

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

1	Uvodni zadaci	1
1.1	Naredba izraza	1
2	Kontrola toka	11
2.1	Naredbe grananja	11
2.2	Petlje	22
2.3	Funkcije	43
2.4	Rešenja	52

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!
```

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
```

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

1 Uvodni zadaci

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
```

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
```

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 2

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

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.
```


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
```

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
```

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
```

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
```

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.*

1 Uvodni zadaci

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
```

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
```

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
```

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.*

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
```

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.*

1 Uvodni zadaci

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
```

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

Primer 1

```
|| INTERAKCIJA SA PROGRAMOM:
|| 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
```

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

Primer 1

```
|| INTERAKCIJA SA PROGRAMOM:
|| 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
```

Zadatak 1.1.19 Napisati program koji učitava temperaturu izraženu u farenhajtima, konvertuje tu vrednost u celzijuse i ispisuje je zaokruženu na dve decimalne. 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
```

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

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

Pri ispisu vrednost zaokružiti na 4 decimalne.

Primer 1

```

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

```

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

Primer 1

```

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

```

Primer 2

```

|| INTERAKCIJA SA PROGRAMOM:
|| Unesite duzinu poluprecnika kruga: 14.932
|| Obim: 93.82, povrsina: 700.46

```

Zadatak 1.1.23 Napisati program koji za unetu realnu vrednost dužine stranice jednakostraničnog trougla ispisuje njegov obim i površinu zaokružene na dve decimale. NAPOMENA: *Pretpostaviti da je unos ispravan.*

Primer 1

```

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

```

Primer 2

```

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

```

Zadatak 1.1.24 Napisati program koji za unete realne vrednosti dužina stranica trougla ispisuje njegov obim i površinu zaokružene na dve decimale.

1 Uvodni zadaci

NAPOMENA: *Pretpostaviti da je unos ispravan.*

Primer 1

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

Primer 2

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

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 decimale.

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 decimale.

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
```

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 krećenje 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 krećenja. Sve realne vrednosti ispisati zaokružene na dve decimale.

Primer 1

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

Primer 2

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

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 . 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
```

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
```

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

1 Uvodni zadaci

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
```

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
```

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
```


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.
```

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.

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
```

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.
```

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
```

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
```

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: 777
Najveca cifra je: 7
```

Primer 4

```
INTERAKCIJA SA PROGRAMOM:
Unesite broj: -202
Najveca cifra je: 2
```

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.
```

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.

2 Kontrola toka

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
```

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
```

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
```

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
```

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.
```

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.
```

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

Primer 1

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

Primer 2

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

Primer 3

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

Primer 4

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

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 cetvorocifren!
```

Primer 3

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

Primer 4

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

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.

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 $[a1, b1]$ i $[a2, b2]$, 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.
```

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.
```

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 neopadajuće.
```

Primer 2

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

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.
```

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
```

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.
```

Zadatak 2.1.25 Data je funkcija $f(x) = 2 \cdot \cos(x) - x^3$. Napisati program koji za učitanu vrednost realne promenljive x i vrednost 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
```

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: cetvrtak
```

Primer 2

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

Primer 3

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

Primer 4

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

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.
```

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
```

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
```

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.

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!
```

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.
```

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 pravougaonika čije su stranice paralelne koordinatnim osama i u slučaju da jesu, ispisuje vrednost obima datog pravougaonika. Možemo podrazumevati da će korisnik koordinate tačaka unositi 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.

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:

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

2.2 Petlje

Zadatak 2.2.1 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.
```

Zadatak 2.2.2 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
```

Zadatak 2.2.3 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!
```

Zadatak 2.2.4 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!
```

Zadatak 2.2.5 Napisati program koji učitava ceo pozitivan broj i izračunava njegov faktoriyel. U slučaju neispravnog unosa ispisati odgovarajuću poruku.

Primer 1

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

Primer 2

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

Primer 3

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

Zadatak 2.2.6 Sa standradnog 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
```

2 Kontrola toka

Zadatak 2.2.7 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
```

Zadatak 2.2.8 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.
```

Zadatak 2.2.9 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
```

Zadatak 2.2.10 Napisati program koji učitava ceo broj i ispisuje njegove cifre u obrnutom poretку.

