrojne vrednosti:");
9      scanf("%d%d", &x, &y);
10
11     printf("pre zamene: x=%d, y=%d\n", x, y);
12
13     /* Pomocna promenljiva p je potrebna da sacuva vrednost
14        promenljive x pre nego sto se ona izmeni i dobije vrednost
```

## 1 Uvodni zadaci

```
    promenljive y. */
16  p = x;
    x = y;
18  y = p;
    printf("posle zamene: x=%d, y=%d\n", x, y);
20
    return 0;
22 }
```

### Rešenje 1.1.10

```
1  #include <stdio.h>
2
3  int main()
4  {
5      /* S obzirom da broj treba da bude pozitivan, koristimo tip
        unsigned. */
6      unsigned x;
7
8      /* Promenljive koje cuvaju cifre treba da budu najmanjeg
        celobrojnog tipa jer nece sadrzati druge vrednosti osim
11     jednocifrenih celih brojeva. Zbog toga za njih biramo tip
        char. */
12     char cifra_jedinice;
13     char cifra_desetice;
14     char cifra_stotine;
15
16     printf("Unesi trocifreni broj:");
17     scanf("%u", &x);
18
19     /* Na primer, neka je uneti broj 374. Potrebno je da koriscenjem
        racunskih operacija za rad sa celim brojevima pristupimo
21     njegovoj cifri jedinice, cifri desetice i cifri stotine.
22
23     Prisetimo najpre sledece:  $374/10 = 37$   $374\%10 = 4$  Dakle,
        operacijama celobrojnog deljenja i ostatka pri deljenju mozemo
25     iz svakog broja izdvojiti njegovu poslednju cifru (u ovom
        slucaju 4) i broj sastavljen od svih cifara osim poslednje (u
27     ovom slucaju 37).
28
29     Cifri jedinice sada lako pristupamo koriscenjem ostatka pri
        deljenju sa 10. Ona iznosi upravo 4.
31
32     Pri trazanju cifre desetice mozemo ponovo primeniti princip
        izdvajanja poslednje cifre kao ostatka pri deljenju sa 10.
33     Razlika je sto ne mozemo desetice izdvojiti ako primenimo %10
        na 374 (tada dobijamo 4), vec %10 primenjujemo na 37, pri cemu
35     37 dobijamo kao ceo deo pri deljenju broja 374 brojem 10.
        Dakle, cifru desetice dobijamo kao  $(374/10)\%10$ .
37
38     S obzirom da znamo da je u pitanju trocifreni broj, cifru
39
```

```

41     stotine mozemo izdvojiti celobrojn timer deljenjem sa 100: 374/100
        iznosi upravo 3. */
43     cifra_jedinice = x % 10;
        cifra_desetice = (x / 10) % 10;
45     cifra_stotine = x / 100;

47     /* Ako zelimo da odstampamo numericku vrednost promenljive tipa
        char, koristimo format %d. Ako zelimo da odstampamo karakter
49     ciji je ASCII kod jednak vrednosti te promenljive, koristimo %c
        (na primer, ako bismo promenljivu cija je vrednost 65 stampali
51     pomocu formata %d, ispis bi bio 65, ali ako bismo je stampali
        pomocu formata %c, ispis bi bio A). U ovom slucaju nam je
53     neophodna numericka vrednost. */
        printf("jedinica %d, desetice %d, stotina %d\n", cifra_jedinice,
55             cifra_desetice, cifra_stotine);

57     /* 2. nacin, bez uvodjenja dodatnih promenljivih cifra_jedinice,
        cifra_desetice i cifra_stotine:

59         printf("Cifre unetog broja su %d,%d,%d\n", x%10, (x/10)%10,
61         x/100); */
        return 0;
63 }

```

### Rešenje 1.1.11

```

1  #include <stdio.h>

3  int main()
    {
5      unsigned x;
        printf("Unesi cenu:");
7      scanf("%u", &x);

9      /* Na primer, neka je uneta cena 8347 dinara. Vrednost x/5000
        predstavlja broj novcanica od 5000 dinara pomocu kojih mozemo
11     sakupiti celokupnu sumu. 8347 celobrojno deljeno sa 5000
        (operacija / nad celim brojevima) iznosi 1. */
13     printf("%u=%u*5000+ ", x, x / 5000);
        /* Potrebna nam je 1 novcanica od 5000 dinara, a koliko nam je
15     potrebno ostalih novcanica? Za to moramo pristupiti preostaloj
        sumi. Jedan nacin je da nadjemo ostatak pri deljenju unete
17     vrednosti x (u primeru 8347) sa 5000 (operacija %). On iznosi
        3347. Ovu vrednost dodeljujemo promenljivoj x. */
19     x = x % 5000;

21     /* Nastavljamo postupak trazanjem broja novcanica od 2000 dinara
        i redom za ostale monete. */
23     printf("%u*2000 +", x / 2000);
        x = x % 2000;
25     printf("%u*1000 +", x / 1000);

```

## 1 Uvodni zadaci

---

```

    x = x % 1000;
27  printf("%u*500 +", x / 500);
    x = x % 500;
29  printf("%u*200 +", x / 200);
    x = x % 200;
31  printf("%u*100 +", x / 100);
    x = x % 100;
33  printf("%u*50 +", x / 50);
    x = x % 50;
35  printf("%u*20 +", x / 20);
    x = x % 20;
37  printf("%u*10 +", x / 10);
    x = x % 10;
39  printf("%u*1\n", x);

41  return 0;
}
```

### Rešenje 1.1.12

```

1  #include <stdio.h>

3  int main()
{
5      unsigned x;
      unsigned obrnuto_x;
7      char cifra_jedinice;
      char cifra_desetice;
9      char cifra_stotine;

11     printf("Unesi trocifreni broj:");
      scanf("%u", &x);

13

      cifra_jedinice = x % 10;
15     cifra_desetice = (x / 10) % 10;
      cifra_stotine = x / 100;

17

      obrnuto_x = cifra_jedinice * 100 +
19         cifra_desetice * 10 + cifra_stotine;

21     printf("Obrnuto: %u\n", obrnuto_x);

23     return 0;
}
```

### Rešenje 1.1.13

```

1  #include <stdio.h>
```



```

3 int main()
4 {
5
6     unsigned n, broj_obrnuto, broj_zamena;
7     char j, d, s, h;
8     int proizvod_cifara, razlika_cifara, suma_kvadrata;
9
10    /* Ucitavamo vrednost sa ulaza */
11    printf("Unesite cetvorocifreni broj: ");
12    scanf("%u", &n);
13
14    /* Izdvajamo cifre broja i to redom: j - jedinice, d - desetice,
15       s - stotine i h - hiljade */
16    j = n % 10;
17    d = (n / 10) % 10;
18    s = (n / 100) % 10;
19    h = n / 1000;
20
21    /* Izracunavamo proizvod cifara */
22    proizvod_cifara = j * d * s * h;
23    printf("Proizvod cifara: %d\n", proizvod_cifara);
24
25    /* Izracunavamo razliku sume krajnjih i srednjih cifara */
26    razlika_cifara = (h + j) - (s + d);
27    printf("Razlika sume krajnjih i srednjih: %d\n", razlika_cifara);
28
29    /* Izracunavamo sumu kvadrata cifara */
30    suma_kvadrata = j * j + d * d + s * s + h * h;
31    printf("Suma kvadrata cifara: %d\n", suma_kvadrata);
32
33    /* Odredjujemo broj zapisan istim ciframa ali u obrnutom
34       redosledu */
35    broj_obrnuto = j * 1000 + d * 100 + s * 10 + h;
36    printf("Broj u obrnutom poretku: %u\n", broj_obrnuto);
37
38    /* Odredjujemo broj u kojem su cifra jedinica i cifra stotina
39       zamenile mesta */
40    broj_zamena = h * 1000 + j * 100 + d * 10 + s;
41    printf("Broj sa zamenjenom cifrom jedinica i stotina: %u\n",
42           broj_zamena);
43
44    return 0;
45 }

```

### Rešenje 1.1.15

```

#include <stdio.h>

2 int main()
3 {
4

```

## 1 Uvodni zadaci

```
        unsigned broj, novibroj;
6       unsigned levi, desni, m;

8       printf("Unesite pozitivan prirodan broj: ");
       scanf("%u", &broj);
10      printf("Unesite pozitivan dvocifreni broj:");
       scanf("%u", &m);

12
       /* Na primer, za unete broj 12345 i 67, potrebno je ubaciti 67
14         izmedju cifre hiljade (2) i cifre stotine (3). Rezultat je
           12|67|345. Potrebno je da razdvojimo uneti broj na levi i
16         desni deo: 12 i 345 i izmedju njih umetnemo broj m */

18      levi = broj / 1000;
       desni = broj % 1000;

20
       /* Kada levi deo pomnozimo sa 100 000, dobijamo 1 200 000 Kada m
22         pomnozimo sa 1000, dobijamo 67 000 Dobijene vrednosti saberemo
           sa desnim delom 345 ----- Konacan rezultat: 1 267 345 */
24      novibroj = levi * 100000 + m * 1000 + desni;

26      printf("Novi broj je %u\n", novibroj);

28      return 0;
    }
```

### Rešenje 1.1.16

```
1  #include <stdio.h>

3  int main()
    {
5      /* float - realni tip jednostruke tacnosti */
       float in;
7      float cm;

9      printf("Unesi broj inca: ");

11     /* "%f" - format za unos/ispis float promenljivih */
       scanf("%f", &in);

13
       /* 1 inch = 2.54 cm */
15     cm = in * 2.54;

17     /* "%.2f" - ispis realne promenljive na 4 decimale */
       printf("%.2f in = %.2f cm\n", in, cm);

19     return 0;

21 }
```

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 1.1.21

```
1 #include <stdio.h>
2
3 int main()
4 {
5     float a, b;
6     float obim, površina;
7
8     /* Učitavamo potrebne podatke */
9     printf("Unesite dužine stranica pravougaonika: ");
10    scanf("%f%f", &a, &b);
11
12    /* Obim */
13    obim = 2 * (a + b);
14
15    /* Površina */
16    površina = a * b;
17
18    /* Ispisujemo tražene vrednosti */
19    printf("Obim: %.2f\n", obim);
20    printf("Površina: %.2f\n", površina);
21
22    /* Završavamo sa programom */
23    return 0;
24 }
```

Rešenje 1.1.22

```
1 #include <stdio.h>
2 #include <math.h>
3
4 /* Zaglavlje math.h sadrži deklaracije velikog broja matematičkih
5    funkcija i konstanti. U ovom zadatku se koristi zbog konstante
6    pi (M_PI)
7
8    Ukoliko se koristi i neka funkcija matematičke biblioteke, za
9    prevodjenje je neophodno uključiti opciju -lm npr. gcc primer.c
10    -lm */
11 int main()
12 {
13     float r;
```

## 1 Uvodni zadaci

---

```
float O;  
15 float P;  
printf("Unesite duzinu poluprecnika kruga:");  
17 scanf("%f", &r);  
  
19 O = 2 * r * M_PI;  
P = r * r * M_PI;  
  
21 printf("Obim: %.2f, površina: %.2f\n", O, P);  
23 return 0;  
25 }
```

### Rešenje 1.1.23

```
#include <stdio.h>  
2 #include <math.h>  
  
4 int main()  
{  
6     float a;  
    float P, O;  
  
8     printf("Unesi duzinu stranice trougla:");  
10    scanf("%f", &a);  
  
12    O = 3 * a;  
    P = (a * a * sqrt(3)) / 4;  
  
14    printf("Obim: %.2f\n", O);  
16    printf("Površina: %.2f\n", P);  
  
18    return 0;  
}
```

### Rešenje 1.1.24

```
#include <stdio.h>  
2 #include <math.h>  
  
4 int main()  
{  
6     float a, b, c;  
    float obim, s, površina;  
  
8     /* Ucitavamo potrebne podatke */  
10    printf("Unesite duzine stranica trougla: ");  
    scanf("%f%f%f", &a, &b, &c);  
12 }
```

```

14  /* Obim */
    obim = a + b + c;

16  /* Povrsina - koristice Heronov obrazac */
    s = obim / 2;
18  povrsina = sqrt(s * (s - a) * (s - b) * (s - c));

20  /* Ispisujemo trazene vrednosti */
    printf("Obim: %.2f\n", obim);
22  printf("Povrsina: %.2f\n", povrsina);

24  return 0;
}

```

### Rešenje 1.1.26

```

1  #include<stdio.h>

3  int main()
{
5      int a, b, c;
      float as;

7      printf("Unesite tri cela broja:");
9      scanf("%d%d%d", &a, &b, &c);

11     /* pogresan nacin: as = (a+b+c)/3;

13     Ukoliko podelimo zbir a+b+c sa 3, to ce biti primena
        operatora / na dva cela broja. Na ovaj nacin izracunacemo
15     koliko iznosi a+b+c celobrojno podeljeno sa 3. To znaci da ce
        za unete vrednosti 11, 5 i 4 aritmeticka sredina biti 6.00.
17     Zaista, zbir 11+5+4 iznosi 20, a kada 20 celobrojno podelimo
        sa 3 dobijamo 6. Ovu celobrojnu vrednost dodeljujemo realnoj
19     promenljivoj as, cime se ona konvertuje u 6.000000 i
        ispisujemo je zaokruzenu na dve decimale. Izlaz iz programa bi
21     bio pogresan: 6.00.

23     Da bismo dobili kolicnik prilikom primene operatora / na dva
        cela broja, a ne celobrojno deljenje, jedan argument mora da
25     bude realan broj. Jedan nacin je da umesto sa celobrojnog
        trojkom (3) deljenje izvedemo sa realnom trojkom (3.0): */
27     as = (a + b + c) / 3.0;

29     /* Trazeni kolicnik mozemo dobiti na razne nacine:
        as=1.0*(a+b+c)/3; ili as=(0.0+a+b+c)/3; ili
31     as=((float)(a+b+c))/3; itd. */

33     printf("Aritmeticka sredina unetih brojeva je %.2f\n", as);
        return 0;
35 }

```

### Rešenje 1.1.27

```
1 #include <stdio.h>
3 int main()
4 {
5     unsigned duzina, sirina, visina;
6     unsigned cena;
7     float površina_za_krecenje;
8     float ukupna_cena;
9
10    /* Ucitavamo duzinu, sirinu i visinu sobe */
11    printf("Unesite dimenzije sobe: ");
12    scanf("%u%u%u", &duzina, &sirina, &visina);
13
14    /* Ucitavamo cenu krecenja */
15    printf("Unesite cenu po m2: ");
16    scanf("%u", &cena);
17
18    /* Povrsina za krecenje odgovara površini kvadra - bez poda jer
19       se on ne kreci */
20    površina_za_krecenje = 0.8 * (duzina * sirina +
21                                2 * duzina * visina +
22                                2 * sirina * visina);
23    ukupna_cena = površina_za_krecenje * cena;
24
25    /* Ispisujemo trazene podatke */
26    printf("Moler treba da okreći %.2f m2\n", površina_za_krecenje);
27
28    printf("Cena krecenja je %.2f\n", ukupna_cena);
29
30    /* Završavamo sa programom */
31    return 0;
32 }
```

### Rešenje 1.1.28

```
1 #include <stdio.h>
2 #include <math.h>
3
4 int main()
5 {
6     unsigned x, p;
7     char c;
8     unsigned levo, desno;
9     unsigned novo_x;
```

```

11  /* Ucitavamo potrebne vrednosti. Sa unosom podataka tipa char
13     moramo biti pazljivi i o tome ce vise biti reci u narednim
15     poglavljima kod zadatka za rad sa funkcijama getchar i
17     putchar. Zbog toga cemo ovde za ucitavanje podataka zatraziti
19     da podatke razdvajamo blanko znakovima (a ne znakom za novi
21     red, zarezom ili nekim drugim separatorom). Ovaj zahtev
23     navodimo u format stringu funkcije scanf tako sto
25     specifikatore promenljivih razdvajamo blanko znakovima.

    Ukoliko specifikatore promenljivih u format stringu pisemo
    spojeno, tada ih prilikom unosa mozemo razdvojiti bilo kojim
    karakterom. Zbog toga blanko znakovne u format stringu funkcije
    scanf treba izbegavati i ovo je redak slucaj kada je njihova
    upotreba opravdana.

    Ako zelimo da odstampamo znak ", u format stringu funkcije
    printf navodimo \". */
27  printf("Unesite vrednosti u formatu \"%x p c\": ");
29  scanf("%u %u %c", &x, &p, &c);

31  /* Kada ucitavamo karaktersku promenljivu, njena numericka
33     vrednost je jednaka ASCII kodu unetog karaktera. Na primer,
35     ako karakter '0' ucitamo u promenljivu c, njena numericka
37     vrednost bice 48. Da bismo pretvorili ovu numericku vrednost u
39     numericku vrednost koja odgovara cifri, od nje oduzimamo ASCII
41     kod karakterske konstante '0' koji iznosi upravo 48. */
43  c = c - '0';

45  /* Odredjujemo deo broja koji se nalazi desno od pozicije p */
47  desno = x % (unsigned) pow(10, p);

49  /* Odredjujemo deo broja koji se nalazi levo od pozicije p */
51  levo = x / (unsigned) pow(10, p);

53  /* Odredjujemo novi broj */
55  novo_x =
    levo * (unsigned) pow(10, p + 1) +
    c * (unsigned) pow(10, p) + desno;

    /* Ispisujemo dobijenu vrednost */
    printf("Rezultat je: %u\n", novo_x);

    /* Završavamo sa programom */
    return 0;
}

```

### Rešenje 1.1.29

```

1  #include <stdio.h>

```

## 1 Uvodni zadaci

```
3 int main()
4 {
5     int a, b, rezultata, rezultatb, rezultatc;

6     printf("Unesite dva cela broja:");
7     scanf("%d%d", &a, &b);

8     /* Izraz a!=b ima vrednost 1 ako je ova relacija tacna, a 0 ako
9        je netacna */
10    rezultata = a != b;

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

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

20    printf("a) rezultat=%d\n", rezultata);
21    printf("b) rezultat=%d\n", rezultatb);
22    printf("c) rezultat=%d\n", rezultatc);

23    return 0;
24 }
```

### Rešenje 1.1.30

```
1 #include <stdio.h>

2 int main()
3 {
4     int a, b, max;
5     printf("Unesite dva cela broja:");
6     scanf("%d%d", &a, &b);

7     /* Ternarni operator uslova :? koristi se u sledecem obliku:
8
9        izraz1 ? izraz2 : izraz3;
10
11       Izraz izraz1 se izracunava prvi. Ako je njegova vrednost
12       razlicita od nule (tj. ako ima istinitosnu vrednost tacno),
13       onda se izracunava vrednsot izraza izraz2 i to je vrednost
14       citavog uslovnog izraza. U suprotnom, izracunava se vrednost
15       izraz3 i to je vrednost citavog uslovnog izraza
16
17       */
18 }
```



```
19     max = (a > b) ? a : b;
21     printf("Maksimum je %d\n", max);
23     return 0;
25 }
```

Rešenje [2.1.36](#)      Zadatak se rešava analogno zadatku [2.1.36](#)

Rešenje [1.1.32](#)

```
1  #include <stdio.h>
3  int main()
4  {
5      float a, b, rez;
6      float min, max;
7      printf("Unesite dva realna broja:");
8      scanf("%f%f", &a, &b);
9
10     /* Odredjujemo minimalnu i maksimalnu vrednost unetih brojeva */
11     min = (a < b) ? a : b;
12     max = (a > b) ? a : b;
13
14     /* Racunamo vrednost promenljive rez */
15     rez = (min + 0.5) / (1 + max * max);
16
17     printf("Rezultat je %.2f\n", rez);
18
19     return 0;
20 }
```



## 2

# Kontrola toka

## 2.1 Naredbe grananja

**Zadatak 2.1.1** Napisati program koji za dva uneta cela broja ispisuje njihov minimum.

*Primer 1*

```
|| INTERAKCIJA SA PROGRAMOM:  
| Unesite dva cela broja: 5 18  
| Minimum je 5.
```

*Primer 2*

```
|| INTERAKCIJA SA PROGRAMOM:  
| Unesite dva cela broja: 43 -16  
| Minimum je -16.
```

[Rešenje 2.1.1]

**Zadatak 2.1.2** Napisati program koji za dva uneta cela broja ispisuje njihov maksimum.

*Primer 1*

```
|| INTERAKCIJA SA PROGRAMOM:  
| Unesite dva cela broja: 141 67  
| Maksimum je 141.
```

*Primer 2*

```
|| INTERAKCIJA SA PROGRAMOM:  
| Unesite dva cela broja: -893 -54  
| Maksimum je -54.
```

**Zadatak 2.1.3** Napisati program koji za uneti realan broj ispisuje njegovu apsolutnu vrednost zaokruženu na dve decimale.

## 2 Kontrola toka

---

### Primer 1

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesite jedan realan broj: 7.42  
|| Njegova apsolutna vrednost je: 7.42
```

### Primer 2

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesite jedan realan broj: -562.428  
|| Njegova apsolutna vrednost je: 562.43
```

### Primer 3

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesite jedan realan broj: 0  
|| Njegova apsolutna vrednost je: 0.00
```

### Primer 4

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesite jedan realan broj: 52  
|| Njegova apsolutna vrednost je: 52.00
```

[Rešenje 2.1.3]

**Zadatak 2.1.4** Napisati program koji za uneti ceo broj ispisuje njegovu recipročnu vrednost zaokruženu na četiri decimalne.

### Primer 1

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

### Primer 2

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

### Primer 3

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

### Primer 4

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

[Rešenje 2.1.4]

**Zadatak 2.1.5** Napisati program koji učitava tri cela broja i ispisuje zbir pozitivnih.

### Primer 1

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

### Primer 2

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesite tri cela broja: -15 81 0  
|| Suma unetih pozitivnih brojeva: 81
```

### Primer 3

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

### Primer 4

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesite tri cela broja: 16 2 576  
|| Suma unetih pozitivnih brojeva: 594
```

[Rešenje 2.1.5]

**Zadatak 2.1.6** 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. NAPOMENA: *Pretpostaviti da su cene artikala pozitivni celi brojevi.*

### Primer 1

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

### Primer 2

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

### Primer 3

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

### Primer 4

```
INTERAKCIJA SA PROGRAMOM:
Unesite cene tri artikla: 247 133 126
Cena sa popustom: 381
Usteda: 125
```

[Rešenje 2.1.6]

**Zadatak 2.1.7** Napisati program koji za uneti četvorocifreni broj ispisuje njegovu najveću cifru.

### Primer 1

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

### Primer 2