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
```

Zadatak 2.2.11 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.

```

Zadatak 2.2.12 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.

```

Zadatak 2.2.13 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.

```

Zadatak 2.2.14 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.
```

Zadatak 2.2.15 Nikola želi da obradi bazu i da joj kupi jedan poklon u radnji. On na raspolaganju ima m novaca. U radnji se nalazi n artikala i zanima ga koliko ima artikala u radnji čija cena je manja ili jednaka m . Napisati program koji pomaže Nikoli da brzo 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.
```

Zadatak 2.2.16 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.
```

Zadatak 2.2.17 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!
```

Zadatak 2.2.18 Program učitava cele brojeve sve do unosa broja nula 0. Napisati program koji izračunava i ispisuje aritmetičku sredinu unetih brojeva na četiri decimalne.

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.
```

Zadatak 2.2.19 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.
```

Zadatak 2.2.20 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.
```

Zadatak 2.2.21 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 može biti negativna.
```

Primer 3

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

Zadatak 2.2.22 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.
```

Zadatak 2.2.23 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.

```

Zadatak 2.2.24 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 1

```

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

```

Zadatak 2.2.25 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

```

Zadatak 2.2.26 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

```

Zadatak 2.2.27 Napisati program koji uneti prirodan broj transformiše tako što svaku parnu cifru u zapisu broja uveća za jedan. Ispisati novodobijeni

2 Kontrola toka

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
```

Zadatak 2.2.28 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
```

* **Zadatak 2.2.29** 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.
```

* **Zadatak 2.2.30** 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!
```

Zadatak 2.2.31 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
```

* **Zadatak 2.2.32** 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
```

* **Zadatak 2.2.33** 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
```

Zadatak 2.2.34 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!
```

Zadatak 2.2.35 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
```

Zadatak 2.2.36 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
```

Zadatak 2.2.37 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 moze se napisati rec Zima.

```

Primer 2

```

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

```

Zadatak 2.2.38 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

```

Zadatak 2.2.39 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

```

Zadatak 2.2.40 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
```

Zadatak 2.2.41 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
```

* **Zadatak 2.2.42** 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 tačnost eps: 0.001
|| S=7.388713
```

Primer 2

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

* **Zadatak 2.2.43** 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 tačnost eps: 0.000001
|| S=-1.131112
```

Primer 2

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite x: 3.14
|| Unesite tačnost eps: 0.01
|| S=-1.010296
```

Zadatak 2.2.44 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
```

* **Zadatak 2.2.45** 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.
```

* **Zadatak 2.2.46** 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=1.163199
```

Primer 2

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite x i n: 14.32 11
|| S=-31022.636719
```

Primer 3

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

* **Zadatak 2.2.47** Program učitava ceo pozitivan broj n veći od 0. Napisati program koji računa proizvod

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

2 Kontrola toka

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
```

* **Zadatak 2.2.48** 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.
```

Zadatak 2.2.49 Program učitava realne brojeve x i a i ceo pozitivan broj n veći od 0. Napisati program koji izračunava:

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

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.

```

Zadatak 2.2.50 Program učitava ceo pozitivan broj n . 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)

```

Zadatak 2.2.51 Program učitava prirodan broj n . Napisati program koji za uneto n zvezdicama iscrtava odgovarajuću sliku.

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
****
** *
* * *
****
```

* **Zadatak 2.2.52** Napisati program koji za uneti prirodan broj n zvezdicama iscrtava slovo X dimenzije n .

Primer 1

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

Primer 2

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

* **Zadatak 2.2.53** Napisati program koji za uneti prirodan broj n korišćenjem znaka $+$ iscrtava veliko $+$ dimenzije n .

Primer 1

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

Primer 2

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

Zadatak 2.2.54 Napisati program koji učitava prirodan broj n , a potom iscrtava odgovarajuću sliku.

- 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
****
***
**
*
```

- 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 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
*
**
***
****
***
**
*
```

- d) Slika sadrži rub jednakokrakog pravougloug 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: +
+
++
+ +
+ +
++++
```

Zadatak 2.2.55 Napisati program koji učitava ceo broj n , a potom iscrtava odgovarajuću sliku.

- a) Slika sadrži jednakokranični 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 trougao koji se dobija spajanjem dva jednakokranični trougla stranice n koji su sastavljeni od zvezdica.

Primer 1

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

- c) Slika sadrži rub jednakokraničnog trougla čija stranica je dužine n .

Primer 1

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

- d) 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

  *
 * *
* * *
 * *
  *
```

* **Zadatak 2.2.56** Napisati program koji za uneti prirodan broj n iscrtava strelice dimenzije n .

Primer 1

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

 *
 *
***
 *
 *
```

Primer 2

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

 *
 *
 *
 *
*****
 *
 *
 *
 *
```

* **Zadatak 2.2.57** 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 .

Primer 1

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

 *
 * *
***
 * * *
*****
 * * * *
*****
```

* **Zadatak 2.2.58** 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.

Primer 1

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

Primer 2

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

* **Zadatak 2.2.59** Program učitava prirodan broj n . Napisati program koji štampa romb sastavljen od minusa u pravougaoniku sastavljenom od zvezdica.

Primer 1

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

Primer 2

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

Zadatak 2.2.60 Napisati program koji učitava ceo broj n ($n \geq 2$) i koji iscrtava sliku kuće sa krovom: kuća je kocka stranice n , a krov jednakokranični trougao stranice n .

Primer 1

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

Primer 1

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

* **Zadatak 2.2.61** 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.

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: 5
|| 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
```

* **Zadatak 2.2.62** 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
```

2.3 Funkcije

TODO U nekim zadacima pise samo Napisati program koji testira rad ove funkcije, dok u nekim zadacima je detaljnije opisan sam program, npr "Napisati program koji sa standardnog ulaza učitava tri cela broja i ispisuje rezultat poziva funkcije."Nekako bi trebalo to ujednaciti. Meni je lepsi ovaj drugi stil, jer je precizniji.

TODO Smisliti odgovarajući redosled za ove zadatke

Zadatak 2.3.1 Ovaj zadatak bih razbila na dva zadatka jer u resenju ima dve poenteod kojih je prvi resen, a drugi neresen. Poentu o vidljivosti promenljivih bih ostavila za kasnije, za neki drugi zadatak, jer mi je ovde to mnogo rano — prvih par zadataka ne bi trebalo dodatno time opterecivati. Napisati funkcije `int kvadrat(int x)` i `int kub(int x)` koje računaju, redom, kvadrat i kub datog broja. Napisati program koji testira rad ovih funkcija.

Zadatak 2.3.2 Napisati funkciju `float stepen(float x, int n)` koja računa vrednost n -tog stepena realnog broja x . Napisati program koji testira rad ove funkcije.

Zadatak 2.3.3 Napisati funkciju `int euklid(int x, int y)` koja za dva data cela broja određuje najveći zajednički delilac primenom Euklidovog algoritma. Napisati program koji testira rad ove funkcije.

Zadatak 2.3.4 Napisati funkciju `float zbir_reciprocnih(int n)` koja za dato n vraća zbir recipročnih vrednosti brojeva od 1 do n . Napisati program koji testira rad ove funkcije. Rezultat zaokružiti na dve decimale.

Zadatak 2.3.5 Napisati funkciju `float aritmeticka_sredina(int n)` koja računa aritmetičku sredinu cifara datog broja. Napisati i program koji testira rad ove funkcije. Rezultat ispisivati na tri decimale.

Primer 1

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

Primer 2

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

Primer 3

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

Zadatak 2.3.6 Napisati funkciju `void ispis(float x, float y, unsigned n)` koja za dva realna broja x i y i jedan neoznačeni ceo broj n ispisuje vrednosti sinusne funkcije u n ravnomerno raspoređenih tačaka intervala $[x, y]$. Napisati program koji testira rad ove funkcije.