```
INTERAKCIJA SA PROGRAMOM:
Unesite broj: 238
Greska: Niste uneli cetvorocifren broj!
```

### Primer 3

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

### Primer 4

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

[Rešenje 2.1.7]

**Zadatak 2.1.8** Napisati program koji za uneto vreme (broj sati iz intervala  $[0, 24)$  i broj minuta iz intervala  $[0, 60)$ ) ispisuje koliko je sati i minuta ostalo do ponoći.

### Primer 1

```
INTERAKCIJA SA PROGRAMOM:
Unesite vreme (broj sati u intervalu [0,24),
broj minuta u intervalu [0,60)): 18 19
Do ponoci je ostalo 5 sati i 41 minuta.
```

### Primer 2

```
INTERAKCIJA SA PROGRAMOM:
Unesite vreme (broj sati u intervalu [0,24),
broj minuta u intervalu [0,60)): 23 7
Do ponoci je ostalo 0 sati i 53 minuta.
```

### Primer 3

```
INTERAKCIJA SA PROGRAMOM:
Unesite vreme (broj sati u intervalu [0,24),
broj minuta u intervalu [0,60)): 24 20
Neispravan unos.
```

### Primer 4

```
INTERAKCIJA SA PROGRAMOM:
Unesite vreme (broj sati u intervalu [0,24),
broj minuta u intervalu [0,60)): 14 0
Do ponoci je ostalo 10 sati i 0 minuta.
```

[Rešenje 2.1.8]

**Zadatak 2.1.9** 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, njegov ASCII kod: 48
```

### Primer 2

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

### Primer 3

```
INTERAKCIJA SA PROGRAMOM:
Unesite karakter: A
Uneti karakter: A, njegov ASCII kod: 65
odgovarajuće malo slovo: a, njegov ASCII kod: 97
```

### Primer 4

```
INTERAKCIJA SA PROGRAMOM:
Unesite karakter: v
Uneti karakter: v, njegov ASCII kod: 118
odgovarajuće veliko slovo: V, njegov ASCII kod: 86
```

[Rešenje 2.1.9]

**Zadatak 2.1.10** Napisati program koji za unetih pet karaktera ispisuje koliko je među njima malih slova.

### Primer 1

```
INTERAKCIJA SA PROGRAMOM:
Unesite karaktere: A u E f h
Broj malih slova: 3
```

### Primer 2

```
INTERAKCIJA SA PROGRAMOM:
Unesite karaktere: k L M 9 o
Broj malih slova: 2
```

[Rešenje 2.1.10]

**Zadatak 2.1.11** Program učitava pet karaktera. Napisati koliko se puta pojavilo veliko ili malo slovo a.

*Primer 1*

```
|| INTERAKCIJA SA PROGRAMOM:  
| Unesite karaktere: aBcAe  
| 2
```

*Primer 2*

```
|| INTERAKCIJA SA PROGRAMOM:  
| Unesite karaktere: aa4A_  
| 3
```

*Primer 3*

```
|| INTERAKCIJA SA PROGRAMOM:  
| Unesite karaktere: aAaAa  
| 5
```

*Primer 4*

```
|| INTERAKCIJA SA PROGRAMOM:  
| Unesite karaktere: B6(vV  
| 0
```

[Rešenje 2.1.11]

**Zadatak 2.1.12** Program učitava pet karaktera. Ispisati koliko puta su se pojavile cifre.

*Primer 1*

```
|| INTERAKCIJA SA PROGRAMOM:  
| Unesite karaktere: A1cA3  
| 2
```

*Primer 2*

```
|| INTERAKCIJA SA PROGRAMOM:  
| Unesite karaktere: 2a45_  
| 2
```

*Primer 3*

```
|| INTERAKCIJA SA PROGRAMOM:  
| Unesite karaktere: 43986  
| 5
```

*Primer 4*

```
|| INTERAKCIJA SA PROGRAMOM:  
| Unesite karaktere: B6(vV  
| 1
```

[Rešenje 2.1.12]

**Zadatak 2.1.13** Napisati program koji za unetu godinu ispisuje da li je prestupna.

*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 2.1.13]

**Zadatak 2.1.14** Broj je Armstrongov ako je jednak zbiru kubova svojih cifara. Napisati program koji za dati trocifren broj proverava da li je Armstrongov.

*Primer 1*

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

*Primer 2*

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

*Primer 3*

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesite broj: 84  
|| Greska: Niste uneli trocifren broj!
```

*Primer 4*

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

[Rešenje 2.1.14]

**Zadatak 2.1.15** Napisati program koji ispisuje proizvod parnih cifara unetog četvorocifrenog broja.

*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: -1234  
|| Proizvod parnih cifara: 8
```

*Primer 4*

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesite broj: 288  
|| Broj nije četvorocifren!
```

[Rešenje 2.1.15]

**Zadatak 2.1.16** Napisati program koji učitava četvorocifreni broj i ispisuje broj koji se dobija kada se unetom broju razmene najmanja i najveća cifra. NAPOMENA: *U slučaju da se najmanja ili najveća cifra pojavljuju na više pozicija, uzeti prvo pojavljivanje.*

*Primer 1*

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

*Primer 2*

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesite broj: 247  
|| Broj nije četvorocifren!
```



### Primer 3

```
INTERAKCIJA SA PROGRAMOM:
Unesite broj: 1192
9112
```

### Primer 4

```
INTERAKCIJA SA PROGRAMOM:
Unesite broj: -4239
-4932
```

[Rešenje 2.1.16]

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

[Rešenje 2.1.17]

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

**Zadatak 2.1.19** Napisati program za rad sa intervalima. Za dva intervala realne prave  $[a_1, b_1]$  i  $[a_2, b_2]$ , program treba da odredi:

- dužinu zajedničkog dela ta dva intervala
- najveći interval sadržan u datim intervalima (presek), a ako on ne postoji dati odgovarajuću poruku.
- dužinu realne prave koju pokrivaju ta dva intervala
- najmanji interval koji sadrži date intervale.

### Primer 1

```
INTERAKCIJA SA PROGRAMOM:
Unesite redom a1, b1, a2 i b2: 2 9 4 11
Duzina zajednickog dela: 5
Presek intervala: [4,9]
Zajednicka duzina intervala: 9
Najmanji interval: [2, 11]
```

### Primer 2

```
INTERAKCIJA SA PROGRAMOM:
Unesite redom a1, b1, a2 i b2: 1 2 10 13
Duzina zajednickog dela: 0
Presek intervala: prazan
Zajednicka duzina intervala: 4
Najmanji interval: [1, 13]
```

**Zadatak 2.1.20** Napisati program koji za uneti ceo broj  $x$  ispisuje njegov znak, tj da li je broj jednak nuli, manji od nule ili veći od nule.

### Primer 1

```
INTERAKCIJA SA PROGRAMOM:
Unesite jedan ceo broj: 17
Broj je veci od nule.
```

### Primer 2

```
INTERAKCIJA SA PROGRAMOM:
Unesite jedan ceo broj: 0
Broj je jednak nuli.
```

### Primer 3

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesite jedan ceo broj: -586  
|| Broj je manji od nule.
```

### Primer 4

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesite jedan ceo broj: 62  
|| Broj je veci od nule.
```

[Rešenje 2.1.20]

**Zadatak 2.1.21** 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 2.1.21]

**Zadatak 2.1.22** Napisati program koji za uneti četvorocifreni broj proverava da li su njegove cifre uređene rastuće, opadajuće ili nisu uređene i štampa odgovarajuću poruku.

### Primer 1

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesite cetvorocifreni broj: 1389  
|| Cifre su uredjene neopadajuce.
```

### Primer 2

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

### Primer 3

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

### Primer 4

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesite cetvorocifreni broj: 88  
|| Uneti broj nije cetvorocifren.
```

[Rešenje 2.1.22]

**Zadatak 2.1.23** Napisati program koji učitava karakter i:

- a) ako je  $c$  malo slovo, ispisuje odgovarajuće veliko
- b) ako je  $c$  veliko slovo, ispisuje odgovarajuće malo

- c) ako je  $c$  cifra, ispisuje poruku *cifra*  
d) u ostalim slučajevima, ispisuje karakter  $c$  između dve zvezdice.

### Primer 1

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite karakter: K
|| k
```

### Primer 2

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite karakter: 8
|| cifra
```

### Primer 3

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite karakter: >
|| **
```

### Primer 4

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite karakter: o
|| 0
```

[Rešenje 2.1.23]

**Zadatak 2.1.24** 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.

### 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.
```

[Rešenje 2.1.24]

**Zadatak 2.1.25** Data je funkcija  $f(x) = 2 \cdot \cos(x) - x^3$ . Napisati program koji za učitane vrednosti realne promenljive  $x$  i vrednosti celobrojne promenljive  $k$  koje može biti 1, 2 ili 3 izračunava vrednost funkcije  $F(k, x) = f(f(f(\dots f(x))))$  gde je funkcija  $f$  primenjena  $k$ -puta i ispisuje je zaokruženu na dve decimalne. U slučaju neispravnog ulaza, odštampati 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 2.1.25]

**Zadatak 2.1.26** Napisati program koji za uneti redni broj dana u nedelji ispisuje ime odgovarajućeg dana. U slučaju pogrešnog unosa ispisati odgovarajuću poruku.

*Primer 1*

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

*Primer 2*

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

*Primer 3*

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesite broj: 8  
|| Greska: nedozvoljeni unos!
```

*Primer 4*

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

[Rešenje 2.1.26]

**Zadatak 2.1.27** Napisati program koji za uneti karakter ispituje da li je samoglasnik.

*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 2.1.27]

**Zadatak 2.1.28** 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 pogrešnog unosa ispisati odgovarajuću poruku.

*Primer 1*

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesite operator i dva cela broja: - 8 11  
|| Rezultat je: -3
```

*Primer 2*

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesite operator i dva cela broja: / 14 0  
|| Greska: deljenje nulom nije dozvoljeno!
```

### Primer 3

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesite operator i dva cela broja: ? 5 7  
|| Greska: nepoznat operator!
```

### Primer 4

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesite operator i dva cela broja: / 19 5  
|| Rezultat je: 3
```

[Rešenje [2.1.28](#)]

**Zadatak 2.1.29** Napisati program koji za uneti dan i mesec ispisuje godišnje doba kojem pripadaju. NAPOMENA: *Podrazumevati da je unos korektan.*

### Primer 1

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesite dan i mesec: 14 10  
|| jesen
```

### Primer 2

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesite dan i mesec: 2 8  
|| leto
```

### Primer 3

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesite dan i mesec: 27 2  
|| zima
```

### Primer 4

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesite dan i mesec: 19 5  
|| prolece
```

[Rešenje [2.1.29](#)]

**Zadatak 2.1.30** Napisati program koji za unetu godinu i mesec ispisuje naziv meseca kao i koliko dana ima u tom mesecu te godine.

[Rešenje [2.1.30](#)]

**Zadatak 2.1.31** Napisati program koji za uneti datum u formatu *dan.mesec.godina*. proverava da li je korektan.

### Primer 1

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

### Primer 2

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

[Rešenje [2.1.31](#)]

**Zadatak 2.1.32** 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.
```

[Rešenje 2.1.32]

**Zadatak 2.1.33** 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.
```

**Zadatak 2.1.34** Korisnik unosi tri cela broja:  $P$ ,  $Q$  i  $R$ . Nakon toga unosi i dva karaktera,  $op1$  i  $op2$ . Ovi karakteri predstavljaju operacije nad unetim brojevima i imaju naredno značenje:

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

Program treba da sračuna vrednost izraza  $P \text{ op1 } Q \text{ op2 } R$  i da ga ispiše.

### Primer 1

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite tri cela broja: 0 1 2
|| Unesite dva karaktera cela broja: k m
|| 1
```

### Primer 2

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite tri cela broja: -3 -1 0
|| Unesite dva karaktera cela broja: d k
|| 0
```

\* **Zadatak 2.1.35** Program učitava jedan karakter i osam realnih brojeva koji predstavljaju koordinate četiri tačke:  $A(x_1, y_1)$ ,  $B(x_2, y_2)$ ,  $C(x_3, y_3)$ ,  $D(x_4, y_4)$ . Na osnovu unetog karaktera ispisuje se odgovarajuća poruka na standardni izlaz:

- ukoliko je uneti karakter  $k$  - proverava da li su date tačke temena pravougona čije su stranice paralelne koordinatnim osama i u slučaju da jesu, ispisuje vrednost obima datog pravougona. Možemo podrazumevati da će

korisnik koordinate tačaka unosi redom  $A, B, C, D$ , pri čemu  $ABCD$  opisuje pravougaonik čije su stranice  $AB, BC, CD, DA$ , a dijagonale  $AC$  i  $BD$ . Na primer, tačke  $(1, 1), (2, 1), (2, 2), (1, 2)$  čine pravougaonik čije su stranice paralelne koordinatnim osama i čiji je obim 4 a tačke  $(1, 1), (2, 2), (3, 3), (4, 4)$  ne čine pravougaonik.

- ukoliko je uneti karakter  $h$  - proverava da li su unete tačke kolinearne i ukoliko jesu, ispisuje jednačinu prave kojoj pripadaju. Na primer, tačke  $(1, 2), (2, 3), (3, 4), (4, 5)$  su kolinearne i pripadaju pravoj  $y = x + 1$ , tačke  $(1, 1), (1, 2), (1, 3), (1, 4)$  su kolinearne i pripadaju pravoj  $x = 1$ , a tačke  $(1, 1), (2, 1), (2, 2), (1, 2)$  nisu kolinearne.
- ukoliko je uneti karakter  $j$  - Kramerovim pravilom proverava da li je sistem jednačina  $x_1 * p + x_2 * q = x_4 - x_3, y_1 * p + y_2 * q = y_4 - y_3$  određen, neodređen ili nema rešenja, i u slučaju da je određen ispisuje rešenja.

[Rešenje 2.1.35]

**Zadatak 2.1.36** Polje šahovske table se definiše parom prirodnih brojeva ne većih od 8: prvi se odnosi na red, drugi na kolonu. Ako su dati takvi parovi, napisati program koji proverava:

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

## 2.2 Rešenja

### Rešenje 2.1.1

```

1 #include <stdio.h>
3 int main()
4 {
5     int a, b, min;
6     printf("Unesite dva cela broja: ");
7     scanf("%d%d", &a, &b);
8
9     /* Promenljiva min dobija vrednost promenljive a. */
10    min = a;

```

## 2 Kontrola toka

---

```
11  /*
13     Ako je b<a, promenljiva min ce promeniti vrednost tj. bice joj
    dodeljena vrednost promenljive b. U suprotnom, vrednost ostaje
15     ista. */
17     if (b < a)
18         min = b;
19
20     printf("Minimum je %d\n", min);
21
22     return 0;
23 }
```

Rešenje 2.1.2      Rešenje je analogno rešenju broj 2.1.1.

### Rešenje 2.1.3

```
1  #include<stdio.h>
2
3  int main()
4  {
5      float x;
6      float apsolutno_x;
7
8      printf("Unesite jedan realan broj:");
9      scanf("%f", &x);
10
11     apsolutno_x = x;
12     if (x < 0)
13         apsolutno_x = -x;
14
15     printf("Njegova apsolutna vrednost je %.2f\n", apsolutno_x);
16
17     /* 2. nacin, pomocu funkcije fabs za koju je neophodno ukljuciti
        zaglavlje math.h: apsolutno_x=fabs(x); */
18     return 0;
19 }
```

### Rešenje 2.1.4

```
1  #include <stdio.h>
2
3  int main()
4  {
5      int x;
6      float rx;
7  }
```



```

9  printf("Unesite jedan ceo broj:");
   scanf("%d", &x);

11 /*
   Obratiti paznju: x==0 - relacija jednakosti (da li je
13  promenljiva x jednaka nuli) x=0 - naredba dodele (promenljiva
   x dobija vrednost nula) */

15 /*
17  Proveravamo da li je uneti broj jednak nuli. Ako jeste,
   prekidamo sa daljim izvršavanjem programa navodjenjem naredbe
19  return. Argument -1 u naredbi return oznacava da program nije
   uspesno završen */
21  if (x == 0) {
23      printf("Nedozvoljeno deljenje nulom\n");
       return -1;
   }

25 /*
27  Primenom operatora / na argumente 1 i x dobijamo rezultat
   celobrojnog deljenja ovih argumenata. Da bismo dobili
29  kolicnik, koji je realna vrednost, neophodno je da jedan od
   argumenata zapisemo kao realnu vrednost, npr celobrojnu
31  vrednost 1 zapisemo kao realnu vrednost 1.0. Ovakav postupak
   se naziva implicitna konverzija. */

33
   rx = 1.0 / x;
35  printf("Recipročna vrednost unetog broja: %.4f\n", rx);

37  return 0;
}

```

### Rešenje 2.1.5

```

1  #include<stdio.h>

3  int main()
   {
5      int a, b, c;
       int s;
7      printf("Unesite tri cela broja:");
       scanf("%d%d%d", &a, &b, &c);

9
11     /* inicijalizujemo promenljivu s na nulu */
       s = 0;

13
15     /*
       U naredbi dodele s=s+a vrednost izraza sa desne strane znaka
       jednakosti dodeljujemo promenljivoj sa leve strane znaka
       jednakosti. Staru vrednost promenljive s sabereemo sa vrednoscu
17     promenljive a i dobijenu vrednost upisemo u promenljivu s. */

```

```
19  if (a > 0)
    s = s + a;
21
    /* s+=b je skraceni zapis za s=s+b */
23
    if (b > 0)
25        s += b;
27
    if (c > 0)
        s += c;
29
    printf("Suma unetih pozitivnih brojeva: %d\n", s);
31    return 0;
}
```

### Rešenje 2.1.6

```
#include <stdio.h>
2
int main()
4 {
    unsigned a, b, c;
    unsigned min;
    unsigned cena_bez_popusta, cena_sa_popustom;
8
    printf("Unesite cene tri artikla:");
10    scanf("%u%u%u", &a, &b, &c);
12
    /*
        Racunamo minimum tri broja. Dodeljujemo promenljivoj min
14        vrednost prvog broja. */
    min = a;
16
    /*
        Ako je drugi broj manji od minimuma, to znaci da promenljiva
18        min ne sadrzi najmanji broj. Dodeljujemo joj vrednost drugog
        broja. */
20    if (min > b)
22        min = b;
24
    /*
        Ako je treci broj manji od minimuma, to znaci da promenljiva
26        min ne sadrzi najmanji broj. Dodeljujemo joj vrednost treceg
        broja. */
28    if (min > c)
        min = c;
30
    cena_bez_popusta = a + b + c;
32    cena_sa_popustom = cena_bez_popusta - min + 1;
```

```
34 printf("Cena sa popustom: %u\nUsteda: %u\n",
35        cena_sa_popustom, cena_bez_popusta - cena_sa_popustom);
36
37 return 0;
38 }
```

### Rešenje 2.1.7

```
1  #include <stdio.h>
2
3  int main()
4  {
5      int n, j, d, s, h, max;
6
7      /* Ucitavamo broj */
8      printf("Unesite cetvorocifreni broj: ");
9      scanf("%d", &n);
10
11
12     /*
13      Za slucaj da je broj negativan, uzimamo apsolutnu vrednost
14      unetog broja */
15     n = abs(n);
16     /*
17      Ako uneti broj nije cetvorocifren, ispisujemo poruku o gresci
18      i prekidamo izvršavanje programa. */
19     if (n < 1000 || n > 9999) {
20         printf("Greska: Niste uneli cetvorocifren broj!\n");
21         return -1;
22     }
23
24     /*
25      Ako je broj cetvorocifren, izdvajamo cifre broja: j -jedinice,
26      d - desetice, s - stotine i h - hiljade */
27     j = n % 10;
28     d = (n / 10) % 10;
29     s = (n / 100) % 10;
30     h = n / 1000;
31
32     /* Odredjujemo maksimalnu cifru */
33     max = j;
34
35     if (d > max)
36         max = d;
37
38     if (s > max)
39         max = s;
40
41     if (h > max)
42         max = h;
43 }
```

## 2 Kontrola toka

```
45  /* II nacin: if(j>d && j>s && j>h) max=j; if(d>j && d>s && d>h)
    max=d; if(s>j && s>d && s>h) max=s; if(h>j && h>d && h>s)
    max=h; */
47
49  /* Ispisujemo rezultat */
    printf("Najveca cifra je: %d\n", max);
51  return 0;
}
```

### Rešenje 2.1.8

```
#include<stdio.h>
2
int main()
4 {
    int sati;
    int minuti;
    int preostali_sati;
    int preostali_minuti;
8
10  /*
    Ukoliko naredbu printf zelimo da napisemo u dva reda, i tom
12  prilikom prekidamo deo pod navodnicima, to mozemo uraditi
    navodjenjem navodnika na kraju prvog i na pocetku narednog
14  reda: */

16  printf("Unesite vreme (broj sati u intervalu [0,24),\n"
    "broj minuta u intervalu [0,60)):");
18  scanf("%d%d", &sati, &minuti);

20  /*
    U slucaju da je unos neispravan, ispisujemo poruku o gresci i
22  prekidamo dalje izvorsavanje programa.

    Uslov u if naredbi je disjunkcija (operator ||) sastavljena od
24  4 disjunkata. Svaki od njih je izraz sa relacionim operatorom
    i ima vrednost 1 ako je izraz tacan i 0 u suprotnom. Da bi
26  disjunkcija bila tacna, bar jedan od disjunkata mora da bude
    tacan. Zbog lenjog izracunavanja, vrednost disjunkata ce biti
28  racunata do vrednosti prvog disjunkta koji je tacan. To je
    znak da je uslov u if naredbi ispunjen i nema potrebe racunati
30  vrednosti drugih disjunkata. */

32  if (sati > 24 || sati < 0 || minuti > 59 || minuti < 0) {
34      printf("Neispravan unos.\n");
    return -1;
36  }

38  preostali_sati = 24 - sati - 1;
    preostali_minuti = 60 - minuti;
```

```

40     if (preostali_minuti == 60) {
42         preostali_sati++;
43         preostali_minuti = 0;
44     }
45
46     /*
47      Ukoliko naredbu printf zelimo da napisemo u dva reda i pritom
48      ne prekidamo deo pod navodnicima, to mozemo uraditi bez
49      navodjenja dodatnih karaktera: */
50     printf("Do ponoci je ostalo %d sati i %d minuta\n",
51           preostali_sati, preostali_minuti);
52
53     return 0;
54 }

```

### Rešenje 2.1.9

```

#include <stdio.h>
2
int main()
4 {
    char c;
6     printf("Unesite jedan karakter:");
    scanf("%c", &c);
8
    printf("Uneti karakter: %c, njegov ASCII kod: %d\n", c, c);
10
    /*
12     Razlika izmedju ASCII koda svakog malog i odgovarajuceg
    velikog slova je konstanta koja se moze sracunati izrazom
14     'a'-'A' (i iznosi 32) */
16
    if (c >= 'A' && c <= 'Z')
        printf("odgovarajuce malo slovo: %c, njegov ASCII kod: ",
18              "%d\n", c + ('a' - 'A'), c + ('a' - 'A'));
20
    if (c >= 'a' && c <= 'z')
        printf("odgovarajuce veliko slovo: %c, njegov ASCII kod: ",
22              "%d\n", c - ('a' - 'A'), c - ('a' - 'A'));
24
    return 0;
}

```

### Rešenje 2.1.10

```

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

```

```
{
5
   char c1, c2, c3, c4, c5;
7   int broj_malih_slova = 0;

9
   /* Citamo karaktere */
11  printf("Unesite karaktere: ");
   scanf("%c %c %c %c %c", &c1, &c2, &c3, &c4, &c5);

13
   /* Proveravamo da li je prvi karakter malo slovo */
15  if (c1 >= 'a' && c1 <= 'z') {
       /* I ako jeste, uvecavamo broj malih slova */
17     broj_malih_slova++;
   }

19
   /* Proveravamo da li je drugi karakter malo slovo */
21  if (c2 >= 'a' && c2 <= 'z') {
       /* I ako jeste, uvecavamo broj malih slova */
23     broj_malih_slova++;
   }

25
   /* Proveravamo da li je treci karakter malo slovo */
27  if (c3 >= 'a' && c3 <= 'z') {
       /* I ako jeste, uvecavamo broj malih slova */
29     broj_malih_slova++;
   }

31
   /* Proveravamo da li je cetvrti karakter malo slovo */
33  if (c4 >= 'a' && c4 <= 'z') {
       /* I ako jeste, uvecavamo broj malih slova */
35     broj_malih_slova++;
   }

37
   /* Proveravamo da li je peti karakter malo slovo */
39  if (c5 >= 'a' && c5 <= 'z') {
       /* I ako jeste, uvecavamo broj malih slova */
41     broj_malih_slova++;
   }

43
   /* Ispisujemo rezultat */
45  printf("Broj malih slova: %d\n", broj_malih_slova);

47  return 0;
}
```

### Rešenje 2.1.11

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

```

5  int main()
6  {
7      /*
8       Broj pojavljivanja slova a i A se inicijalizuje na 0 */
9      int br_a = 0;
10
11     /*
12      Funkcija getchar ucitava jedan karakter. Njena povratna
13      vrednost je ASCII kod ucitanog karaktera.
14
15      Funkcija tolower za dati karakter vraca: - odgovarajuce malo
16      slovo, ako je dati karakter veliko slovo - taj isti karakter,
17      u suprotnom Ova funkcija je definisana u biblioteci ctype.h
18
19      U slucaju da je uslov ispunjen, uvecavamo brojac br_a za jedan
20      pomocu operatora inkrementacije ++ */
21     if (tolower(getchar()) == 'a')
22         br_a++;
23     if (tolower(getchar()) == 'a')
24         br_a++;
25     if (tolower(getchar()) == 'a')
26         br_a++;
27     if (tolower(getchar()) == 'a')
28         br_a++;
29     if (tolower(getchar()) == 'a')
30         br_a++;
31
32     printf("%d\n", br_a);
33
34     return 0;
35 }

```

### Rešenje 2.1.12

```

1  #include <stdio.h>
2  #include <ctype.h>
3
4  int main()
5  {
6      int br_cif = 0;
7
8      /*
9       Funkcija isdigit vraca 1 ako je dati karakter cifra i 0 u
10      suprotnom. Nalazi se u biblioteci ctype.h. */
11     if (isdigit(getchar()))
12         br_cif++;
13     if (isdigit(getchar()))
14         br_cif++;
15     if (isdigit(getchar()))
16         br_cif++;
17     if (isdigit(getchar()))

```

## 2 Kontrola toka

---

```
    br_cif++;
19  if (isdigit(getchar()))
    br_cif++;
21
    printf("%d\n", br_cif);
23
    return 0;
25 }
```

### Rešenje 2.1.13

```
1  #include <stdio.h>
3  int main()
{
5      int x;
    printf("Unesite godinu:");
7      scanf("%d", &x);

9      if ((x % 4 == 0 && x % 100 != 0) || x % 400 == 0)
        printf("Godina je prestupna\n");
11     else
        printf("Godina nije prestupna\n");
13
    return 0;
15 }
```

### Rešenje 2.1.14

```
1  #include <stdio.h>
   #include <stdlib.h>           /* abs */
3
4  int main()
5  {
6      int n, j, d, s;
7
8      /* Ucitavamo broj */
9      printf("Unesite broj: ");
10     scanf("%d", &n);
11
12     /*
13      * Uzimamo apsolutnu vrednost broja za slucaj da je uneti broj
14      * negativan */
15     n = abs(n);
16
17     /*
18      * Ako broj nije trocifren, izdajemo poruku o gresci i prekidamo
19      * dalje izvršavanje programa */
20     if (n < 100 || n > 999) {
```



```

21     printf("Greska: Niste uneli trocifren broj!\n");
    return -1;
23 }

25 /*
    Izdvajamo cifre broja: j -jedinice, d - desetice, s - stotine */
27 j = n % 10;
    d = (n / 10) % 10;
29 s = n / 100;

31 /* Proveravamo da li je broj Armstrongov */
    if (n == j * j * j + d * d * d + s * s * s)
33     printf("Broj je Armstrongov.\n");
    else
35     printf("Broj nije Armstrongov.\n");

37     return 0;
}

```

### Rešenje 2.1.15

```

1  #include <stdio.h>

3  int main()
{
5     int n, j, d, s, h;
    int broj_parnih, proizvod_parnih;

7     printf("Unesite cetvorocifreni broj: ");
9     scanf("%d", &n);

11    n = abs(n);

13    if (n < 1000 || n > 9999) {
        printf("Broj nije cetvorocifren.\n");
15        return -1;
    }

17    /* Izdvajamo cifre broja: j -jedinice, d - desetice, s - stotine
        i h - hiljade */
19    j = n % 10;
    d = (n / 10) % 10;
21    s = (n / 100) % 10;
    h = n / 1000;

23    /* Inicijalizujemo broj parnih cifara na 0 */
    broj_parnih = 0;
27    /* Postavljamo proizvod parnih cifara na 1 (neutral za mnozenje) */
    proizvod_parnih = 1;

29    /* Proveravamo da li je cifra jedinica parna */

```

## 2 Kontrola toka

```
31  if (j % 2 == 0) {
    proizvod_parnih = proizvod_parnih * j;
33  broj_parnih++;
    }

35

/* Proveravamo da li je cifra desetica parna */
37  if (d % 2 == 0) {
    proizvod_parnih = proizvod_parnih * d;
39  broj_parnih++;
    }

41

/* Proveravamo da li je cifra stotina parna */
43  if (s % 2 == 0) {
    proizvod_parnih = proizvod_parnih * s;
45  broj_parnih++;
    }

47

/* Proveravamo da li je cifra hiljada parna */
49  if (h % 2 == 0) {
    proizvod_parnih = proizvod_parnih * h;
51  broj_parnih++;
    }

53

/* Proveravamo da li u zapisu broja ima parnih cifara i
55  ispisujemo rezultat */
    if (broj_parnih == 0) {
57  printf("Nema parnih cifara.\n");
    } else {
59  printf("Proizvod parnih cifara: %d\n", proizvod_parnih);
    }

61  return 0;
63 }
```

### Rešenje 2.1.16

```
1  #include <stdio.h>

3  int main()
{
5  int broj;
  scanf("%d", &broj);

7

  // Da bismo lakse odredili da li je cetvorocifren
9  int absBroj = broj < 0 ? -broj : broj;
  if (absBroj <= 999 || absBroj >= 10000) {
11  printf("Broj nije cetvorocifren.");
    return -1;
13  }

15  int a = absBroj % 10;
```

```
17  int b = (absBroj / 10) % 10;
18  int c = (absBroj / 100) % 10;
19  int d = absBroj / 1000;

20  int max = a, min = a;
21  // cuvamo i stepen da bismo lakse zamenili cifre
22  /* Ideja: 4179, mesta menjamo tako sto oduzmemo 9 i dodamo 1, ^ ^
23     odnosno oduzemo 100 i dodamo 900 */
24  int stepenMax = 1, stepenMin = 1;

25
26  if (b > max) {
27      max = b;
28      stepenMax = 10;
29  }
30  if (b < min) {
31      min = b;
32      stepenMin = 10;
33  }

34  if (c > max) {
35      max = c;
36      stepenMax = 100;
37  }
38  if (c < min) {
39      min = c;
40      stepenMin = 100;
41  }

42  if (d > max) {
43      max = d;
44      stepenMax = 1000;
45  }
46  if (d < min) {
47      min = d;
48      stepenMin = 1000;
49  }

50
51  int rez;
52
53  /* Ideja: 4179, mesta menjamo tako sto oduzmemo 9 i dodamo 1, ^ ^
54     odnosno oduzemo 100 i dodamo 900 */
55
56  if (broj > 0)
57      rez = broj - max * stepenMax + min * stepenMax
58          - min * stepenMin + max * stepenMin;
59  else
60      rez = broj + max * stepenMax - min * stepenMax
61          + min * stepenMin - max * stepenMin;
62
63  printf("%d\n", rez);
64
65  return 0;
```

```
}

```

### Rešenje 2.1.20

```
1 #include <stdio.h>
3 int main()
4 {
5     int x;
6     printf("Unesite jedan ceo broj:");
7     scanf("%d", &x);
9     if (x == 0)
10        printf("Broj je jednak nuli\n");
11    else if (x < 0)
12        printf("Broj je manji od nule\n");
13    else
14        printf("Broj je veci od nule\n");
15    return 0;
17 }
```

### Rešenje 2.1.21

```
1 #include <stdio.h>
2 #include <math.h>
3
4 int main()
5 {
6     float a, b, c;
7     float D;
8     float x1, x2;
9     printf("Unesite koeficijente A, B i C:");
10    scanf("%f%f%f", &a, &b, &c);
11
12    /* Proveravamo da li je kvadratna jednačina korektno zadata. */
13    if (a == 0)
14        if (b == 0)
15            /* slucaj a==0 && b==0 && c==0 */
16            if (c == 0)
17                printf("Jednacina ima beskonacno mnogo resenja\n");
18    /* slucaj a==0 && b==0 && c!=0 */
19    else
20        printf("Jednacina nema resenja\n");
21    /* slucaj a!=0 && b!=0 */
22    else {
23        x1 = -c / b;
24        printf("Jednacina ima jedinstveno realno resenje %.2f\n", x1);
25    }
```

```

27  /* slucaj a!=0 */
28  else {
29      D = b * b - 4 * a * c;
30      if (D < 0)
31          printf("Jednacina nema realnih resenja\n");
32      else if (D > 0) {
33          /* funkcija sqrt nalazi se u biblioteci math.h (prevodjenje
34             sa -lm opcijom) */
35          x1 = (-b + sqrt(D)) / (2 * a);
36          x2 = (-b - sqrt(D)) / (2 * a);
37          printf("Jednacina ima dva razlicita realna resenja %.2f ",
38                 "i %.2f\n", x1, x2);
39      } else {
40          x1 = (-b) / (2 * a);
41          printf("Jednacina ima jedinstveno realno resenje %.2f\n", x1);
42      }
43  }
44  return 0;
45 }

```

### Rešenje 2.1.22

```

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

4 int main()
5 {
6     int x;
7     char c1;
8     char c10;
9     char c100;
10    char c1000;

12    printf("Unesi jedan cetvorocifreni broj:");
13    scanf("%d", &x);

14
15    /* Uzimamo apsolutnu vrednost unetog broja kako u slucaju da je
16       negativan ne bismo za cifre dobili negativne brojeve. Funkcija
17       abs nalazi se u zaglavlju stdlib.h */
18    x = abs(x);

20    if (x < 1000 || x > 9999) {
21        printf("Uneti broj nije cetvorocifren\n");
22        return -1;
23    }

24
25    /* Izdvajamo cifre broja. */
26    c1 = x % 10;
27    c10 = (x / 10) % 10;
28    c100 = (x / 100) % 10;

```

## 2 Kontrola toka

---

```
    c1000 = (x / 1000) % 10;
30
    if (c1000 <= c100 && c100 <= c10 && c10 <= c1)
32        printf("Cifre su uredjene neopadajuće \n");
    else if (c1000 >= c100 && c100 >= c10 && c10 >= c1)
34        printf("Cifre su uredjene nerastuće \n");
    else
36        printf("Cifre nisu uredjene\n");
38    return 0;
}
```

### Rešenje 2.1.23

```
1  #include <stdio.h>
2
3  int main()
4  {
5      char c;
6
7      printf("Unesite karakter: ");
8      scanf("%c", &c);
9
10     if (c >= 'a' && c <= 'z')
11         printf("%c\n", c - 'a' + 'A');
12     else if (c >= 'A' && c <= 'Z')
13         printf("%c\n", c - 'A' + 'a');
14     else if (c >= '0' && c <= '9')
15         printf("cifra\n");
16     /* Ako nijedan od prethodnih uslova nije ispunjen, bice izvršena
17        naredba u else grani */
18     else
19         printf("%c*\n", c);
21     return 0;
}
```

### Rešenje 2.1.24

```
1  #include <stdio.h>
2
3  int main()
4  {
5      int k, n, broj;
6
7      printf("Unesite k: ");
8      scanf("%d", &k);
9
10     if (k < 10) {
```

```

11      /* Trazi se jednocifren broj */
      printf("Na %d-toj poziciji je broj %d.\n", k, k);
13  } else
      /* Trazi se dvocifreni broj */
15  if (k >= 10 && k <= 189) {

17      /* Odredjujemo broj dvocifrenih brojeva koji se mogu zapisati
         pomocu k cifara */

19
21      if (k % 2 != 0) {
          /* Ako je k neparan broj, zapisan je ceo broj dvocifrenih
             brojeva

23              9 oduzimamo jer je 9 broj cifara potrebnih za zapis
25              jednocifrenih brojeva */
          n = (k - 9) / 2;

27          /* Broj o kojem se radi je */
29          broj = 9 + n;

31          /* Ujedno, za neparno k se trazi cifra jedinica izdvojenog
             broja */
33          printf("Na %d-toj poziciji je broj %d.\n", k, broj % 10);

35      } else {
          /* Ako je k paran broj, zapisan je ceo broj dvocifrenih
             brojeva i zapoceto je sa zapisom sledeceg

37              9 oduzimamo jer je 9 broj cifara potrebnih za zapis
39              jednocifrenih brojeva */
          n = (k - 9) / 2 + 1;

41          /* Broj o kojem se radi je */
43          broj = 9 + n;

45          /* Ujedno, za parno k se trazi cifra desetica izdvojenog
             broja */
47          printf("Na %d-toj poziciji je broj %d.\n", k, broj / 10);

49      }
51  } else {
      printf("Greska: Nedozvoljena vrednost broja k!\n");
53  }

55  return 0;
}

```

### Rešenje 2.1.25

```

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

```

## 2 Kontrola toka

---

```
3
4
5 int main()
6 {
7     float x;
8     int k;
9     float F;
10
11     printf("Unesite redom x i k: ");
12     scanf("%f %d", &x, &k);
13
14     /* Proveravaju se vrednosti za k */
15     if (k < 1 || k > 3) {
16         printf("Greska: nedozvoljena vrednost za k!\n");
17         return 0;
18     }
19     printf("F(%f,%d)=", x, k);
20
21     /* Analiziraju se moguci slucajevi */
22     if (k == 1) {
23         F = 2 * cos(x) - x * x * x;
24     } else {
25         if (k == 2) {
26             x = 2 * cos(x) - x * x * x;
27             F = 2 * cos(x) - x * x * x;
28         } else {
29             x = 2 * cos(x) - x * x * x;
30             x = 2 * cos(x) - x * x * x;
31             F = 2 * cos(x) - x * x * x;
32         }
33     }
34
35     /* Ispisuje se rezultat */
36     printf("%f\n", F);
37
38     return 0;
39 }
```

### Rešenje 2.1.26

```
1 #include <stdio.h>
2
3 int main()
4 {
5     int dan;
6
7     printf("Unesite broj: ");
8     scanf("%d", &dan);
9
10    switch (dan) {
11    case 1:
12        printf("ponedeljak\n");
13    }
```



```
13     break;
14 case 2:
15     printf("utorak\n");
16     break;
17 case 3:
18     printf("sreda\n");
19     break;
20 case 4:
21     printf("cetvrtak\n");
22     break;
23 case 5:
24     printf("petak\n");
25     break;
26 case 6:
27     printf("subota\n");
28     break;
29 case 7:
30     printf("nedelja\n");
31     break;
32 default:
33     printf("Greska: nedozvoljeni unos!\n");
34 }
35
36 return 0;
37 }
```

### Rešenje 2.1.27

```
1  #include <stdio.h>
2
3  int main()
4  {
5      char c;
6      printf("Unesite jedan karakter:");
7      scanf("%c", &c);
8
9      /* Da bi se utvrdilo da li je karakter samoglasnik, neophodno je
10         proveriti da li odgovara nekom od sledecih karaktera:
11         A,E,I,O,U,a,e,i,o,u */
12     switch (c) {
13     case 'A':
14     case 'E':
15     case 'I':
16     case 'O':
17     case 'U':
18     case 'a':
19     case 'e':
20     case 'i':
21     case 'o':
22     case 'u':
23         printf("Uneti karakter je samoglasnik\n");
```

```
        break;
25    default:
        printf("Uneti karakter nije samoglasnik\n");
27        break;
    }
29
    return 0;
31 }
```

### Rešenje 2.1.28

```
1  #include <stdio.h>
3  int main()
4  {
5      char op;
6      int x, y;
7
8      printf("Unesite operator i dva cela broja: ");
9      scanf("%c %d %d", &op, &x, &y);
10
11     switch (op) {
12     case '+':
13         printf("Rezultat je: %d\n", x + y);
14         break;
15     case '-':
16         printf("Rezultat je: %d\n", x - y);
17         break;
18     case '*':
19         printf("Rezultat je: %d\n", x * y);
20         break;
21     case '/':
22         if (y == 0)
23             printf("Greska: deljenje nulom nije dozvoljeno!\n");
24         else
25             printf("Rezultat je: %f\n", x * 1.0 / y);
26         break;
27     case '%':
28         printf("Rezultat je: %d\n", x % y);
29         break;
30     default:
31         printf("Greska: nepoznat operator!\n");
32     }
33
34     return 0;
35 }
```

### Rešenje 2.1.29

```
1  #include <stdio.h>

3  int main()
4  {
5      int d, m;
6      printf("Unesite dan i mesec");
7      scanf("%d%d", &d, &m);

9      /* Argument u naredbi switch mora biti celobrojna promenljiva,
10       dok argument u naredbi case mora biti celobrojna konstanta. */
11     switch (m) {
12         /* Ispitujemo da li vazi m==1 ili m==2 */
13         case 1:
14             case 2:
15                 printf("zima\n");
16                 break;
17         case 3:
18             if (d < 21)
19                 printf("zima\n");
20             else
21                 printf("prolece\n");
22             break;
23         case 4:
24             case 5:
25                 printf("prolece\n");
26                 break;
27         case 6:
28             if (d < 21)
29                 printf("prolece\n");
30             else
31                 printf("leto\n");
32             break;
33         case 7:
34             case 8:
35                 printf("leto\n");
36                 break;
37         case 9:
38             if (d < 23)
39                 printf("leto\n");
40             else
41                 printf("jesen\n");
42             break;
43         case 10:
44             case 11:
45                 printf("jesen\n");
46                 break;
47         case 12:
48             if (d < 22)
49                 printf("jesen\n");
50             else
51                 printf("zima\n");
```

```
    }  
53     return 0;  
55 }
```

### Rešenje 2.1.30

```
1  #include <stdio.h>  
  
3  int main()  
{  
5     int godina;  
     int mesec;  
7     int prestupna;  
  
9     printf("Unesite godinu: ");  
     scanf("%d", &godina);  
11  
     if (godina < 0) {  
13         printf("Lose uneta godina!\n");  
         return -1;  
15     }  
  
17     /* Provera da li je godina prestupna, zbog februara */  
     if ((godina % 4 == 0 && godina % 100 != 0)  
19         || godina % 400 == 0)  
         prestupna = 1;  
21     else  
         prestupna = 0;  
23  
     printf("Unesite redni broj meseca: ");  
25     scanf("%d", &mesec);  
  
27     switch (mesec) {  
     case 1:  
29         printf("Januar, 31 dan\n");  
         break;  
31     case 2:  
         if (prestupna)  
33             printf("Februar, 29 dana\n");  
         else  
35             printf("Februar, 28 dana\n");  
         break;  
37     case 3:  
         printf("Mart, 31 dan\n");  
39         break;  
         case 4:  
41             printf("April, 30 dana\n");  
             break;  
43     case 5:  
         printf("Maj, 31 dan\n");
```

```

45     break;
46 case 6:
47     printf("Jun, 30 dana\n");
48     break;
49 case 7:
50     printf("Jul, 31 dan\n");
51     break;
52 case 8:
53     printf("Avgust, 31 dan\n");
54     break;
55 case 9:
56     printf("Septembar, 30 dana\n");
57     break;
58 case 10:
59     printf("Oktobar, 31 dan\n");
60     break;
61 case 11:
62     printf("Novembar, 30 dana\n");
63     break;
64 case 12:
65     printf("Decembar, 31 dan\n");
66     break;
67 default:
68     printf("Lose unet redni broj meseca!\n");
69 }
70
71 return 0;
72 }