Zadatak 2.3.7 Napisati funkciju `int broj_ncifara(int x)` koja broji neparne cifre u zapisu datog celog broja. Testirati rad ove funkcije u programu koji učitava cele brojeve dok se ne unese nula i ispisuje broj neparnih cifara svakog unetog broja.

Zadatak 2.3.8 Napisati funkciju `int min(int x, int y, int z)` koja izračunava minimum tri broja. Napisati program koji sa standardnog ulaza učitava tri cela broja i ispisuje rezultat poziva funkcije.

Primer 1

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

Primer 2

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

Zadatak 2.3.9 Napisati funkciju `unsigned int apsolutna_vrednost(int x)` koja izračunava apsolutnu vrednost broja x . Napisati program koji sa standardnog ulaza učitava jedan ceo broj i ispisuje rezultat poziva funkcije.

Primer 1

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesite broj: -34  
|| Apsolutna vrednost: 34
```

Primer 2

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesite broj: 5  
|| Apsolutna vrednost: 5
```

Zadatak 2.3.10 Napisati funkciju `float razlomljeni_deo(float x)` koja izračunava razlomljeni deo broja x . Napisati program koji sa standardnog ulaza učitava jedan realan broj i ispisuje rezultat poziva funkcije.

Primer 1

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

Primer 2

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

Zadatak 2.3.11 Napisati funkciju `void romb(int n)` koja iscrtava romb čija je stranica dužine n . Napisati program koji učitava ceo pozitivan broj i ispisuje rezultat poziva funkcije. U slučaju pogrešnog unosa, ispisati poruku o grešci.

Primer 1

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

Primer 2

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

Primer 3

```
INTERAKCIJA SA PROGRAMOM:
Unesite broj n: -5
Greska: pogresna dimenzija!
```

Zadatak 2.3.12 Napisati funkciju `void grafikon_h(int a, int b, int c, int d)` koja iscrtava horizontalni prikaz zadatih vrednosti. Napisati program koji učitava četiri pozitivna cela broja i prikazuje rezultat poziva funkcije. U slučaju pogrešnog unosa, ispisati poruku o grešci.

Primer 1

```
INTERAKCIJA SA PROGRAMOM:
Unesite vrednosti: 4 1 7 5
****
*
*****
*****
```

Primer 2

```
INTERAKCIJA SA PROGRAMOM:
Unesite vrednosti: 8 -2 5 4
Greska: pogresan unos!
```

Primer 3

```
INTERAKCIJA SA PROGRAMOM:
Unesite vrednosti: 5 2 2 10
*****
**
**
*****
```

Zadatak 2.3.13 Napisati funkciju `void grafikon_v(int a, int b, int c, int d)` koja iscrtava vertikalni prikaz zadatih vrednosti. Napisati program koji učitava četiri pozitivna cela broja i ispisuje rezultat poziva funkcije. U slučaju pogrešnog unosa, ispisati poruku o grešci.

Primer 1

```
INTERAKCIJA SA PROGRAMOM:
Unesite vrednosti: 4 1 7 5
*
*
**
* **
* **
* **
****
```

Primer 2

```
INTERAKCIJA SA PROGRAMOM:
Unesite vrednosti: 8 -2 5 4
Greska: pogresan unos!
```

Primer 3

```
INTERAKCIJA SA PROGRAMOM:
Unesite vrednosti: 5 2 2 4
*
* *
* *
****
****
```

Zadatak 2.3.14 Napisati funkciju *int prestupna(int godina)* koja za zadatu godinu proverava da li je prestupna. Funkcija treba da vrati 1 ako je godina prestupna ili 0 ako nije. Napisati program koji učitava dva cela broja *g1* i *g2* i ispisuje sve godine iz intervala $[g1, g2]$ koje su prestupne.

Primer 1

```
INTERAKCIJA SA PROGRAMOM:
Unesite dve godine: 2001 2010
Prestupne godine su: 2004 2008
```

Primer 2

```
INTERAKCIJA SA PROGRAMOM:
Unesite dve godine: 2005 2015
Prestupne godine su: 2008 2012
```

Primer 3

```
INTERAKCIJA SA PROGRAMOM:
Unesite dve godine: 2010 2001
Greska: pogresan unos!
```

Primer 4

```
INTERAKCIJA SA PROGRAMOM:
Unesite dve godine: 2001 2002
Nema prestupnih godina u ovom intervalu!
```

Zadatak 2.3.15 Napisati funkciju *int broj_dana(int mesec, int godina)* koja za dati mesec i godinu vraća broj dana u datom mesecu. Napisati program koji testira ovu funkciju. U slučaju nekorektnog unosa ispisati odgovarajuću poruku o grešci.