```

### Rešenje 2.1.31

```

1  #include <stdio.h>
2
3  int main()
4  {
5      int dan, mesec, godina, dozvoljen_broj_dana;
6
7      /* Citamo datum */
8      printf("Unesite datum: ");
9      scanf("%d.%d.%d", &dan, &mesec, &godina);
10
11     /* Proveravamo godinu */
12     if (godina < 0) {
13         printf("Datum nije korektan (neispravna godina)!\n");
14         return 0;
15     }
16
17     /* Proveravamo mesec */
18     if (mesec < 1 || mesec > 12) {
19         printf("Datum nije korektan (neispravan mesec)!\n");
20         return 0;
21     }
22 }

```

```
21 }

23 /* Ako je mesec korektan, proveravamo broj dana */
24 switch (mesec) {
25     case 1:
26     case 3:
27     case 5:
28     case 7:
29     case 8:
30     case 10:
31     case 12:
32         /* Dozvoljeni broj dana za januar, mart, maj, jul, avgust,
33            oktobar i decembar je 31 */
34         dozvoljen_broj_dana = 31;
35         break;
36     case 2:
37         /* Proveravamo da li je godina prestupna */
38         if (godina % 4 == 0 && godina % 100 != 0 || godina % 400 == 0)
39             /* Ako jeste, dozvoljeni broj dana za februar je 29 */
40             dozvoljen_broj_dana = 29;
41         else
42             /* Ako nije, dozvoljeni broj dana za februar je 28 */
43             dozvoljen_broj_dana = 28;
44         break;
45     case 4:
46     case 6:
47     case 9:
48     case 11:
49         /* Dozvoljeni broj dana za april, jun, septembar i novembar je
50            30 */
51         dozvoljen_broj_dana = 30;
52         break;
53 }
54 /* Proveravamo dan */
55 if (dan < 0 || dan > dozvoljen_broj_dana) {
56     printf("Datum nije korektan (neispravan dan)!\n");
57     return 0;
58 }
59
60 /* Sve provere su ispunjene pa zakljucujemo da je datum korektan */
61 printf("Ispravan datum!\n");
62
63 return 0;
64 }
```

### Rešenje 2.1.32

```
1 #include <stdio.h>

3 int main()
4 {
```

```
5  int dan, mesec, godina;
6  int prethodni_dan, prethodni_mesec, prethodni_godina;
7
8  /* Citamo datum */
9  printf("Unesite datum: ");
10 scanf("%d.%d.%d", &dan, &mesec, &godina);
11
12 /* Racunamo dan, mesec i godinu prethodnog dana */
13 prethodni_dan = dan - 1;
14 prethodni_mesec = mesec;
15 prethodni_godina = godina;
16
17 /* I po potrebi vrsimo korekcije */
18
19 /* Ako je u pitanju prvi u mesecu */
20 if (prethodni_dan == 0) {
21     /* Treba korigovati mesec */
22     prethodni_mesec = mesec - 1;
23     /* Ako je u pitanju januar */
24     if (prethodni_mesec == 0) {
25         /* Treba korigovati i godinu */
26         prethodni_mesec = 12;
27         prethodni_godina = godina - 1;
28     }
29
30     /* Analiziramo redni broj meseca kako bi odredili tacan dan */
31     switch (prethodni_mesec) {
32         case 1:
33         case 3:
34         case 5:
35         case 7:
36         case 8:
37         case 10:
38         case 12:
39             prethodni_dan = 31;
40             break;
41         case 2:
42             if ((prethodni_godina % 4 == 0 && prethodni_godina % 100 != 0)
43                 || prethodni_godina % 400 == 0)
44                 prethodni_dan = 29;
45             else
46                 prethodni_dan = 28;
47             break;
48         case 4:
49         case 6:
50         case 9:
51         case 11:
52             prethodni_dan = 30;
53     }
54 }
55
56 /* Ispisujemo datum koji smo izracunali */
```

## 2 Kontrola toka

---

```
57     printf("Prethodni datum: %d.%d.%d\n",
           prethodni_dan, prethodni_mesec, prethodni_godina);
59
60     return 0;
61 }
```

Rešenje 2.1.33      Rešenje je analogno rešenju zadatka 2.1.32.

### Rešenje 2.1.35

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

int main()
{
    char c;
    float x1, y1, x2, y2, x3, y3, x4, y4;
    float kab, kbc, kad;
    float dab, dad;
    float delta, deltap, deltaq;
    float 0;
    float k, n;

    printf("Unesi jedan karakter:");
    scanf("%c", &c);

    printf("Unesi realne koordinate 4 tacke:");
    scanf("%f%f%f%f%f%f%f", &x1, &y1, &x2, &y2, &x3, &y3, &x4, &y4);

    switch (c) {
    case 'k':
        if (y1 == y2 && y3 == y4 && x1 == x4 && x2 == x3) {
            dab = sqrt(pow(x1 - x2, 2) + pow(y1 - y2, 2));
            dad = sqrt(pow(x1 - x4, 2) + pow(y1 - y4, 2));
            0 = 2 * dab + 2 * dad;
            printf("Obim pravougaonika je %f\n", 0);
        } else
            printf("Tacke ne cine pravougaonik sa stranicama ",
                  "koje su paralelne koordinatnim osama\n");
        break;
    case 'h':
        /*
         * Ukoliko se tacke A(x1,y1) i B(x2,y2) ne nalaze na pravoj
         * koja je paralelna x osi, izracunamo k,n za pravu odredjenu
         * tackama A(x1,y1) i B(x2,y2) */
        if ((x1 - x2) != 0) {
            k = (y1 - y2) / (x1 - x2);
            n = y1 - k * x1;
            /*
             * Proverimo da li tacke C(x3,y3) i D(x4,y4) nalaze na toj
```



```

42     pravoj */
43     if (y3 == x3 * k + n && y4 == x4 * k + n)
44         printf("Tacke su kolinearne, pripadaju pravoj ",
45             "y=%f*x+%f\n", k, n);
46     else
47         printf("Tacke nisu kolinearne\n");
48 }
49 /*
50     Ukoliko se A i B nalaze na pravoj koja je paralelna x osi,
51     proverimo da li tacke C(x3,y3) i D(x4,y4) nalaze na toj
52     pravoj */
53     else if (x3 == x1 && x4 == x1)
54         printf("Tacke su kolinearne, pripadaju pravoj ",
55             "x=%f\n", x1);
56     else
57         printf("Tacke nisu kolinearne\n");
58     break;
59 case 'j':
60     delta = x1 * y2 - x2 * y1;
61     deltap = x2 * (y4 - y3) - y2 * (x4 - x3);
62     deltaq = x1 * (y4 - y3) - y1 * (x4 - x3);
63     if (delta != 0)
64         printf("Sistem ima jedinstveno resenje %.2f, %.2f\n",
65             deltap / delta, deltaq / delta);
66     else if (deltap == 0 && deltaq == 0)
67         printf("Sistem je neodredjen ili nema resenja.\n");
68     else
69         printf("Sistem nema resenja\n");
70     break;
71 default:
72     printf("Nekorektan unos\n");
73 }
74 return 0;
75 }

```

## 2.3 Petlje

**Zadatak 2.3.1** Napisati program koji 5 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 ceo broj  $n$  i ispisuje  $n$  puta tekst Mi volimo da programiramo.

*Primer 1*

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

[Rešenje 2.3.2]

**Zadatak 2.3.3** Napisati program koji učitava pozitivan ceo broj  $n$  a potom ispisuje sve cele brojeve od 0 do  $n$ .

*Primer 1*

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

*Primer 2*

```
|| INTERAKCIJA SA PROGRAMOM:  
| Unesite ceo pozitivan broj: -10  
| Neispravan unos. Promenljiva mora biti  
| pozitivna!
```

[Rešenje 2.3.3]

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

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

*Primer 1*

```
|| INTERAKCIJA SA PROGRAMOM:  
| Unesite dva cela broja: -2 4  
| -2 -1 0 1 2 3 4
```

*Primer 2*

```
|| INTERAKCIJA SA PROGRAMOM:  
| Unesite dva cela broja: 10 6  
| Neispravan unos. Nisu dobro zadate granice  
| intervala!
```

[Rešenje 2.3.4]

**Zadatak 2.3.5** Napisati program koji učitava ceo pozitivan broj i izračunava njegov faktorijel. U slučaju neispravnog unosa ispisati odgovarajuću poruku.

*Primer 1*

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite pozitivan broj: 18
|| Faktorijel = 640237370572800
```

*Primer 2*

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite pozitivan broj: 8
|| Faktorijel = 40320
```

*Primer 3*

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite pozitivan broj: 40
|| Broj je veliki, dolazi do
||   prekoračenja.
```

[Rešenje 2.3.5]

**Zadatak 2.3.6** Sa standardnog ulaza unose se realan broj  $x$  i ceo pozitivan broj  $n$ . Napisati program koji izračunava  $n$ -ti stepen broja  $x$ , tj.  $x^n$ .

*Primer 1*

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

*Primer 2*

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

*Primer 3*

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

[Rešenje 2.3.6]

**Zadatak 2.3.7** Sa standardnog ulaza unose se realan broj  $x$  i ceo broj  $n$ . Napisati program koji izračunava  $n$ -ti stepen broja  $x$ .

*Primer 1*

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

*Primer 2*

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite redom brojeve x i n: -3 2
|| 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 ceo pozitivan broj  $x$  ispisuje sve njegove prave delioce. U slučaju greške pri unosu podataka ispisati odgovarajuću poruku.

*Primer 1*

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite ceo broj veci od 0: 100
|| 2 4 5 10 20 25 50
```

*Primer 2*

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

[Rešenje 2.3.8]

**Zadatak 2.3.9** Napisati program koji za uneti prirodan broj uklanja sve nule sa njegove desne strane. Ispisati novodobijeni broj.

*Primer 1*

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

*Primer 2*

```
|| INTERAKCIJA SA PROGRAMOM:  
| Unesite broj: 856  
| 856
```

*Primer 3*

```
|| INTERAKCIJA SA PROGRAMOM:  
| Unesite broj: 140  
| 14
```

[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 prirodan broj ispisuje da li je on deljiv sumom svojih cifara.

*Primer 1*

```
|| INTERAKCIJA SA PROGRAMOM:  
| Unesite broj: 12  
| Deljiv je sumom svojih  
|   cifara.
```

*Primer 2*

```
|| INTERAKCIJA SA PROGRAMOM:  
| Unesite broj: 2564  
| Nije deljiv sumom svojih  
|   cifara.
```

*Primer 3*

```
|| INTERAKCIJA SA PROGRAMOM:  
| Unesite broj: -4  
| Neispravan ulaz.
```

*Primer 4*

```
|| INTERAKCIJA SA PROGRAMOM:  
| Unesite broj: 0  
| Neispravan ulaz.
```

[Rešenje 2.3.11]

**Zadatak 2.3.12** Napisati program koji učitava pozitivan ceo broj  $n$ , a zatim učitava  $n$  celih brojeva i ispisuje sumu pozitivnih i sumu negativnih unetih brojeva.

*Primer 1*

```

INTERAKCIJA SA PROGRAMOM:
Unesite broj: 7
Unesite brojeve:
8 -50 45 2007 -67 -123 14
Suma pozitivnih: 2074
Suma negativnih: -240

```

*Primer 2*

```

INTERAKCIJA SA PROGRAMOM:
Unesite broj: 5
Unesite brojeve:
-5 -20 -4 -200 -8
Suma pozitivnih: 0
Suma negativnih: -237

```

*Primer 3*

```

INTERAKCIJA SA PROGRAMOM:
Unesite broj: -6
Neispravan unos.

```

[Rešenje 2.3.12]

**Zadatak 2.3.13** Program unosi ceo pozitivan broj  $n$ , a potom i  $n$  celih brojeva. Izračunati i ispisati zbir onih brojeva koji su neparni i negativni.

*Primer 1*

```

INTERAKCIJA SA PROGRAMOM:
Unesite broj n: 5
Unesite n brojeva:
1 -5 -6 3 -11
-16

```

*Primer 2*

```

INTERAKCIJA SA PROGRAMOM:
Unesite broj n: 4
Unesite n brojeva:
5 8 13 17
0

```

*Primer 3*

```

INTERAKCIJA SA PROGRAMOM:
Unesite broj n: -4
Neispravan unos.

```

[Rešenje 2.3.13]

**Zadatak 2.3.14** Program učitava ceo pozitivan broj  $n$ , a potom  $n$  celih brojeva. Naći sumu brojeva koji su deljivi sa 5, a nisu deljivi sa 7. U slučaju greške pri unosu podataka ispisati odgovarajuću poruku.

*Primer 1*

```

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

```

*Primer 2*

```

INTERAKCIJA SA PROGRAMOM:
Unesite broj n: -3
Neispravan unos.

```

*Primer 3*

```

INTERAKCIJA SA PROGRAMOM:
Unesite broj n: 10
Unesite brojeve:
-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** Nikola želi da obraduje baku i da joj kupi jedan poklon u radnji. On na raspolaganju ima  $m$  novaca. U radnji se nalazi  $n$  artikala i

## 2 Kontrola toka

zanima ga koliko ima artikala u radnji čija cena je manja ili jednaka  $m$ . Napisati program koji pomaže Nikoli da brzo odrediti broj artikala. Program učitava realan pozitivan broj  $m$ , ceo pozitivan broj  $n$  i  $n$  realnih pozitivnih brojeva različitih od 0. Ispisati koliko artikala ima manju ili jednaku cenu od  $m$ . U slučaju greške ispisati odgovarajuću poruku.

### Primer 1

```
INTERAKCIJA SA PROGRAMOM:
Unesite broj m: 12.37
Unesite broj n: 5
Unesite n brojeva: 11 54.13 6 13 8
3
```

### Primer 2

```
INTERAKCIJA SA PROGRAMOM:
Unesite broj m: 2
Unesite broj n: 4
Unesite n brojeva: 1 11 4.32 3
1
```

### Primer 3

```
INTERAKCIJA SA PROGRAMOM:
Unesite broj m: 2
Unesite broj n: -4
Broj artikala ne može biti negativan.
```

### Primer 4

```
INTERAKCIJA SA PROGRAMOM:
Unesite broj m: 30
Unesite broj n: 4
Unesite n brojeva: 67 -100 23 98
Cena ne može biti negativna.
```

[Rešenje 2.3.15]

**Zadatak 2.3.16** Napisati program koji učitava cele brojeve sve dok se ne unese nula. Nakon toga ispisati proizvod onih unetih brojeva koji su pozitivni.

### Primer 1

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

### Primer 2

```
INTERAKCIJA SA PROGRAMOM:
Unesite brojeve:
-5 -200 -43 0
Nisu uneseni pozitivni
brojevi.
```

### Primer 2

```
INTERAKCIJA SA PROGRAMOM:
Unesite brojeve: 0
Nisu uneseni brojevi.
```

[Rešenje 2.3.16]

**Zadatak 2.3.17** Napisati program koji za pozitivan ceo broj proverava i ispisuje da li se cifra 5 nalazi u njegovom zapisu.

### Primer 1

```
INTERAKCIJA SA PROGRAMOM:
Unesite broj: 1857
Cifra 5 se nalazi u zapisu!
```

### Primer 2

```
INTERAKCIJA SA PROGRAMOM:
Unesite broj: 84
Cifra 5 se ne nalazi u
zapisu!
```

### Primer 3

```
INTERAKCIJA SA PROGRAMOM:
Unesite broj: -235515
Cifra 5 se nalazi u zapisu!
```

[Rešenje 2.3.17]

**Zadatak 2.3.18** Program učitava cele brojeve sve do unosa broja nula 0. Napisati program koji 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: 762 -12 800 2010 -356 899 -101
|| 0
|| Aritmeticka sredina: 571.7143
```

*Primer 3*

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

[Rešenje 2.3.18]

**Zadatak 2.3.19** U prodavnici se nalaze artikala čije cene su realni pozitivni brojevi. Program unosi cene artikala sve do unosa broja nula 0. Napisati program koji izračunava i ispisuje prosečnu vrednost cena u radnji.

*Primer 1*

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

*Primer 2*

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite cene: 6.32 -9
|| Cena ne može biti negativana.
```

*Primer 3*

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite cene: 0
|| Nisu unesene cene.
```

[Rešenje 2.3.19]

**Zadatak 2.3.20** Program učitava ceo pozitivan broj  $n$ , a potom  $n$  realnih brojeva. Odrediti koliko puta je prilikom unosa došlo do promene znaka. Ispisati dobijenu vrednost.

*Primer 1*

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite broj n: 10
|| Unesite brojeve:
|| 7.82 4.3 -1.2 56.8 -3.4 -72.1 8.9 11.2 -11.2
|| -102.4
|| 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  
|| Neispravan unos.
```

### Primer 4

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesite broj n: 0  
|| Broj promena je 0.
```

[Rešenje 2.3.20]

**Zadatak 2.3.21** U prodavnici se nalazi  $n$  artikala čije cene su realni brojevi. Napisati program koji učitava  $n$ , a potom i cenu svakog od  $n$  artikala i određuje i ispisuje najmanju cenu.

### Primer 1

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesite broj artikla: 6  
|| Unesite artikle:  
|| 12 3.4 90 100.53 53.2 12.8  
|| Minimalna cena je: 3.400000
```

### Primer 2

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesite broj artikla: 3  
|| Unesite artikle: 4 -8 92  
|| Cena ne moze biti negativna.
```

### Primer 3

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesite broj artikla: -9  
|| Neispravan unos.
```

[Rešenje 2.3.21]

**Zadatak 2.3.22** Program učitava ceo pozitivan broj  $n$ , a zatim i  $n$  celih brojeva. Napisati program koji ispisuje 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 je 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 je -89.
```

### Primer 3

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesite broj n: -12  
|| Neispravan unos.
```

[Rešenje 2.3.22]

**Zadatak 2.3.23** Program učitava ceo pozitivan broj  $n$ , a zatim i  $n$  celih brojeva. Napisati program koji ispisuje 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 25 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.

```

[Rešenje 2.3.23]

**Zadatak 2.3.24** Program učitava ceo pozitivan broj  $n$ , a zatim i  $n$  celih brojeva. Napisati program koji ispisuje 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: 41 669 8
|| Broj sa najvećom vodećom cifrom je 8.

```

[Rešenje 2.3.24]

**Zadatak 2.3.25** Vršna su merenja nadmorskih visina na određenom delu teritorije i naučnike zanima razlika između najveće i najmanje nadmorske visine. Napisati program koji učitava realne 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

```

[Rešenje 2.3.25]

**Zadatak 2.3.26** Program učitava cele pozitivane brojeve  $n$  ( $n > 1$ ) i  $d$ , a zatim i  $n$  celih brojeva. Napisati program koji izračunava koliko ima parova uzastopnih brojeva među unetim brojevima koji se nalaze na rastojanju  $d$ . Rastojanje između brojeva je definisano sa  $d(x, y) = |y - x|$ . Ispisati rezultat.

### 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
```

[Rešenje 2.3.26]

**Zadatak 2.3.27** Napisati program koji uneti prirodan broj transformiše tako što svaku parnu cifru u zapisu broja uveća za jedan. Ispisati novodobijeni broj.

### Primer 1

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

### Primer 2

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

### Primer 3

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

[Rešenje 2.3.27]

**Zadatak 2.3.28** Napisati program koji formira i ispisuje broj koji se dobija izbacivanjem svake druge cifre polaznog celog broja, počevši od krajnje desne cifre.

### Primer 1

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

### Primer 2

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

### Primer 3

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

### Primer 4

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

[Rešenje 2.3.28]

\* **Zadatak 2.3.29** Napisati program koji na osnovu unetog prirodnog broja formira i ispisuje broj koji se dobija izbacivanjem cifara koje su jednake zbiru svojih suseda.

*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
|| Neispravan unos.

```

[Rešenje 2.3.29]

\* **Zadatak 2.3.30** Broj je *palindrom* ukoliko se isto čita i sa leve i sa desne strane. Napisati program koji učitava prirodan broj i proverava da li je učitani broj palindrom.

*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 2.3.30]

**Zadatak 2.3.31** Fibonačijev niz počinje ciframa 1 i 1, a svaki član se dobija zbirom prethodna dva. Napisati program koji učitava ceo prirodan broj  $n$  i određuje i ispisuje  $n$ -ti član Fibonačijevog niza.

*Primer 1*

```

|| INTERAKCIJA SA PROGRAMOM:
|| Unesite ceo broj: 10
|| Trazeni broj je: 55

```

*Primer 2*

```

|| INTERAKCIJA SA PROGRAMOM:
|| Unesite ceo broj: -100
|| Neispravan unos. Pozicija u Fibonacijevom
|| nizu mora biti pozitivan broj koji nije 0!

```

*Primer 3*

```

|| INTERAKCIJA SA PROGRAMOM:
|| Unesite ceo broj: 78
|| Trazeni broj je: 375819880

```

*Primer 4*

```

|| INTERAKCIJA SA PROGRAMOM:
|| Unesite ceo broj: 20
|| Trazeni broj je: 6765

```

[Rešenje 2.3.31]

\* **Zadatak 2.3.32** 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  $a_0$  (ceo pozitivan broj) štampa niz brojeva sve do onog člana niza koji je jednak 1.

### Primer 1

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesite ceo broj: 56  
|| 56 28 14 7 11 17 26 13 20 10  
|| 5 8 4 2 1
```

### Primer 2

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesite ceo broj: -48  
|| Nekorektan unos. Broj mora biti pozitivan.
```

### Primer 3

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

[Rešenje 2.3.32]

**\* Zadatak 2.3.33** 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 prirodan broj  $k$  ispisuje dimenzije papira  $A_k$  u milimetrima. Rezultat ispisati kao celobrojne vrednosti.

### Primer 1

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

### Primer 2

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesite format papira: 3  
|| 297 420
```

### Primer 3

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesite format papira: 7  
|| 74 105
```

### Primer 4

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

[Rešenje 2.3.33]

**Zadatak 2.3.34** 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 2.3.34]

**Zadatak 2.3.35** 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 2.3.35]

**Zadatak 2.3.36** Program učitava ceo pozitivan 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.

*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
```

[Rešenje 2.3.36]

**Zadatak 2.3.37** Program učitava ceo broj  $n$ , a zatim i  $n$  karaktera. Napisati program koji proverava da li se od unetih karaktera može napisati reč *Zima*.

*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 može 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: -
Može se napisati rec Zima.
```

[Rešenje 2.3.37]

## 2 Kontrola toka

---

**Zadatak 2.3.38** Napisati program koji učitava ceo pozitivan broj  $n$  i ispisuje vrednost sume kubova brojeva od 1 do  $n$ , odnosno  $s = 1 + 2^3 + 3^3 + \dots + n^3$ . U slučaju greške pri unosu podataka ispisati odgovarajuću poruku.

*Primer 1*

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesite pozitivan ceo broj: 14  
|| Suma kubova od 1 do 14: 11025
```

*Primer 2*

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesite pozitivan ceo broj 25  
|| Suma kubova od 1 do 25: 105625
```

[Rešenje 2.3.38]

**Zadatak 2.3.39** Napisati program koji učitava ceo pozitivan broj  $n$  i ispisuje sumu kubova,  $s = 1 + 2^3 + 3^3 + \dots + k^3$ , za svaku vrednost  $k = 1, \dots, n$ . U slučaju greške pri unosu podataka ispisati odgovarajuću poruku.

*Primer 1*

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesite pozitivan ceo broj: 5  
|| i=1, s=1  
|| i=2, s=9  
|| i=3, s=36  
|| i=4, s=100  
|| i=5, s=225
```

*Primer 2*

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesite pozitivan ceo broj 8  
|| i=1, s=1  
|| i=2, s=9  
|| i=3, s=36  
|| i=4, s=100  
|| i=5, s=225  
|| i=6, s=441  
|| i=7, s=784  
|| i=8, s=1296
```

[Rešenje 2.3.39]

**Zadatak 2.3.40** Program učitava realan broj  $x$  i ceo pozitivan broj  $n$ . Napisati program koji izračunava i ispisuje sumu  $S = x + 2 \cdot x^2 + 3 \cdot x^3 + \dots + n \cdot x^n$ .

*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
```

[Rešenje 2.3.40]

**Zadatak 2.3.41** Program učitava realan broj  $x$  i ceo pozitivan broj  $n$ . Napisati program koji izračunava i ispisuje sumu  $S = 1 + \frac{1}{x} + \frac{1}{x^2} + \dots + \frac{1}{x^n}$ .

## 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
```

[Rešenje 2.3.41]

**\* Zadatak 2.3.42** Napisati program koji učitava realane 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 2.3.42]

**\* Zadatak 2.3.43** Napisati program koji učitava realane 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 2.3.43]

**Zadatak 2.3.44** Napisati program koji učitava realan broj  $x$  i prirodan broj  $n$  izračunava sumu  $S = (1 + \cos(x)) \cdot (1 + \cos(x^2)) \cdot \dots \cdot (1 + \cos(x^n))$ . NAPOMENA: *Voditi računa o efikasnosti rešenja.*

### Primer 1

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

### Primer 2

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

[Rešenje 2.3.44]

\* **Zadatak 2.3.45** Napisati program koji učitava ceo prirodan broj  $n$  i ispisuje vrednost razlomka

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

### Primer 1

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite prirodan broj: 4
|| Razlomak = 0.697674
```

### Primer 2

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite prirodan broj: 20
|| Razlomak = 0.697775
```

### Primer 3

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite prirodan broj: 0
|| Neispravan unos.
```

[Rešenje 2.3.45]

\* **Zadatak 2.3.46** Napisati program koji računa sumu

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

za unete cele brojeve  $x$  i  $n$ . NAPOMENA: Voditi računa o efikasnosti rešenja i o mogućnosti prekoračenja.

### Primer 1

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite x i n: 5.6 8
|| S=0.735084
```

### Primer 2

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite x i n: 14.32 11
|| S=17273.136719
```

### Primer 3

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite prirodan broj: -6
|| Neispravan unos.
```

[Rešenje 2.3.46]



\* **Zadatak 2.3.47** Program učitava ceo pozitivan broj  $n$  veći od 0. Napisati program koji računa proizvod

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

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: 5
|| 1.838108
```

*Primer 2*

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

*Primer 3*

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

*Primer 4*

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

[Rešenje 2.3.47]

\* **Zadatak 2.3.48** Program učitava ceo pozitivan neparan broj  $n$ . Napisati program koji za uneto  $n$  izračunava:

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

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
|| Neispravan unos
```

*Primer 4*

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

[Rešenje 2.3.48]

## 2 Kontrola toka

**Zadatak 2.3.49** Program učitava realne brojeve  $x$  i  $a$  i ceo pozitivan broj  $n$  veći od 0. Napisati program koji izračunava:

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

U slučaju greške pri unosu podataka ispisati odgovarajuću poruku.

### Primer 1

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite dva relana broja x i a:: 3.2 0.2
|| Unesite prirodan broj: 5
|| Izraz = 135380494030332048.000000
```

### Primer 2

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite dva relana broja x i a:: 2 1
|| Unesite prirodan broj: 3
|| Izraz = 10201.000000
```

### Primer 3

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite dva relana broja x i a:: 2.6 0.3
|| Unesite prirodan broj: 3
|| Izraz = 5800.970129
```

### Primer 4

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite dva relana broja x i a:: 5.4 7
|| Unesite prirodan broj: -2
|| Neispravan unos.
```

[Rešenje 2.3.49]

**Zadatak 2.3.50** Za unetu pozitivnu celobrojnu vrednost  $n$  napisati programe koji ispisuju odgovarajuće brojeve. Pretpostaviti da je unos korektan.

- (a) Napisati program koji za unetu pozitivnu celobrojnu 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 2.3.50]

## 2 Kontrola toka

---

**Zadatak 2.3.51** Napisati program koji za unet prirodan broj  $n$  zvezdicama iscrtava odgovarajuću sliku. Pretpostaviti da je unos korektan.

- (a) Slika sadrži 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 sadrži rub kvadrata dimenzije  $n$ .

*Primer 1*

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

*Primer 2*

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

- (c) Slika sadrži 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 2.3.51]

\* **Zadatak 2.3.52** Napisati program koji za uneti prirodan broj  $n$  zvezdicama iscrtava slovo  $X$  dimenzije  $n$ . Pretpostaviti da je unos korektan.

*Primer 1*

```

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

```

*Primer 2*

```

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

```

[Rešenje 2.3.52]

**\* Zadatak 2.3.53** Napisati program koji za uneti prirodan neparan broj  $n$  korišćenjem znaka + iscrtava veliko + dimenzije  $n$ . Pretpostaviti da je unet prirodan broj.

*Primer 1*

```

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

```

*Primer 2*

```

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

```

*Primer 3*

```

INTERAKCIJA SA PROGRAMOM:
Unesite broj n: 4
Pogresan unos.

```

[Rešenje 2.3.53]

**Zadatak 2.3.54** Napisati program koji učitava prirodan broj  $n$ , a potom iscrtava odgovarajuću sliku. Pretpostaviti da je unos korektan.

- (a) Slika sadrži 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
****
***
**
*

```

## 2 Kontrola toka

---

- (b) Slika sadrži 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 sadrži 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 sadrži 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 sadrži 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 sadrži 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 [2.3.54](#)]

**Zadatak 2.3.55** Napisati program koji učitava ceo broj  $n$ , a potom iscrtava odgovarajuću sliku.

- (a) Slika sadrži 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 sadrži 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 sadrži 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 sadrži 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 2.3.56** Napisati program koji za uneti prirodan broj  $n$  iscrtava strelice dimenzije  $n$ . Pretpostaviti da je unos korektan.

Primer 1

```

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

```

Primer 2

```

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

```

[Rešenje 2.3.56]

\* **Zadatak 2.3.57** Napisati program koji učitava ceo broj  $n$ , i iscrtava 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  $n$ . Pretpostaviti da je unos korektan.

### Primer 1

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

[Rešenje 2.3.57]

\* **Zadatak 2.3.58** Program učitava prirodne brojeve  $m$  i  $n$ . Napisati program koji iscrtava jedan do drugog stranice  $n$  kvadrata čija je svaka strana sastavljena od  $m$  zvezdica razdvojenih prazninom. Podrazumevati da je unos korektan.

### Primer 1

```
INTERAKCIJA SA PROGRAMOM:
Unesite broj n: 5 3
* * * * *
*   *   *   *
*   *   *   *
*   *   *   *
* * * * *
```

### Primer 2

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

[Rešenje 2.3.58]

\* **Zadatak 2.3.59** Program učitava prirodan broj  $n$ . Napisati program koji štampa romb sastavljen od minusa u pravougaoniku sastavljenom od zvezdica. Podrazumevati da je unos korektan.

### Primer 1

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

### Primer 2

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

[Rešenje 2.3.59]

**Zadatak 2.3.60** Napisati program koji učitava ceo broj  $n$  ( $n \geq 2$ ) i koji iscrtava sliku kuće sa krovom: kuća je kocka 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 2.3.60]

\* **Zadatak 2.3.61** Program učitava ceo pozitivan broj  $n$ . Napisati program koji 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. Za neispravan unos, program ispisuje odgovarajuću poruku. Pretpostaviti da je unos korektan.

*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: -4
-1
```

*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 2.3.61]

\* **Zadatak 2.3.62** Napisati program koji učitava ceo pozitivan 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). U slučaju greške pri unosu podataka odštampati odgovarajuću poruku.

### 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
```

### Primer 4

```
INTERAKCIJA SA PROGRAMOM:  
Unesite broj n: -23  
Neispravan unos.
```

[Rešenje 2.3.62]

## 2.4 Rešenja

### Rešenje 2.3.1

```
#include <stdio.h>  
  
2  
int main()  
4 {  
    /* Promenljiva i kontrolise koliko puta ce se petlja izvorsiti i  
6     naziva se brojac petlje. Njenu pocetnu vrednost postavljamo na  
    0 jer se u pocetku petlja nije ni jednom izvorsila. */  
8     int i = 0;  
  
10    /* Pre ulaska u telo petlje proverava se da li je ispunjen uslov  
    ulaska u petlju. */  
12    while (i < 5) {  
        /* Ukoliko uslov ulaska u petlju jeste ispunjen, izvorsava se  
14        telo petlje. */  
        /* Ispisujemo trazeni tekst. */  
16        printf("Mi volimo da programiramo.\n");  
  
18        /* Uvecava se brojac za jedan jer je jednom izvorseno telo  
        petlje. Ako bi ova vrednost ostala nepromenjena, petlja bi  
20        se izvorsavala beskonacno. */  
        i++;  
22  
        /* Nakon poslednje naredbe tela petlje ponovo se ispituje uslov  
24        petlje. */
```

```

    }
26     return 0;
28 }

```

### Rešenje 2.3.2

```

#include<stdio.h>
2
int main()
4 {
    /* Brojac u petlji. */
6     int i = 0;
    /* Promenljiva koja oznacava koliko puta cemo ispisati trazeni
8     tekst. */
    int n;

10     printf("Unesite ceo broj: ");
12     scanf("%d", &n);

14     /* Pre ulaska u telo petlje proverava se da li je ispunjen uslov
        ulaska u petlju. */
16     while (i < n) {
        printf("Mi volimo da programiramo.\n");
18         i++;
    }

20     return 0;
22 }

```

### Rešenje 2.3.3

```

1  #include <stdio.h>
3  int main()
4  {
5      /* Promenljivu x koristimo u dve svrhe. Prvo, ova promenljiva
        kontrolise koliko puta se petlja izvrasila. Drugo, ovu
7      promenljivu koristimo za ispis potrebnih vrednosti. */
        int x;
9      /* Promenljiva n se unosi i odredjuje koliko brojeva ispisujemo. */
        int n;

11     printf("Unesi pozitivan ceo broj: ");
13     scanf("%d", &n);

15     /* U slucaju neispravnih podataka ispisujemo odgovarajucu poruku
        i izlazimo iz programa. */
17     if (n < 0) {

```

## 2 Kontrola toka

```
    printf("Neispravan unos. Promenljiva mora biti pozitivna!\n");
19     return -1;
    }

21     /* Ispis pocinjemo od 0, zato promenljivu x postavljamo na 0. */
23     x = 0;
    while (x <= n) {
25         /* Ispisujemo broj. */
        printf("%d\n", x);
27         /* Uvecavamo promenljivu za jedan jer smo broj ispisali i sada
           zelimo da ispisemo sledeci broj. */
29         x++;
    }
31     return 0;
33 }
```

### Rešenje 2.3.4

```
1  /* Resenje pod a). */
3  #include <stdio.h>

5  int main()
6  {
7      /* Promenljive koje oznacavaju granice intervala. */
      int n, m;
9      /* Promenljiva koja oznacava trenutno ispisani broj intervala. */
      int i;

11     printf("Unesi dva cela broja: ");
13     scanf("%d%d", &n, &m);

15     if (m < n) {
16         printf
17         ("Neispravan unos. Nisu dobro zadate granice intervala!\n");
18         return -1;
19     }

21     /* Na pocetku ispisujemo prvi broj intervala, a to je n. */
      i = n;
23     /* uslov petlje se proverava pre ulaska u telo petlje */
      while (i <= m) {
25         printf("%d ", i);
           i++;
27     }

29     printf("\n");

31     return 0;
```

33 }

```
1  /* Resenje pod b). */
3  #include <stdio.h>
5  int main()
6  {
7      /* Promenljive koje oznacavaju granice intervala. */
9      int n, m;
10     /* Promenljiva koja oznacava trenutno ispisani broj intervala. */
11     int i;
13     printf("Unesi dva cela broja: ");
14     scanf("%d%d", &n, &m);
15
16     if (m < n) {
17         printf
18             ("Neispravan unos. Nisu dobro zadate granice intervala!\n");
19         return -1;
20     }
21
22     /* naredba i=n se izvsava jednom, pre prve iteracije */
23     for (i = n; i <= m; i++)      /* uslov petlje i<=m se proverava
24                                     pre svake iteracije */
25         printf("%d ", i);        /* naredba i++ se izvsava nakon
26                                     svake iteracije */
27
28     printf("\n");
29
30     return 0;
31 }
32 }
```

```
1  /* Resenje pod c). */
2  #include <stdio.h>
3
4  int main()
5  {
6      /* Promenljive koje oznacavaju granice intervala. */
7      int n, m;
8      /* Promenljiva koja oznacava trenutno ispisani broj intervala. */
9      int i;
10
11     printf("Unesi dva cela broja: ");
12     scanf("%d%d", &n, &m);
13
14     if (m < n) {
15         printf
```

## 2 Kontrola toka

```
17     ("Neispravan unos. Nisu dobro zadate granice intervala!\n");
    return -1;
}

19
/* Uslov petlje se proverava na kraju svake iteracije. */
/* Zbog toga se do while petlja izvršava bar jednom, čak i u
21   slučaju da uslov petlje nikada nije ispunjen. */
23   i = n;
   do {
25       /* Petlja se zapocinje bez provere
           uslova. */
       printf("%d ", i);      /* Stampa se vrednost promenljive
27                               i. */
       i++;                  /* Uvecava se vrednost promenljive
29                               i. */
   }
31   while (i <= m);          /* Proverava se uslov i ukoliko je
                               ispunjen, nastavlja se sa
33                               sledecom iteracijom. */
/* U suprotnom, petlja se završava i program se nastavlja od prve
35   naredbe koja sledi za petljom. */
   printf("\n");
37   return 0;
39 }
```

### Rešenje 2.3.5

```
1 #include<stdio.h>

3 int main()
{
5     int x;
    /* U promenljivoj f se pamti izracunati faktorijel. Kako
7     faktorijel je jako veliki broj, za tip podataka se uzima
        unsigned long, da bi mogla da se upise sto veca vrednost. */
9     unsigned long f;
    int i;
11    int original;

13    printf("Unesite pozitivan broj: ");
    scanf("%d", &x);

15
    if (x < 0) {
17        printf("Nekorektan unos\n");
        return -1;
19    }

21    if (x >= 22) {
        printf("Broj je veliki, dolazi do prekoracenja.\n");
23        return -1;
    }
}
```



```
25     original = x;
27     f = 1;

29     while (x > 1) {
31         f = f * x;
31         x--;
33     }

33     printf("Faktorijel = %lu\n", f);
35
35     return 0;
37 }
```

### Rešenje 2.3.6

```
1  #include <stdio.h>

3  int main()
4  {
5      int n;
6      float x;
7      float vrednost;
8      unsigned exp;

9

10     printf("Unesite redom brojeve x i n: ");
11     scanf("%f %d", &x, &n);

12
13     if (n < 0) {
14         printf("Neispravan unos.\n");
15         return -1;
16     }

17
18     /* Pocetna vrednost stepena koji se racuna. */
19     vrednost = 1;

20
21     for (exp = 1; exp <= n; exp++)
22         vrednost = vrednost * x;

23     printf("%f\n", vrednost);

24
25     return 0;
26 }
27 }
```

### Rešenje 2.3.7

```
1  #include <stdio.h>

3  int main(void)
```

## 2 Kontrola toka

---

```
{
5   int n, n_abs;
   float x;
7   float vrednost;
   unsigned exp;

9   printf("Unesite redom brojeve x i n: ");
11  scanf("%f %d", &x, &n);

13  /* Pocetna vrednost stepena koji se racuna. */
   vrednost = 1;

15

17  /* Stepenovanje. */
   n_abs = abs(n);
   for (exp = 1; exp <= n_abs; exp++)
19     vrednost = vrednost * x;

21  /* Ukoliko je stepen bio negativan treba odrediti 1/x^n, sto je
      zapravo 1/vrednost. */
23  if (n < 0) {
      printf("%.3f\n", 1 / vrednost);
25  } else {
      printf("%.3f\n", vrednost);
27  }

29  return 0;
}
```

### Rešenje 2.3.8

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

3

5  int main()
   {
7     int x;
     /* Brojac u petlji. */
     int i;

9

11    /* Ucitavamo broj. */
     printf("Unesi ceo broj veci od 0: ");
     scanf("%d", &x);

13

15    if (x <= 0) {
        printf("Neispravan unos.\n");
        return -1;
17    }

19    /* 1. nacin */
     printf("----- 1. nacin ----- \n");
21    for (i = 2; i < x; i++) {
```

```

23     /* Proverava se da li i deli broj x i ako je to slucaj ispusje
        se i. */
25     if (x % i == 0)
        printf("%d \n", i);
    }

27     /* 2. nacin (brzi) -- Ne proveravaju se svi brojevi od 2 do x,
        vec se petlja izvršava dok ne stignemo do korena broja. */
29     printf("----- 2. nacin ----- \n");
    for (i = 2; i <= sqrt(x); i++) {
        /* Proveravamo da li i deli broj x. */
31         if (x % i == 0)
            /* U slucaju kada je delilac koren broja, npr. 4 za 16,
33             ispisujemo ga jednom. */
            if (i == x / i)
35                 printf("%d \n", i);
            /* U suprotnom, npr. 2 za 16, ispisujemo i 2 i 8. */
37         else
            printf("%d %d \n", i, x / i);
39     }

41     return 0;
43 }

```

### Rešenje 2.3.9

```

1  #include <stdio.h>

3  int main()
4  {
5      int n;

7      /* Ucitavamo broj */
8      printf("Unesite broj: ");
9      scanf("%d", &n);

11     if (n == 0) {
12         printf("0\n");
13     } else {
14         /* Sve dok je poslednja cifra u zapisu broja n nula */
15         while (n % 10 == 0) {
16             /* Broj delimo sa 10 tj. uklanjamo mu nulu sa kraja */
17             n = n / 10;
18         }

19         /* Ispisujemo rezultat */
21         printf("%d\n", n);
22     }

23     return 0;
25 }

```

### Rešenje 2.3.10

```
1 #include<stdio.h>
2 #include<stdlib.h>
3
4 int main()
5 {
6     int x;
7     char cifra;
8     printf("Unesi ceo broj:");
9     scanf("%d", &x);
10
11     /* Pretvaranje u apsolutnu vrednost se vrsi za slucaj kada je
12        unet negativan broj kako bismo osigurali da ce nam izdvojene
13        cifre biti pozitivne. */
14     x = abs(x);
15
16     /* Kako uklanjamo cifre broja (pogledati telo petlje) u nekom
17        trenutku broj ce postati 0 jer smo uklonili sve njegove cifre.
18        Tada prekidamo rad petlje. */
19     while (x > 0) {
20         /* Izdvajamo poslednju cifru broja x. */
21         cifra = x % 10;
22         printf("%d\\n", cifra);
23         /* Uklanjamo poslednju cifru broja x. */
24         x /= 10;
25     }
26
27     return 0;
28 }
```

### Rešenje 2.3.11

```
1 #include <stdio.h>
2
3 int main()
4 {
5     /* Prirodni broj koji se unosi. */
6     int n;
7     /* Promenljiva u koju se smesta suma cifara broja. */
8     int suma = 0;
9     /* Pomocna promenljiva u koju se smesta unesen broj. */
10    int pom_n;
11
12    printf("Unesi broj ");
13    scanf("%d", &n);
```

```

15  /* U zadatku pise da se unosi prirodan broj, sto znaci da treba
    proveriti da li je veci od 0 */
17
18  if (n <= 0) {
19      printf("Neispravan unos.\n");
20      return -1;
21  }
22
23  /* Potrebno je koristiti pomocnu promenljivu jer u telu petlje se
    odstranjuju cifre broja i na taj nacin uneseni broj se menja.
25      Nakon rada petlje potrebno je ponovo koristiti uneseni broj, a
    to znaci da treba sacuvati neizmenjen broj. */
27  pom_n = n;
28
29  while (pom_n != 0) {
30      /* Na sumu dodajemo poslednju cifru. */
31      suma += pom_n % 10;
32      /* Sa broja skidamo poslednju cifru. */
33      pom_n /= 10;
34  }
35
36  if (n % suma == 0)
37      printf("Deljiv je sumom svojih cifara.\n");
38  else
39      printf("Nije deljiv sumom svojih cifara.\n");
40
41  return 0;
42  }

```

### Rešenje 2.3.12

```

1  #include<stdio.h>
2
3  int main()
4  {
5      int n;
6      /* Oznaka broja koji unosimo u jednoj iteraciji petlje. */
7      int x;
8      int suma_poz;
9      int suma_neg;
10     /* Brojac. */
11     int i;
12
13     printf("Unesi pozitivan ceo broj:");
14     scanf("%d", &n);
15
16     if (n < 0) {
17         printf("Neispravan unos.\n");
18         return -1;
19     }

```

## 2 Kontrola toka

---

```
21  /* Promenljivama koje ce sadrzati sume se pre ulaska u petlju
    dodeljuje 0 (neutral za sabiranje). */
23  suma_poz = 0;
    suma_neg = 0;
25  i = 0;

27  printf("Unesite brojeve: ");
    while (i < n) {
29      scanf("%d", &x);

31      if (x < 0)
          suma_neg += x;
33      else
          suma_poz += x;

35      i++;
37  }

39  printf(" Suma pozitivnih: %d\n Suma negativnih: %d\n", suma_poz,
        suma_neg);
41  return 0;
}
```

### Rešenje 2.3.13

```
1  #include <stdio.h>

3  int main()
{
5      /* Promenljiva x oznacava tekuci uneti broj. */
    int n, x;
    /* Brojac. */
7    int i;
    int zbir = 0;

9

11     printf("Unesite broj n: ");
        scanf("%d", &n);

13

    if (n < 0) {
15        printf("Neispravan unos.\n");
        return -1;
17    }

19     printf("Unesite n brojeva: ");

21     /* Inicijalizuje se brojac sa kojim se kontrolise broj *
        ucitavanja - treba da ih bude tacno n. */
23     i = 0;
    while (i < n) {
25        /* Ucitava se broj. */
        scanf("%d", &x);
```

```
27      /* Proverava se da li broj negativan i neparan. */
29      if (x < 0 && x % 2 != 0) {
31          /* Ako jeste, dodajemo ga na zbir. */
32          zbir = zbir + x;
33      }
34
35      /* Uvecava se brojac iteracija. */
36      i++;
37  }
38
39  /* Ispisuje se rezultat. */
40  printf("%d\n", zbir);
41
42  return 0;
43 }
```

### Rešenje 2.3.14

```
1  #include <stdio.h>
2
3  int main()
4  {
5      int n, broj;
6      int suma = 0;
7      /* Brojac. */
8      int i;
9
10     printf("Unesite broj n: ");
11     scanf("%d", &n);
12
13     if (n < 0) {
14         printf("Neispravan unos.\n");
15         return -1;
16     }
17
18     printf("Unesite brojeve: ");
19     for (i = 0; i < n; i++) {
20         scanf("%d", &broj);
21
22         if (broj % 5 == 0 && broj % 7 != 0)
23             suma += broj;
24     }
25
26     printf("Suma je %d.\n", suma);
27
28     return 0;
29 }
```

### Rešenje 2.3.15

```
1 #include <stdio.h>
3 int main()
4 {
5     /* Promenljiva cena oznacava trenutno unesenu cenu. */
6     float cena;
7     float m;
8     int n, i;
9     int broj_brojeva = 0;
11    printf("Unesite koliko novaca ima Nikola: ");
12    scanf("%f", &m);
13
14    if (m < 0) {
15        printf("Nikola ne moze imati negativno novaca.\n");
16        return -1;
17    }
18
19    printf("Unesite broj artikala: ");
20    scanf("%d", &n);
21
22    if (n < 0) {
23        printf("Broj artikala ne moze biti negativan.\n");
24        return -1;
25    }
26
27    printf("Unesite cene artikala: ");
28
29    i = 0;
30    while (i < n) {
31        /* Ucitava se cena artikla. */
32        scanf("%f", &cena);
33
34        if (cena <= 0) {
35            printf("Cena ne moze biti negativna.\n");
36            return -1;
37        }
38
39        /* Provera da li je cena manji od zadatog broja m. */
40        if (cena < m) {
41            /* Ako jeste, uvecava se brojac brojeva za 1. */
42            broj_brojeva++;
43        }
44
45        i++;
46    }
47
48    printf("%d\n", broj_brojeva);
49
50    return 0;
```



51 }

## Rešenje 2.3.16

```

1  #include <stdio.h>
   int main()
3  {
   int x;
5
   /* U promenljivoj p se cuva proizvod. */
7  int p;
9
   /* Promenljiva u služi za proveru da li su brojevi uopšte
      uneseni. Na početku se pretpostavlja da nisu i postavlja se na
11  0. */
   int u = 0;
13
   /* Promenljiva unesen_pozitivan služi za proveru da li su
      pozitivni brojevi uopšte uneseni. Na početku se pretpostavlja
15  da nisu i postavlja se na 0. */
   int unesen_pozitivan = 0;
17
19  p = 1;
21
   /* Izraz 1 je konstantan, različit je od nule što znači da je to
      tačan izraz. Uslov petlje je uvek tačan! */
23  printf("Unesite brojeve:");
25
   while (1) {
       scanf("%d", &x);
27
       /* Proveravanje da li je uneta nula. */
29       if (x == 0)
31
           /* Naredba break prekida petlju. Izvršavanje se nastavlja od
              prve naredbe nakon petlje. */
33           break;
35
           /* Ako je makar 1 broj različit od 0 promenljiva u će biti
              postavljena na 1. */
37           u = 1;
39
           /* Ako je unet negativan broj, taj broj se ne množi sa ukupnim
              proizvodom p; zato se nastavlja dalje. */
41           if (x < 0)
43
               /* Naredba continue prekida trenutnu iteraciju petlje tako
                  što preskace sve naredbe koje nakon njega slede.
                  Izvršavanje se nastavlja od provere uslova petlje. */
45               continue;
47
           /* Ako je makar jedan broj pozitivan, promenljiva

```

## 2 Kontrola toka

```

        unesen_pozitivan se postavlja na 1. */
49     unesen_pozitivan = 1;
        p = p * x;
51 }

53 if (u == 0)
    printf("Nisu uneseni brojevi.\n");
55 else if (unesen_pozitivan == 0)
    printf("Nisu uneseni pozitivni brojevi. \n");
57 else
    printf("Proizvod pozitivnih unetih brojevi je %d.\n", p);
59
    return 0;
61 }
```

### Rešenje 2.3.17

```

1  #include <stdio.h>
   #include <stdlib.h>
3
   int main()
5  {
       int n, cifra;
       int indikator = 0;

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

13      if (n < 0)
           n = abs(n);

15
       /* Sve dok imamo cifara u zapisu broja. */
17      while (n > 0) {

19          /* Izdvajamo posledjnu cifru broja. */
           cifra = n % 10;

21
           /* Proveravamo da li je bas ona jednaka broju 5 */
23           if (cifra == 5) {
               /* Ako jeste postavljamo indikator na vrednost 1 tako da
25                  znamo da smo pronasli peticu i prekidamo sa izvrsavanjem
                   petlje. */
27               indikator = 1;
                   break;
29           }

           /* Ako izdvojena cifra nije jednaka broju 5, broj delimo sa 10
31          * kako bi mogli da izdvojimo i preostale cifre broja na isti
           * nacin.
33          */
           n = n / 10;

```

```
35 }
37 /* Ispisujemo rezultat */
38 if (indikator == 0) {
39     printf("Cifra 5 se ne nalazi u zapisu!\n");
40 } else {
41     printf("Cifra 5 se nalazi u zapisu!\n");
42 }
43
44 return 0;
45 }
```

## Rešenje 2.3.18

```
1  #include <stdio.h>
3  int main()
4  {
5      int x;
6      int broj_brojeva;
7      int suma;
8
9      broj_brojeva = 0;
10     suma = 0;
11
12     printf("Unesite brojeve: ");
13
14     while (1) {
15         /* Ucitavanje broja. */
16         scanf("%d", &x);
17
18         /* Ako je unesena 0, prekida se petlja. */
19         if (x == 0)
20             break;
21
22         /* Procitani broj dodaje se na sumu. */
23         suma += x;
24
25         /* I uvecava se broj ucitanih brojeva. */
26         broj_brojeva++;
27     }
28
29     if (broj_brojeva == 0)
30         printf("Nisu uneseni brojevi.\n");
31     else
32         /* Prilikom deljenja celih brojeva kao rezultat se dobija ceo
33            broj. Kako je aritmeticka sredina realan broj, potrebno je
34            izvesti konverziju prilikom deljenja da bi se dobio
35            ispravan rezultat. */
36         printf("Aritmeticka sredina: %.4f\n",
37                (double) suma / broj_brojeva);
38 }
```

```
39     return 0;
    }
```

### Rešenje 2.3.19

```
1  #include <stdio.h>
3  int main()
4  {
5      float cena;
6      int broj_artikla;
7      float suma;
9      broj_artikla = 0;
10     suma = 0;
11
12     printf("Unesite cene: ");
13
14     while (1) {
15         scanf("%f", &cena);
16
17         if (cena == 0)
18             break;
19
20         if (cena < 0) {
21             printf("Cena ne moze biti negativna.\n");
22             return -1;
23         }
24
25         suma += cena;
26
27         /* I uvecava se broj ucitanih brojeva. */
28         broj_artikla++;
29     }
30
31     if (broj_artikla == 0)
32         printf("Nisu unesene cene.\n");
33     else
34         printf("Aritmeticka sredina: %.4f\n", suma / broj_brojeva);
35
36     return 0;
37 }
```

### Rešenje 2.3.20

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

```
{
5   int n;
   /* Ucitavaju se dva broja, broj i sledbenik, i proverava se da li
7   su razlicitog znaka. */
   double broj, sledbenik;
   /* Brojac. */
9   int i;
   int broj_promena = 0;

13  printf("Unesite broj n ");
   scanf("%d", &n);

15  if (n < 0) {
17      printf("Neispravan unos.\n");
      return -1;
19  }

21  /* Prvo se proveara da li uopste ima unosa, i ako unosa nema,
      ispisuje se odgovarajuca poruka i izlazi iz programa. */
23  if (n == 0) {
      printf("Broj promena je 0.\n");
      return 0;
25  }
27

   printf("Unesite brojeve: ");
29  /* Pre petlje učitava se jedan broj, a u petlji se učitava njegov
      sledbenik i proverava se da li su razlicitog znaka. */
   scanf("%lf", &broj);

33  /* Kako je vec jedan broj unesen, brojac se postavlja na 1, a ne
      na 0. */
35  for (i = 1; i < n; i++) {
      /* Ucitava se sledbenik. */
37      scanf("%lf", &sledbenik);

39      /* Ako su razlicitog znaka proizvod je manji od 0. */
      if (sledbenik * broj < 0)
41          broj_promena++;
      /* Problem je ako je proizvod jednak 0. Tada mora proveriti da li
43      je jedan od brojeva negativan jer tada postoji promena
      znaka. */
      else if (sledbenik * broj == 0 && (sledbenik < 0 || broj < 0))
45          broj_promena++;

47      /* Tekuci sledbenik postaje tekuci broj, a u sledecoj iteraciji
      petlje se učitava sledeci sledbenik. */
49      broj = sledbenik;
51  }

53  printf("Broj promena je %d.\n", broj_promena);

55  return 0;
```

```
}
```

### Rešenje 2.3.21

```
1 #include <stdio.h>
3 int main()
4 {
5     /* Broj artikala. */
6     int n;
7     /* Brojac. */
8     int i;
9     /* Cena trenutno unetnog artikla. */
10    float cena;
11    /* Minimalna cena. */
12    float min_cena;
13
14    printf("Unesite broj artikala:");
15    scanf("%d", &n);
16
17    if (n <= 0) {
18        printf("Neispravan unos\n");
19        return -1;
20    }
21
22    /* Prva cena se unosi iznad petlje kako bi bio njegova vrednost
23       bila dodeljena promenljivoj min_cena. Neophodno je da
24       promenljiva min bude inicijalizovana pre ulaska u petlju da bi
25       uslov x<min mogao da bude ispitan u prvoj iteraciji. */
26    printf("Unesite cenu artikala:");
27    scanf("%f", &cena);
28    /* Proveravamo da li je cena isprano uneta vrednost. */
29    if (cena <= 0) {
30        printf("Cena ne moze biti negativna.\n");
31        return -1;
32    }
33
34    min_cena = cena;
35    i = 0;
36    while (i < n - 1) {
37        scanf("%f", &cena);
38
39        if (cena <= 0) {
40            printf("Cena ne moze biti negativna.\n");
41            return -1;
42        }
43
44        /* Provera da li je uneta cena manja od tekuce minimalne cene. */
45        if (cena < min_cena)
46            min_cena = cena;
47        i++;
```

```

    }
49     printf("Minimalna cena je: %f\n", min_cena);
51     return 0;
53 }

```

### Rešenje 2.3.22

```

1  #include <stdio.h>
   #include <stdlib.h>
3
   int main()
5  {
   int n;
7  int x, x_desetica;
   int max_desetica, broj;
9  int i;

11  printf("Unesite broj n: ");
   scanf("%d", &n);
13
   if (n < 0) {
15     printf("Neispravan unos.\n");
       return -1;
17 }

19  if (n == 0) {
       printf("Nisu uneseni brojevi.\n");
21     return 0;
   }

23
   /* Maksimalna cifra desetice se postavlja na 0 jer 0 je
25    svakako najmanja cifra pa je pocetna vrednost neutralna tj. ne
       moze da utice na izracunavanje maksimuma. Ipak, treba biti
27    pazljiv jer nije uvek zgodno pretpostaviti da je maksimalna
       vrednost 0. Na primer, ako je zadatak naci maksimum celih
29    brojeva, a korisnik unese -32 -7 i -22, maksimalni je broj -7,
       sto je manje od 0. */
31  max_desetica = 0;

33  printf("Unesite brojeve: ");
   for (i = 0; i < n; i++) {
35     scanf("%d", &x);

37     /* Izdvajanje cifre desetice procitanog broja. */
       x_desetica = (abs(x) / 10) % 10;
39
       /* Provera da li je izdvojena cifra veca od trenutne
41        maksimalne cifre desetice. */
       if (x_desetica > max_desetica) {

```

## 2 Kontrola toka

```
43      /* Ako jeste vece, pamti se nova najveca cifra, kao i broj u
      kom se pojavila. */
45      max_desetica = x_desetica;
      broj = x;
47  }
  }
49
  printf("Broj sa najvecom cifrom desetica je %d\n", broj);
51
  return 0;
53 }
```

### Rešenje 2.3.23

```
1  #include <stdio.h>
   #include <stdlib.h>
3
   int main()
5  {
7      int n;
      int x, x_kopija, br_cifara;
9      int max_br_cifara, broj;
      int i;
11
      printf("Unesite broj n: ");
13      scanf("%d", &n);
15
      if (n < 0) {
          printf("Neispravan unos.\n");
17         return -1;
      }
19
      if (n == 0) {
          printf("Nisu uneseni brojevi.\n");
21         return 0;
      }
23
25      /* Maksimalan broj cifara se postavlja na 0, svaki broj ima vise
      od 0 cifara pa je ova vrednost neutralna. */
27      max_br_cifara = 0;
29
      printf("Unesite n brojeva: ");
      for (i = 0; i < n; i++) {
31         scanf("%d", &x);
33
35         /* Odredjivanje broja cifara unetog broja x. */
          x_kopija = abs(x);
          br_cifara = 0;
          while (x_kopija != 0) {
37             x_kopija = x_kopija / 10;
```



```

    br_cifara++;
39 }

41 /* Ako je broj cifara unetog broja veci od maksimalnog */
42 if (br_cifara > max_br_cifara) {
43     /* Cuvamo ga */
44     max_br_cifara = br_cifara;
45     /* I zbog ispisa rezultata, cuvamo i originalni broj */
46     /* Zbog ovoga smo morali i da racunamo broj cifara nad
47        kopijom broja x kako ne bismo promenili njegovu vrednost */
48     broj = x;
49 }
50 }
51
52 printf("Najvise cifara ima broj %d\n", broj);
53
54 return 0;
55 }

```

### Rešenje 2.3.24

```

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

4 int main()
5 {
6
7     int n;
8     int x, x_kopija;
9     int broj;
10    int vodeca_cifra, max_vodeca_cifra;
11    int i;
12
13    /* Citamo vrednost sa ulaza */
14    printf("Unesite broj n: ");
15    scanf("%d", &n);
16
17    /* Postavljamo maksimalnu vodecu cifru na 0 - cifre broja su vece
18       ili jednake od 0 pa je ova vrednost neutralna */
19    max_vodeca_cifra = 0;
20
21    /* Ucitavamo broj po broj */
22    printf("Unesite n brojeva: ");
23    for (i = 0; i < n; i++) {
24        scanf("%d", &x);
25
26        /* Odredjujemo vodecu cifru broja */
27        x_kopija = abs(x);
28        while (x_kopija > 10) {
29            x_kopija = x_kopija / 10;
30        }

```

```

    vodeca_cifra = x_kopija;
32
    /* Ako je izdvojena cifra veca od maksimalne vodece cifre */
34    if (vodeca_cifra > max_vodeca_cifra) {
        /* Cuvamo je */
36        max_vodeca_cifra = vodeca_cifra;
        /* I zbog ispisa, cuvamo i broj u kojem se ona pojavljuje */
38        /* Zbog ovoga smo morali i da racunamo vodecu cifru nad
            kopijom broja x kako ne bismo promenili njegovu vrednost */
40        broj = x;
    }
42 }

44 /* Ispisujemo rezultat */
    printf("%d\n", broj);
46
    return 0;
48
}
```

### Rešenje 2.3.25

```

1  #include <stdio.h>

3  int main()
{
5      int x;
    int min, max;

7

    printf("Unesite brojeve: ");

9

    /* Prvi broj se ucitava izvan petlje zbog inicijalizacije
11     maksimuma i minimuma. */
    scanf("%d", &x);
13     max = x;
    min = x;

15

    /* Sve dok se ne unese 0, ucitavaju se brojevi u petlji. */
17     while (x != 0) {

19         /* Provera da li je procitani broj veci od aktuelnog maksimuma.
            */
21         if (x > max)
            max = x;
23         /* Provera da li je procitani broj manji od aktuelnog minimuma.
            */
25         if (x < min)
            min = x;

27

        /* Ucitavanje narednog broja. */
29         scanf("%d", &x);
    }
```

```
    }  
31  
    printf("Razlika: %d\n", max - min);  
33  
    return 0;  
35 }
```

### Rešenje 2.3.26

```
1  #include <stdio.h>  
   #include <stdlib.h>  
3  
   int main()  
5 {  
   int n;  
   int d;  
7   /* Uzastopni brojevi za koje se racuna rastojanje. */  
   int x, y;  
   int broj_parova;  
11  int i;  
  
13  
   printf("Unesite brojeve n i d: ");  
15   scanf("%d %d", &n, &d);  
  
17   if (n < 0 || d < 0) {  
       printf("Neispravan unos.\n");  
19       return -1;  
   }  
21  
   broj_parova = 0;  
23  
   printf("Unesite n brojeva: ");  
25  
   /* Prvi broj se ucitava pre petlje. */  
27   scanf("%d", &x);  
  
29   for (i = 1; i < n; i++) {  
       scanf("%d", &y);  
31  
       /* Provera da li su brojevi na rastojanju d. */  
33       if (abs(y - x) == d)  
           broj_parova++;  
35  
       /* Broj iz tekuce iteracije se cuva kako bi mogao da se  
37          upotrebljava u narednoj iteraciji. */  
       x = y;  
39   }  
  
41   printf("Broj parova: %d\n", broj_parova);
```

```
43     return 0;
    }
```

### Rešenje 2.3.27

```
1  #include <stdio.h>
3  int main()
4  {
5      int x;
6      /* Tezina trenutne pozicije u broju. Moze biti 1, 10, 100, 1000
7         itd. */
8      int pozicija;
9      /* Trenutna izdvojena cifra iz broja x. */
10     int cifra;
11     /* Broj dobijen nakon transformacije. */
12     unsigned int y;
13
14     printf("Unesite broj: ");
15     scanf("%d", &x);
16
17
18     if (x <= 0) {
19         printf("Nekorektan unos.\n");
20         return -1;
21     }
22
23     /* Posto pocinjemo sa izdvajanjem cifara od cifre jedinica,
24        postavlja se tezinu (stepen) pozicije na 1. */
25     pozicija = 1;
26     y = 0;
27
28     /* Provera da li ima cifara u zapisu broja. */
29     while (x > 0) {
30
31         /* Izdvaja se poslednja cifra iz zapisa. */
32         cifra = x % 10;
33
34         /* Provera da li je cifra parna. */
35         if (cifra % 2 == 0) {
36             /* I ako jeste, uvecava se. */
37             cifra++;
38         }
39     }
40
41     /* Novi broj se formira tako sto se izdvojena cifra pomnozi
42        odgovarajucom tezinom (stepenom) pozicije. */
43     y += cifra * pozicija;
44
45     /* Priprema se broj za izdvajanje naredne cifre, uklanja se
46        poslednja cifra broja. */
```

```

47     x /= 10;

49     /* Uvecava se tezinu (stepen) pozicije. */
    pozicija *= 10;
51 }

53 /* Ispisuje se izracunatu vrednost. */
    printf("%d\n", y);
55
57     return 0;
}

```

### Rešenje 2.3.28

```

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

5  int main()
   {
7      int x;
      /* Tezina trenutne pozicije u broju. Moze biti 1, 10, 100, 1000
9         itd. */
      int stepen_deset;
11     /* Trenutna izdvojena cifra iz broja x. */
      int cifra;
13     /* Redni broj cifre koja se trenutno obradjuje, gledano s desna
        na levo. */
15     int rbr;
      /* Broj dobijen nakon transformacije. */
17     int y;
      /* Promenljiva znak cuva znak unesenog broja. Moze biti -1 za
19         negativnu vrednost ili 1 za pozitivnu vrednost. */
      int znak = 1;

21
      /* Ucitavanje broja. */
23     printf("Unesite broj: ");
      scanf("%d", &x);
25
27     if (x <= 0) {
        x = abs(x);
        znak = -1;
29     }
      /* Postavlja se vrednost stepena na 0 - to znaci da se prvo mnozi
31         sa 10^0=1. */
      stepen_deset = 0;

33
      /* Postavlja se vrednost broja koji se formira na 0. */
35     y = 0;
      /* Postavlja se redni broj pozicije na 0. */
37     rbr = 0;

```

```
39  /* Provera da li ima cifara u zapisu broja. */
    while (x > 0) {
41
43      /* Izdvajanje cifre. */
        cifra = x % 10;

45      /* Proverava se da li je pozicija izdvojene cifre parna - cifre
        na parnim pozicijama se zadržavaju. */
47      if (rbr % 2 == 0) {
49          /* Ako jeste parna izdvojena cifra se dodaje novom broju.
            Neophodno je izvršiti promenu tipova, jer je double
            povratni tip funkcije pow. */
51          y += cifra * ((int) pow(10, stepen_deset));

53          /* Uvecava se stepen zbog naredne cifre. */
            stepen_deset++;
55      }

57      /* Azurira se redni broj cifre. */
        rbr++;
59      /* I priprema se broj za naredno izdvajanje. */
        x /= 10;
61    }

63    y = znak * y;

65    /* Ispisuje se rezultat. */
    printf("%d\n", y);
67
69    return 0;
}
```

### Rešenje 2.3.29

```
1  #include <stdio.h>

3  int main()
    {
5      int n, novo_n;
        int stepen;
7      int cifra_levo, cifra_sredina, cifra_desno;

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

13     if (n <= 0) {
        printf("Neispravan unos.\n");
15     return -1;
    }
}
```

```
17  /* Stepen broja 10 sa kojim se mnoze cifre izdvojenog broja. */
19  stepen = 1;

21  /* Nova vrednost broja. */
   novo_n = 0;

23  /* Provera da li u zapisu broja postoje barem tri cifre. */
25  while (n > 99) {
   /* Izdvaja se srednja cifra, cifra desno od nje i cifra levo od
27     nje: npr. za trojku 583 8 je srednja cifra, 3 je cifra
       desno, a 5 cifra levo. */
29     cifra_desno = n % 10;
       cifra_sredina = (n / 10) % 10;
31     cifra_levo = (n / 100) % 10;

33     /* U novi broj se smesta desna cifra. */
       novo_n += cifra_desno * stepen;

35     /* Azurira se vrednost stepena. */
37     stepen = stepen * 10;

39     /* Provera da li je srednja cifra jednaka zbiru leve i desne
       cifre. */
41     if (cifra_levo + cifra_desno == cifra_sredina) {

43         /* Treba izbaciti srednju cifru, pa broj n se azurira tako
           sto se podeli sa 100. */
45         n = n / 100;
       } else {

47         /* Inace, zadrzava se srednja cifra i odbacuje se samo
49            poslednja. */
           n = n / 10;
51     }
   }

53  /* Na novi broj se dodaje preostali dvocifreni ili jednocifreni
55     broj. */
   novo_n = n * stepen + novo_n;

57  /* Ispisivanje rezultata. */
59  printf("%d\n", novo_n);

61  return 0;
63 }
```

## Rešenje 2.3.30

```
1 #include <stdio.h>
2 #include <math.h>
3
4
5 int main()
6 {
7     int x;
8     int broj_cifara;
9     int min_stepen, max_stepen;
10    int pom;
11    int leva_cifra, desna_cifra;
12    int indikator;
13
14    printf("Unesite broj: ");
15    scanf("%d", &x);
16
17    /* Ako je korisnik uneo negativan broj, analizira se njegova
18       apsolutna vrednost. */
19    if (x < 0)
20        x = -x;
21
22    /* Odredjuje se broj cifara u zapisu broja x da bi moglo da se
23       izdvajaju istovremeno cifre i sa leve i sa desne strane. */
24    broj_cifara = 0;
25    pom = x;
26    while (pom > 0) {
27        pom /= 10;
28        broj_cifara++;
29    }
30
31    /* Odredjuje se stepen koji stoji uz krajnju levu cifru broja. */
32    max_stepen = (int) pow(10, broj_cifara - 1);
33
34    /* Indikator je promenljiva koja ukazuje da li je broj palindrom
35       ili ne. */
36    indikator = 1;
37    while (x != 0 && indikator == 1) {
38        /* Izdvaja se leva cifra. */
39        leva_cifra = x / max_stepen;
40        /* Izdvaja se desna cifra. */
41        desna_cifra = x % 10;
42        /* Ako su cifre razlicite, odmah se moze zakljuciti da broj
43           nije palindrom i prekida se izvršavanje petlje. */
44        if (leva_cifra != desna_cifra) {
45            indikator = 0;
46            break;
47        }
48        /* Formira se nova vrednost broja x tako sto se odbacuje
49           krajnja leva i krajnja desna cifra. */
50        x = (x % max_stepen - x % 10) / 10;
51        /* Koriguje se maksimalan stepen tako dobijenog broja - deli se
52           sa 100 jer su odbacene 2 cifre. */
```



```
53     max_stepen = max_stepen / 100;
54 }
55
56 /* Ispisuje se rezultat. */
57 if (indikator == 1)
58     printf("Broj je palindrom!\n");
59 else
60     printf("Broj nije palindrom!\n");
61
62 return 0;
63 }
```

## Rešenje 2.3.31

```
1  #include <stdio.h>
2
3  int main()
4  {
5      /* Pamtimo uzastopna dva Fibonacijeva broja i na osnovu njih
6         racunamo sledeci. */
7      /* Promenljive prvi i drugi su brojevi koje pamtimo i na osnovu
8         njih racunamo treci. */
9      /* Na osnovu teksta zadatka, promenljive prvi i drugi postavljamo
10         na 1. */
11     int prvi = 1;
12     int drugi = 1;
13     int treci;
14     /* Promenljiva pozicija je podatak koji ucitavamo i odnosi se na
15        poziciju u Fibonacijevom nizu za koju treba izracunati
16        vrednost. */
17     int pozicija;
18     /* Promenljiva i oznacava do koje pozicije smo izracunali
19        vrednosti. Kako imamo prve dve vrednosti, ovu promenljivu
20        postavljamo na 2. */
21     int i = 2;
22
23     printf("Unesite poziciju u Fibonacijevom nizu: ");
24     scanf("%d", &pozicija);
25
26     /* Pozicija ne moze biti 0 i ne moze biti negativan broj. */
27     if (pozicija < 1) {
28         printf
29             ("Neispravan unos. Pozicija u Fibonacijevom nizu mora biti
30              pozitivan broj koji nije 0!\n");
31         return -1;
32     }
33
34     while (i < pozicija) {
35         /* Na osnovu dva uzastopna racunamo treci. */
36         treci = prvi + drugi;
```

## 2 Kontrola toka

```
37      /* Potom razmenjujemo vrednosti. Uzastopna dva koja pamtimo
38         postaju sledeca uzastopna dva broja Fibonacijevog niza. */
39      prvi = drugi;
40      drugi = treci;
41
42      /* Prelazimo na racunanje sledeceg broja na sledecoj poziciji. */
43      i++;
44  }
45
46  printf("Trazeni broj je: %d\n", drugi);
47
48  return 0;
49 }
```

### Rešenje 2.3.32

```
#include<stdio.h>
2
3 int main()
4 {
5     int a0;
6     int an, an1;
7     printf("Unesi pocetni clan niza brojeva:");
8     scanf("%d", &a0);
9     if (a0 <= 0) {
10         printf("Nekorektan unos. Broj mora biti pozitivan.\n");
11         return -1;
12     }
13     printf("%d\n", a0);
14     an = a0;
15     while (an != 1) {
16         if (an % 2) {
17             /* Ukoliko je vrednost izraza an%2 razlicita od nule, izraz
18                se tumaci kao tacan i izvrsavaju se naredbe iz if grane. */
19             an1 = (3 * an + 1) / 2;
20         } else {
21             /* U suprotnom, ukoliko je vrednost izraza an%2 jednaka nuli,
22                izraz se tumaci kao netacan i izvrsavaju se naredbe iz
23                else grane. */
24             an1 = an / 2;
25         }
26         printf("%d\n", an1);
27         an = an1;
28     }
29     return 0;
30 }
```

### Rešenje 2.3.33

```

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

3
   int main()
5  {
       int format;
7      /* Pomocna promenljiva koja služi kao brojac u petlji. */
       int i;
9      /* Trenutne vrednosti za sirinu i visinu i pomocna promenljiva za
          promene u petlji. */
12     double sirina, duzina, nova_duzina;
       unsigned int konacna_sirina, konacna_duzina;

13
       printf("Uneti format papira: ");
15     scanf("%d", &format);

17     if (format <= 0) {
         printf("Neispravan unos.\n");
19         return -1;
       }

21
       /* duzina/sirina = 1 : sqrt(2) duzina*sirina = 1000x1000mm^2 Na
          osnovu ovih odnosa dobijamo pocetnu vrednost za sirinu i
23         duzinu, odnosno vrednosti za papir A0. */
       duzina = sqrt(1000 * 1000 / sqrt(2));
25     sirina = sqrt(2) * duzina;

27
       /* Kako vec imamo odredjenu sirinu i duzinu za papir A0, petlju
          krecemo od izracunavanja za papir A1, pa brojac i postavljamo
29         na 1. */
       for (i = 1; i <= format; i++) {
31         nova_duzina = sirina / 2;
33         sirina = duzina;
           duzina = nova_duzina;
35     }

37     /* Duzina i sirina celi brojevi. */
       konacna_sirina = (unsigned int) sirina;
39     konacna_duzina = (unsigned int) duzina;

41     printf("%u %u\n", konacna_duzina, konacna_sirina);

43     return 0;
   }

```

### Rešenje 2.3.34

```

1  #include <stdio.h>

3  int main()

```

```
{
5   char c;

7   /* Funkcija getchar učitava jedan karakter. Naredbom dodele
   (c=getchar()) promenljivoj c bice dodeljena vrednost ascii
9   koda unetog karaktera. Obratiti posebnu paznju na zagrade. */

11  while ((c = getchar()) != '.') {
    if (c >= 'A' && c <= 'Z')
13      /* Razlika izmedju ascii koda svakog malog i odgovarajuceg
        velikog slova je konstanta koja se moze sracunati izrazom
        'a'-'A' (i iznosi 32). */
        putchar(c + 'a' - 'A');
17      else if (c >= 'a' && c <= 'z')
        putchar(c - 'a' + 'A');
19      else
        putchar(c);
21  }

23  return 0;
}
```

### Rešenje 2.3.35

```
#include <stdio.h>

2
int main()
4 {
    char c;

6
    /* Inicijalizacija brojaca na 0. */
8    int br_v = 0;
    int br_m = 0;
10   int br_c = 0;
    int br_b = 0;
12   int br_k = 0;
    int suma = 0;

14
    /* Petlja se završava kada korisnik ne unese karakter, vec zada
16   konstantu EOF . Ova konstanta se zadaje kombinacijom tastera
    CTRL+D. U tom slucaju, getchar() vraca -1. */
18   while ((c = getchar()) != EOF) {
    if (c >= 'A' && c <= 'Z')
20       br_v++;
    else if (c >= 'a' && c <= 'z')
22       br_m++;
    else if (c >= '0' && c <= '9') {
24       br_c++;
        /* Kada od promenljive tipa char oduzimamo karakter (ili neku
26       drugu promenljivu tipa char), zapravo se vrsi oduzimanje
        njihovih ascii vrednosti i dobija se broj. */
    }
  }
}
```

```

28     suma = suma + c - '0';
    } else if (c == '\t' || c == '\n' || c == ' ')
30         br_b++;

32     br_k++;
    }

34     printf("velika: %d, mala: %d, cifre: %d, beline: %d \n", br_v,
36             br_m, br_c, br_b);
    printf("suma cifara: %d\n", suma);

38     return 0;
40 }

```

### Rešenje 2.3.36

```

#include <stdio.h>

2 int main()
{
4     /* Promenljiva i je brojac. */
    int n, i;
6     /* Brojaci za svaki od samoglasnika. */
    int broj_a = 0, broj_e = 0, broj_i = 0, broj_o = 0, broj_u = 0;
8     /* Promenljiva c je tekuci ucitani karakter. */
    char c, belina;

10     printf("Unesite broj n: ");
    scanf("%d", &n);

12     if (n < 0) {
14         printf("Neispravan unos.\n");
16         return -1;
18     }

20     for (i = 0; i < n; i++) {
22         /* Prvo se cita belina koja se nalazi nakon prethodnog unosa,
            pa tek posle procitane beline se cita uneseni karakter. */
        scanf("%c%c", &belina, &c);

24         /* Provera da li je ucitani karakter samoglasnik. */
        switch (c) {
26             case 'a':
28             case 'A':
                broj_a++;
                break;
30             case 'e':
32             case 'E':
                broj_e++;
                break;
34             case 'i':

```

```
36     case 'I':
37         broj_i++;
38         break;
39     case 'O':
40     case 'o':
41         broj_o++;
42         break;
43     case 'u':
44     case 'U':
45         broj_u++;
46         break;
47     }
48 }

50 printf("samoglasnik a: %d\n", broj_a);
51 printf("samoglasnik e: %d\n", broj_e);
52 printf("samoglasnik i: %d\n", broj_i);
53 printf("samoglasnik o: %d\n", broj_o);
54 printf("samoglasnik u: %d\n", broj_u);

56 return 0;
}
```

### Rešenje 2.3.37

```
1  /* Sa standardnog ulaza se unosi ceo broj n, a zatim i n karaktera.
2     Napisati program koji proverava da li se od unetih karaktera
3     moze napisati rec Zima. */

5  #include <stdio.h>
6  #include <math.h>

7
8  int main()
9  {
10     int n;
11     int broj_Z, broj_i, broj_m, broj_a;
12     char novi_red, c;
13     int i;

14
15     broj_Z = 0;
16     broj_i = 0;
17     broj_m = 0;
18     broj_a = 0;

19
20     printf("Unesite broj: ");
21     scanf("%d", &n);

22
23     /* Ucitavanje karakter po karakter. */
24     for (i = 0; i < n; i++) {
25         printf("Unestite %d. karakter: ", i + 1);
26         /* Prvo se cita belina koja se nalazi nakon prethodnog unosa,
```

```

27     pa tek posle procitane beline se cita uneseni karakter. */
scanf("%c%c", &novi_red, &c);

29

/* Analiziramo karakter */
31 switch (c) {
    case 'Z':
33         broj_Z++;
        break;
35     case 'i':
        broj_i++;
37         break;
        case 'm':
39         broj_m++;
        break;
41     case 'a':
        broj_a++;
43         break;
    }
45 }

47 /* Ako u unosu ima barem jedno veliko slovo z i barem po jedno
    malo slovo i, m i a, rec se moze napisati. A u suprotnom ne
49 moze. */
    if (broj_Z && broj_i && broj_m && broj_a) {
51         printf("Moze se napisati rec Zima.\n");
    } else {
53         printf("Ne moze se napisati rec Zima.\n");
    }

55     return 0;
57 }

```

### Rešenje 2.3.38

```

1  #include <stdio.h>
    int main()
3  {
        int n;
5      /* Brojac. */
        int i;
7      /* Promenljiva u kojoj se cuva suma kubova. */
        int s;

9

        printf("Unesite pozitivan ceo broj:");
11     scanf("%d", &n);
        if (n < 0) {
13         printf("Neispravan unos.\n");
            return -1;
15     }

17     for (s = 0, i = 1; i <= n; i++)

```

## 2 Kontrola toka

---

```
    s += i * i * i;
19
    printf("Suma kubova od 1 do %d: %d\n", n, s);
21
    return 0;
23 }
```

### Rešenje 2.3.39

```
#include <stdio.h>
2
int main()
4 {
    int n;
6    /* Brojac. */
    int i;
8    /* Promenljiva u kojoj se cuva suma kubova. */
    int s;
10
    printf("Unesite pozitivan ceo broj:");
12    scanf("%d", &n);

14    if (n < 0) {
        printf("Neispravan unos.\n");
16        return -1;
    }
18
    i = 1;
20    s = 0;

22    for (i = 1; i <= n; i++) {
        s += i * i * i;
24        printf("i=%d, s=%d\n", i, s);
    }
26
    return 0;
28 }
```

### Rešenje 2.3.40

```
1 #include <stdio.h>

3 int main()
{
5     int n, i;
    float x, S, stepen;
7
    printf("Unesite redom brojeve x i n: ");
9     scanf("%f %d", &x, &n);
```



```

11  if (n < 0) {
12      printf("Neispravan unos.\n");
13      return -1;
14  }
15
16  /* Inicijalizacija sume. */
17  S = 0;
18
19  /* Stepen promenljiva ce sadrzati vrednosti stepena x^n. Pocetna
20     vrednost joj je 1 jer je x^0 = 1. */
21  stepen = 1;
22
23  for (i = 1; i <= n; i++) {
24      stepen = stepen * x;
25      S = S + i * stepen;
26  }
27
28  printf("S=%f\n", S);
29
30  return 0;
31 }

```

### Rešenje 2.3.41

```

1  #include <stdio.h>
2
3  int main()
4  {
5      unsigned n, i;
6      float x, S, stepen;
7
8      printf("Unesite redom brojeve x i n: ");
9      scanf("%f %u", &x, &n);
10
11     S = 1;
12     stepen = 1;
13     for (i = 1; i <= n; i++) {
14         stepen = stepen * x;
15         S = S + 1 / stepen;
16     }
17
18     printf("S=%f\n", S);
19
20     return 0;
21 }

```

### Rešenje 2.3.42

```
1 #include <stdio.h>
2 #include <math.h>
3
4 int main()
5 {
6     /* Promenljiva i je brojac, promenljiva S cuva izracunatu sumu, a
7        promenljiva clan je tekuci clan niza. */
8     int i;
9     float S;
10    float x, eps;
11    float clan;
12
13    printf("Unesite x: ");
14    scanf("%f", &x);
15
16    printf("Unesite tacnost eps: ");
17    scanf("%f", &eps);
18
19    S = 0;
20    /* Prvi clan sume je 1. */
21    clan = 1;
22    i = 1;
23    while (clan > eps) {
24        S = S + clan;
25        clan = clan * x / i;
26        i++;
27    }
28
29    printf("S=%f\n", S);
30
31    return 0;
32 }
```

### Rešenje 2.3.43

```
1 #include <stdio.h>
2 #include <math.h>
3
4 int main()
5 {
6     /* Promenljiva i je brojac, promenljiva S cuva izracunatu sumu,
7        promenljiva znak moze biti 1 ili -1 i odredjuje znak trenutnog
8        clana sume, a promenljiva clan je tekuci clan niza. */
9     int i, znak;
10    float S;
11    float x, eps, clan;
12
13    printf("Unesite x: ");
14    scanf("%f", &x);
```

```

16 printf("Unesite tacnost eps: ");
   scanf("%f", &eps);
18
20 S = 0;
   clan = 1;
22 i = 1;
   znak = -1;
24
   /* Kako clanovi sume mogu biti negativni, potrebno je posmatrati
26    apsolutnu vrednost clana. */
   while (fabs(clan) > eps) {
28     S = S + clan;

30     /* Promena znaka. */
     clan = clan * x / i;
32     clan *= znak;

34     i++;
   }
36
   printf("S=%f\n", S);
38
   return 0;
40 }

```

### Rešenje 2.3.44

```

1  #include <stdio.h>
   #include <math.h>
3
   int main()
5  {
   int n, i;
7  double x;
   double stepen = 1;
9  double proizvod = 1;

11 printf("Unesite redom brojeve x i n: ");
   scanf("%lf %d", &x, &n);
13
   if (n <= 0) {
15     printf("Neispravan unos.\n");
     return -1;
17   }

19   for (i = 0; i < n; i++) {
     stepen *= x;
21     proizvod *= 1 + cos(stepen);
   }
23

```

## 2 Kontrola toka

---

```
    printf("Proizvod = %lf\n", proizvod);
25
    return 0;
27 }
```

### Rešenje 2.3.45

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

### Rešenje 2.3.46

```
1  #include <stdio.h>
3  int main()
4  {
5      /* Promenljiva i je brojac, promenljiva S cuva izracunatu sumu,
6         promenljiva znak moze biti 1 ili -1 i odredjuje znak trenutnog
7         clana sume, a promenljiva clan je tekuci clan niza. */
8      int i, znak, n;
9      float S;
10     float x, clan;
11 }
```

```

13 printf("Unesite x i n: ");
   scanf("%f%d", &x, &n);

15 if (n <= 0) {
   printf("Neispravan unos.\n");
17   return -1;
   }

19

21 S = 0;
   clan = 1;
23 i = 1;
   znak = -1;

25
   /* Kako clanovi sume mogu biti negativni, potrebno je posmatrati
27   apsolutnu vrednost clana. */
   while (i <= 2 * n) {
29     S = S + clan;

31     /* Promena znaka. */
     /* Svaki clan suma se od prethodnog clana razlikuje za
33      $x^2/(i*(i+1))$ . */
     clan = clan * x * x / (i * (i + 1));
35     clan *= znak;

37     i += 2;
   }

39 printf("S=%f\n", S);

41 return 0;
43 }

```

### Rešenje 2.3.47

```

1 #include <stdio.h>

3 int main()
{
5   int n, i;
   /* Promenljiva clan je deo proizvoda i predstavlja 1/i!. */
7   double clan;
   double S = 1;

9

   printf("Unesite prirodan broj: ");
11  scanf("%d", &n);

13  if (n <= 1) {
   printf("Neispravan unos.\n");
15  return -1;
   }
}

```

## 2 Kontrola toka

---

```
17     clan = 1;
19     for (i = 2; i <= n; i++) {
21         clan = clan / i;
21         S *= 1 + clan;
21     }
23
23     printf("S = %lf\n", S);
25
25     return 0;
27 }
```

### Rešenje 2.3.48

```
1  #include <stdio.h>
3
3  int main()
3  {
5      int n, i, znak = -1;
5      /* Promenljiva clan je deo proizvoda i predstavlja 1*3*5*...*i. */
7      long int clan;
7      long int S = 0;
9
9      printf("Unesite prirodan broj: ");
11     scanf("%d", &n);
13
13     if (n < 5 || n % 2 == 0) {
13         printf("Neispravan unos.\n");
15         return -1;
15     }
17
17     clan = 1 * 3;
19     for (i = 5; i <= n; i += 2) {
19         clan = znak * clan * i;
21         S += clan;
21     }
23
23     printf("S = %ld\n", S);
25
25     return 0;
27 }
```

### Rešenje 2.3.49

```
1  #include <stdio.h>
2
2  int main()
4  {
4      int n, i;
```

```

6  double P;
   double x, a;

8

10 printf("Unesite dva relana broja x i a: ");
   scanf("%lf%lf", &x, &a);

12 printf("Unesite prirodan broj: ");
   scanf("%d", &n);

14

16 if (n <= 0) {
17     printf("Neispravan unos.\n");
18     return -1;
19 }

20 P = x;
   for (i = 0; i < n; i++)
22     P = (P + a) * (P + a);

24 printf("Izraz = %lf\n", P);

26 return 0;
}

```

### Rešenje 2.3.50

Rešenje (a)

```

#include <stdio.h>

2
int main()
4 {
   unsigned int n, i, j;

6

   printf("Unesite broj n: ");
   scanf("%u", &n);

8

10 for (i = 1; i <= n; i++) {
11     for (j = 1; j <= n; j++)
12         /* U tablici mnozenja vrednost svakog polja je proizvod vrste
13            i kolone u kojoj se nalazi. */
14         printf("%3d ", i * j);

16     printf("\n");
17 }

18 return 0;
20 }

```

Rešenje (b)

## 2 Kontrola toka

---

```
1  #include <stdio.h>
2
3  int main()
4  {
5      unsigned int n, i, j;
6
7      printf("Unesite broj n: ");
8      scanf("%u", &n);
9
10     /* Brojac koji broji koliko brojeva je ispisano u jednom redu. */
11     j = 0;
12     for (i = 1; i <= n * n; i++) {
13         printf("%3d ", i);
14         /* Uvecavamo brojac */
15         j++;
16
17         /* Kada je ispisano n brojeva u jednom redu, ispisuje se znak
18            za novi red, da bi ispis krenuo u novom redu i vrednost
19            brojaca j se postavlja na 0 jer u novom redu jos ni jedan
20            broj nije ispisan. */
21         if (j == n) {
22             j = 0;
23             printf("\n");
24         }
25     }
26
27     return 0;
28 }
```

Rešenje (c)

```
1  #include <stdio.h>
2
3  int main()
4  {
5      unsigned int n, i, j;
6
7      printf("Unesite broj n: ");
8      scanf("%u", &n);
9
10     for (i = 1; i <= n; i++) {
11         for (j = 0; j < n; j++)
12             if ((j + i) % n == 0)
13                 printf("%3d", n);
14             else
15                 printf("%3d", (j + i) % n);
16
17         printf("\n");
18     }
19 }
```



```
20 | return 0;  
    | }
```

Rešenje (d)

```
1 | #include <stdio.h>  
  
3 | int main()  
  | {  
5 |     unsigned int n, i, j;  
  
7 |     printf("Unesite broj n: ");  
  |     scanf("%u", &n);  
  
9 |     for (i = 0; i < n; i++) {  
11 |         for (j = 0; j < n - i; j++)  
  |             printf("(%d, %d)", i, j);  
13 |         printf("\n");  
15 |     }  
  
17 |     return 0;  
  | }
```

### Rešenje 2.3.51

Rešenje (a)

```
1 | #include <stdio.h>  
  
2 |  
3 | int main()  
4 | {  
5 |     unsigned int n, i, j;  
  
6 |  
7 |     printf("Unesite broj n: ");  
8 |     scanf("%u", &n);  
  
10 |    for (i = 0; i < n; i++) {  
11 |        /* Kvadrat predstavlja tabelu sa n vrsta gde svaka vrsta sadrzi  
12 |           n polja, a svako polje je isto i predstavlja karakter *. */  
13 |        for (j = 0; j < n; j++)  
14 |            printf("*");  
15 |        printf("\n");  
16 |    }  
  
18 |    return 0;  
  | }
```

## 2 Kontrola toka

---

### Rešenje (b)

```
1  #include <stdio.h>
3  int main()
4  {
5      unsigned int n, i, j;
7      printf("Unesite broj n: ");
8      scanf("%u", &n);
9
10     for (i = 0; i < n; i++) {
11         /* Kvadrat predstavlja tabelu sa n vrsta gde su ivice karakter
12          * , a unutrašnjost kvadrata je karakter blanko. */
13         for (j = 0; j < n; j++)
14             /* Provera da li je ivica. */
15             if (j == 0 || j == n - 1 || i == 0 || i == n - 1)
16                 printf("*");
17             else
18                 printf(" ");
19         printf("\n");
20     }
21
22     return 0;
23 }
```

### Rešenje (c)

```
1  #include <stdio.h>
3  int main()
4  {
5      unsigned int n, i, j;
7      printf("Unesite broj n: ");
8      scanf("%u", &n);
9
10     for (i = 0; i < n; i++) {
11         /* Kvadrat predstavlja tabelu sa n vrsta gde su ivice karakter
12          * , a unutrašnjost kvadrata je karakter blanko osim na
13          * mestima na kojima je glavna dijagonala. */
14         for (j = 0; j < n; j++)
15             /* Provera da li je ivica ili glavna dijagonala. */
16             if (j == 0 || j == n - 1 || i == 0 || i == n - 1 || i == j)
17                 printf("*");
18             else
19                 printf(" ");
20         printf("\n");
21     }
22 }
```

```

23     return 0;
    }

```

### Rešenje 2.3.52

```

1  #include <stdio.h>
3  int main()
4  {
5      unsigned int n, i, j;
7      printf("Unesite broj n: ");
8      scanf("%u", &n);
9
10     for (i = 0; i < n; i++) {
11         /* Veliko slovo X se moze posmatrati kao dijagonale kvadrata
12            (glavna i sporedna). Zato, treba ispisivati blanko na
13            mestima gde nije dijagonala, a karakter * na mestima gde je
14            neka od dijagonala. */
15         for (j = 0; j < n; j++)
16             /* Provera da li je mesto glavne ili sporedne dijagonale. */
17             if (i == j || i + j == n - 1)
18                 printf("*");
19             else
20                 printf(" ");
21         printf("\n");
22     }
23
24     return 0;
25 }

```

### Rešenje 2.3.53

```

1  #include <stdio.h>
3  int main()
4  {
5      unsigned int n, i, j;
7      printf("Unesite broj n: ");
8      scanf("%u", &n);
9
10     if (n % 2 == 0) {
11         printf("Pogresan unos.\n");
12         return -1;
13     }
14
15     for (i = 0; i < n; i++) {
16         for (j = 0; j < n; j++)

```

## 2 Kontrola toka

---

```
17     if (i == n / 2 || j == n / 2)
18         printf("+");
19     else
20         printf(" ");
21     printf("\n");
22 }
23
24 return 0;
25 }
```

### Rešenje 2.3.54

#### Rešenje (a)

```
#include <stdio.h>
2
int main()
4 {
    unsigned int n, i, j;
6
    printf("Unesite broj n: ");
8    scanf("%u", &n);
10
    for (i = 0; i < n; i++) {
        for (j = 0; j < n - i; j++)
12             printf("*");
        printf("\n");
14    }
16
    return 0;
17 }
```

#### Rešenje (b)

```
1 #include <stdio.h>
2
3 int main()
4 {
5     unsigned int n, i, j;
6
7     printf("Unesite broj n: ");
8     scanf("%u", &n);
9
10    for (i = 0; i < n; i++) {
11        for (j = 0; j <= i; j++)
12            printf("*");
13        printf("\n");
14    }
15 }
```

```
15     return 0;
17 }
```

*Rešenje (c)*

```
1  #include <stdio.h>
3  int main()
4  {
5      unsigned int n, i, j;
7      printf("Unesite broj n: ");
8      scanf("%u", &n);
9
10     for (i = 0; i < n; i++) {
11         /* Prvo se ispisuju beline koje prethode karakterima *. */
12         for (j = 0; j < i; j++)
13             printf(" ");
14         /* Posle belina se ispisuje potreban broj karaktera *. */
15         for (j = 0; j < n - i; j++)
16             printf("*");
17         printf("\n");
18     }
19
20     return 0;
21 }
```

*Rešenje (d)*

```
1  #include <stdio.h>
3  int main()
4  {
5      unsigned int n, i, j;
7      printf("Unesite broj n: ");
8      scanf("%u", &n);
9
10     /* Brojac i odredjuje koji red slike se trenutno ispisuje. */
11     for (i = 0; i < n; i++) {
12         /* Prvo se ispisuju beline koje prethode karakterima *. */
13         for (j = 0; j < n - i - 1; j++)
14             printf(" ");
15         /* Posle belina se ispisuje potreban broj karaktera *. */
16         for (j = 0; j <= i; j++)
17             printf("*");
18         printf("\n");
19     }
```

## 2 Kontrola toka

---

```
21     return 0;
    }
```

*Rešenje (e)*

```
#include <stdio.h>

2
int main()
4 {
    unsigned int n, i, j;

6
    printf("Unesite broj n: ");
    scanf("%u", &n);

10
    /* Brojac i odredjuje koji red slike se trenutno ispisuje. */
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 *. */
16        for (j = 0; j <= i; j++)
            printf("*");
18        printf("\n");
20    }

22    /* Potrebno je iscrtati i donji deo 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. */
24
26    for (i = 1; i < n; i++) {
        /* Prvo se ispisuju beline koje prethode karakterima *. */
28        for (j = 0; j < i; j++)
            printf(" ");
        /* Posle belina se ispisuje potreban broj karaktera *. */
30        for (j = 0; j < n - i; j++)
            printf("*");
32        printf("\n");
34    }

36    return 0;
    }
```

*Rešenje (f)*

```
1 #include <stdio.h>

3 int main()
```

```

5   unsigned int n, i, j;
6   char c, blanko;
7
8   printf("Unesite broj n: ");
9   scanf("%u", &n);
10
11  printf("Unesite karakter c: ");
12  /* Zbog pritiskanja tastera ENTER nakon unosa promenljive broj
13     potrebno je učitati karakter za novi red u promenljivu blanko
14     pre učitavanja karaktera kojim se iscrtava trougao. */
15  scanf("%c%c", &blanko, &c);
16
17  /* Brojac i odredjuje koji red slike se trenutno ispisuje. */
18  for (i = 0; i < n; i++) {
19      /* Iscrtavaju se samo ivice trougla, ostalo se popunjava
20         belinama. */
21      for (j = 0; j <= i; j++)
22          if (i == n - 1 || j == 0 || j == i)
23              printf("%c", c);
24          else
25              printf(" ");
26      printf("\n");
27  }
28
29  return 0;
30 }

```

### Rešenje 2.3.55

Rešenje (a)

```

1  #include <stdio.h>
2
3  int main()
4  {
5      unsigned int n, i, j;
6
7      printf("Unesite broj n: ");
8      scanf("%u", &n);
9
10     /* Brojac i odredjuje koji red slike se trenutno ispisuje. */
11     for (i = 0; i < n; i++) {
12         /* Prvo se ispisuju beline koje prethode karakterima *. */
13         for (j = 0; j < n - i - 1; j++)
14             printf(" ");
15         /* Posle belina se ispisuje potreban broj karaktera *. */
16         for (j = 0; j < 2 * i + 1; j++)
17             printf("*");
18         printf("\n");
19     }
20 }

```

## 2 Kontrola toka

---

```
19 }
21 return 0;
22 }
```

### Rešenje (b)

```
#include <stdio.h>

2 int main()
3 {
4     unsigned int n, i, j;
5
6     printf("Unesite broj n: ");
7     scanf("%u", &n);
8
9     /* Brojac i odredjuje koliko redova se ispisuje. Radi lakseg
10      izracunavanja koliko zvezdica i praznina je potrebno ispisati
11      u svakom redu, i se postavlja na n-1 i smanjuje u svakoj
12      iteraciji petlje. */
13     for (i = n - 1; i--;) {
14         /* Prvo se ispisuju beline koje prethode karakterima *. */
15         for (j = 0; j < n - i + 1; j++)
16             printf(" ");
17         /* Posle belina se ispisuje potreban broj karaktera *. */
18         for (j = 0; j < 2 * i + 1; j++)
19             printf("*");
20         printf("\n");
21
22         /* Posebna paznja mora da se obrati na cinjenicu da su brojac
23         tipa unsigned int. Problem nastaje kada je i==0 i pokusa se
24         oduzimanje (i--). Posto su brojevi unsigned, nova vrednost
25         neće biti -1, već pozitivan ceo broj. Imajući to na umu,
26         uslov i>=0 ne može da se stavi u uslov za for petlju. Mnogo
27         sigurnije je brojac deklarirati da budu tipa int i izbeći
28         ovakvu vrstu problema. */
29         if (i == 0)
30             break;
31     }
32
33     return 0;
34 }
```

### Rešenje (c)

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



```

5  unsigned int n;
   int i, j;

7

   printf("Unesite broj n: ");
9   scanf("%u", &n);

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

21
   /* Sada se ispisuje donji trougao. Kako je prvi red donjeg
23     trougla vec ispisan (poslednji red gornjeg trougla), potrebno
     je naciniti jednu iteraciju manje. */

25
   /* Brojac i odredjuje koliko redova se ispisuje. Radi lakseg
27     izracunavanja koliko zvezdica i praznina je potrebno ispisati
     u svakom redu, i se postavlja na n-1 i smanjuje u svakoj
29     iteraciji petlje. */
   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++)
         printf("*");
37     printf("\n");
   }

39
   return 0;
41 }

```

*Rešenje (d)*

```

1  #include <stdio.h>

3  int main()
   {
5     unsigned int n, i, j;

7     printf("Unesite broj n: ");
     scanf("%u", &n);

9

11    /* Brojac i odredjuje koji red slike se trenutno ispisuje. */
     for (i = 0; i < n; i++) {

```

## 2 Kontrola toka

```
13      /* Prvo se ispisuju beline koje prethode karakterima *. */
      for (j = 0; j < n - i - 1; j++)
          printf(" ");
15      /* Posle belina se ispisuje sam trougao. Ako je brojac na
         ivici onda se ispisuje karakter *, a inace praznina.
17      Takodje, proverava se da li se ispisuje poslednji red (i==n)
         i u njemu se ispisuje svaki drugi put *, a inace praznina.
19      Kako se ispisuje svaki drugi put vrsi se provera j%2 == 0. */
      for (j = 0; j < 2 * i + 1; j++)
21          if (j == 0 || j == 2 * i || (i == n - 1 && j % 2 == 0))
              printf("*");
23          else
              printf(" ");
25      printf("\n");
27  }
29  return 0;
}
```

### Rešenje (c)

```
1  #include <stdio.h>
3  int main()
4  {
5      unsigned int n;
6      int i, j;
7
8      printf("Unesite broj n: ");
9      scanf("%u", &n);
10
11     /* Brojac i odredjuje koji red slike se trenutno ispisuje. */
     for (i = 0; i < n; i++) {
13         /* Prvo se ispisuju beline koje prethode karakterima *. */
         for (j = 0; j < n - i - 1; j++)
15             printf(" ");
16         /* Posle belina se ispisuje sam trougao. Ako je brojac na
            ivici onda se ispisuje karakter *, a inace praznina.
17         Takodje, proverava se da li se ispisuje poslednji red (i==n)
            i u njemu se ispisuje svaki drugi put *, a inace praznina.
19         Kako se ispisuje svaki drugi put vrsi se provera j%2 == 0. */
         for (j = 0; j < 2 * i + 1; j++)
21             if (j == 0 || j == 2 * i || (i == n - 1 && j % 2 == 0))
                printf("*");
23             else
                printf(" ");
25         printf("\n");
27     }
29
30     /* Sada se ispisuje donji trougao. Kako je prvi red donjeg
       trougla vec ispisan (poslednji red gornjeg trougla), potrebno
```

```

31     je naciniti jednu iteraciju manje. */
33     for (i = n - 2; i >= 0; i--) {
34         /* Prvo se ispisuju beline koje prethode karakterima *. */
35         for (j = 0; j < n - i - 1; j++)
36             printf(" ");
37         /* Posle belina se ispisuje potreban broj karaktera *. */
38         for (j = 0; j < 2 * i + 1; j++)
39             if (j == 0 || j == 2 * i)
40                 printf("*");
41             else
42                 printf(" ");
43         printf("\n");
44     }
45     return 0;
46 }

```

### Rešenje 2.3.56

```

1  #include <stdio.h>
2
3  int main()
4  {
5      unsigned int n, i, j;
6
7      printf("Unesite broj n: ");
8      scanf("%u", &n);
9
10     /* Strelica se moze posmatrati kao spojena dva pravougla trougla
11        kojima se ispisuje hipotenuzu i jedna, donja kateta. */
12
13     /* Brojac i odredjuje koji red slike se trenutno ispisuje. */
14     for (i = 0; i < n; i++) {
15         for (j = 0; j <= i; j++)
16             /* Proverava se da li se ispisuje karakter na hipotenuzi (j
17                == i-1) ili da se ispisuje poslednji red (i == n-1). */
18             if (j == i || i == n - 1)
19                 printf("*");
20             else
21                 printf(" ");
22         printf("\n");
23     }
24
25     /* Potrebno je iscrtati i donji deo slike, odnosno donji trougao.
26        Brojac i odredjuje koji red donjeg trougla se trenutno iscrtava.
27        * Kako je prvi red donjeg trougla vec iscrtan (to je poslednji
28        red gornjeg trougla), brojac se postavlja na 1. */
29     for (i = 1; i < n; i++) {
30         for (j = 0; j < n - i; j++)
31             /* Provera da li se ispisuje hipotenuzu. */

```

## 2 Kontrola toka

---

```
        if (j == n - i - 1)
33         printf("*");
        else
35         printf(" ");
        printf("\n");
37     }
39     return 0;
}
```

### Rešenje 2.3.57

```
1  #include <stdio.h>
3  int main()
{
5     unsigned int n;
    int i, j, k;
7
    printf("Unesite broj n: ");
9     scanf("%u", &n);

11     /* Brojac j odredjuje koliko ukupno karaktera (praznina i
        karaktera *) u svakom redu se ispisuje. U svakom drugom redu
13     ovaj broj se povecava za 2. Na pocetku je 1 (jer se ispisuje
        samo jedna zvezda). */
15     j = 1;

17     /* Brojac i odredjuje koji red slike se trenutno ispisuje. */
    for (i = 1; i <= n; i++) {
19         /* U svakom drugom redu broj ispisanih karaktera se uvecava za
            2. */
21         if (i % 2 == 0)
            j += 2;
23         for (k = 0; k < j; k++)
            /* U svakom drugom redu se naizmenicno ispisuje * ili
25             praznina. */
            if (i % 2 == 0) {
27                 if (k % 2 == 0)
                    printf("*");
29                 else
                    printf(" ");
31             } else
                printf("*");
33         printf("\n");
35     }

37     return 0;
}
```

## Rešenje 2.3.58

```
2  #include <stdio.h>
3
4  int main()
5  {
6      unsigned int n, m;
7      int i, j, k;
8
9      printf("Unesite brojeve n i m: ");
10     scanf("%u%u", &n, &m);
11
12     for (i = 1; i <= m; i++) {
13         /* Za svaki kvadrat se racuna duzina bez poslednje ivice.
14          * Kvadrat je sastavljen od (m-1) zvezdice i (m-1) praznine
15          * (praznine se nalaze izmedju zvezdica). Znacni ukupna duzina
16          * je 2*(m-1) karakter, a kako ima n kvadrata, duzina je
17          * n*2*(m-1). */
18         for (j = 0; j <= n * 2 * (m - 1); j++)
19             /* Provera da li se ispisuje prvi ili poslednji red. */
20             if (i == 1 || i == m)
21                 /* Naizmenicno se ispisuje * i praznina. */
22                 if (j % 2 == 0)
23                     printf("*");
24                 else
25                     printf(" ");
26             else
27                 /* Na kraju svakog kvadrata (nakon svake (m-1) zvezdice i
28                  * (m-1) praznine se ispisuje ivica kvadrata. */
29                 if (j % (2 * (m - 1)) == 0)
30                     printf("*");
31                 else
32                     printf(" ");
33         printf("\n");
34     }
35
36     return 0;
37 }
```

## Rešenje 2.3.59

```
1  #include <stdio.h>
2
3  int main()
4  {
5      unsigned int n;
6      int i, j;
7
8      printf("Unesite broj n: ");
```

## 2 Kontrola toka

```
9  scanf("%u", &n);

11 /* Potrebno je spojiti sve slike u jednu, sliku gornjeg dela
    romba i sliku donjeg dela romba. */

13
15 /* Brojac i odredjuje koji red slike se trenutno ispisuje. */
16 for (i = 0; i < n; i++) {
17     /* Prvo se ispisuju * koje prethode karakterima -. */
18     for (j = 0; j < n - i; j++)
19         printf("*");
20     /* Posle * se ispisuje potreban karakter -. */
21     for (j = 0; j < 2 * i; j++)
22         printf("-");
23     /* Potom se ispisuju * koje su nakon karaktera -. */
24     for (j = 0; j < n - i; j++)
25         printf("*");
26     printf("\n");
27 }

29 /* Sada se ispisuje donji trougao. Kako je prvi red donjeg
    trougla vec ispisan (poslednji red gornjeg trougla), potrebno
    je naciniti jednu iteraciju manje. */

31
33 /* Brojac i odredjuje koliko redova se ispisuje. Radi lakseg
    izracunavanja koliko zvezdica i praznina je potrebno ispisati
    u svakom redu, i se postavlja na n-1 i smanjuje u svakoj
    iteraciji petlje. */
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     /* Posle * se ispisuje potreban karakter -. */
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 }

47
49 return 0;
}
```

### Rešenje 2.3.60

```
1  #include <stdio.h>

3  int main()
4  {
5      unsigned int n, i, j;
```

```

7  printf("Unesite broj n: ");
   scanf("%u", &n);

9

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

13 /* Prvo se iscrtava krov, odnosno trougao. */
   for (i = 0; i < n - 1; i++) {
15     /* Prvo se ispisuju beline koje prethode karakterima *. */
       for (j = 0; j < n - i - 1; j++)
17         printf(" ");
       for (j = 0; j < 2 * i + 1; j++)
19         if (j == 0 || j == 2 * i)
           printf("*");
21         else
           printf(" ");
23     printf("\n");
   }

25

27 /* Potom se iscrtava kvadrat. Da bi iscrtavanje bilo lakse
    istovremeno se ispisuju dva karaktera. */
   for (i = 0; i < n; i++) {
29     for (j = 0; j < n; j++)
       /* Provera da li je ivica. */
31       if (j == 0 || j == n - 1 || i == 0 || i == n - 1)
           printf("* ");
33       else
           printf(" ");
35     printf("\n");
   }

37

39 return 0;
}

```

### Rešenje 2.3.61

```

1  #include <stdio.h>

3  int main()
   {
5     unsigned int n, i, j;

7     printf("Unesite broj n: ");
       scanf("%u", &n);

9

11     for (i = 1; i <= (n + 1) / 2; i++) {
       for (j = i; j <= n + 1 - i; j++)
13         printf("%d ", j);

15     return 0;
   }

```

```
}  
}
```

### Rešenje 2.3.62

```
1  #include <stdio.h>  
  
3  int main()  
4  {  
5      unsigned int n, i, j;  
  
7      printf("Unesite broj n: ");  
8      scanf("%u", &n);  
9  
10     for (i = 1; i <= n; i++) {  
11         for (j = 1; j <= n; j++)  
12             if (j % i == 1 || i == 1)  
13                 printf("%d ", j);  
14  
15         printf("\n");  
16     }  
17  
18     return 0;  
19 }
```