Zadatak 2.3.16 Napisati funkciju *int ispravan(int dan, int mesec, int godina)* koja za dati datum proverava da li je ispravan. Napisati program koji testira ovu funkciju.

Zadatak 2.3.17 Napisati funkciju `void sledeci_dan(int dan, int mesec, int godina)` koja za dati datum određuje datum sledećeg dana. Napisati program koji testira ovu funkciju.

Zadatak 2.3.18 *preimenovala sam funkciju, preimenovati i u resenju* Napisati funkciju `int od_nove_godine(int dan, int mesec, int godina)` koja određuje koliko je dana proteklo od Nove godine do datog datuma. Napisati program koji testira napisanu funkciju.

Zadatak 2.3.19 *Grupisati sve zadatke sa datumima preimenovala sam funkciju, preimenovati i u resenju* Napisati funkciju `int do_kraja_godine(int dan, int mesec, int godina)` koja određuje broj dana od datog datuma do kraja godine. Napisati program koji testira napisanu funkciju.

Zadatak 2.3.20 *preimenovala sam funkciju, preimenovati i u resenju* Napisati funkciju `int broj_dana_između(int dan1, int mesec1, int godina1, int dan2, int mesec2, int godina2)` koja određuje broj dana između dva datuma. Napisati program koji testira napisanu funkciju.

Zadatak 2.3.21 Napisati funkciju `int zbir_delilaca(int n)` koja izračunava zbir delilaca broja n . Napisati program koji sa standardnog ulaza učitava ceo broj k i ispisuje zbir delilaca svakog broja od 1 do k .

Primer 1

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

Primer 2

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite broj k: -2
|| Greska: pogresan unos!
```

Zadatak 2.3.22 Napisati funkciju `int ukloni_stotine(int n)` koja modifikuje zadati broj tako što iz njegovog zapisa uklanja cifru stotina (ako postoji). Napisati program koji za brojeve koji se unose sa standardnog ulaza sve do pojave broja 0 ispisuje rezultat primene funkcije.

Primer 1

```

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

```

Primer 2

```

INTERAKCIJA SA PROGRAMOM:
Unesite broj: -9632
-932
Unesite broj: 246
46
Unesite broj: -52
-52
Unesite broj: 0

```

Zadatak 2.3.23 Napisati funkciju *int rotacija(int n)* koja rotira cifre zadatog broja za jednu poziciju u levo. Napisati program koji za brojeve koji se unose sa standardnog ulaza sve do pojave broja 0 ispisuje rezultat primene funkcije.

Primer 1

```

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

```

Primer 2

```

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

```

Zadatak 2.3.24 Napisati funkciju *int prost (int x)* koja ispituje da li je dati ceo broj prost. Funkcija treba da vrati 1 ako je broj prost i 0 u suprotnom. Testirati rad funkcije u programu koji za uneti ceo broj *n* ispisuje prvih *n* prostih brojeva.

Zadatak 2.3.25 Napisati funkciju *int sadrzi(int x, int c)* koja ispituje da li se cifra *c* nalazi u zapisu celog broja *x*. Napisati program koji testira rad ove funkcije.

Zadatak 2.3.26 Ovo je jedini zadatak gde nije dat potpis funkcije. Ili i ovde dodati potpis, ili razmotriti da se jos negde ukloni potpis funkcije! Napisati program za ispitivanje svojstava cifara datog celog broja.

- (a) Napisati funkciju *sve_parne_cifre* koja ispituje da li se dati ceo broj sastoji isključivo iz parnih cifara. Funkcija treba da vrati 1 ako su sve cifre broja parne i 0 u suprotnom.

- (b) Napisati funkciju `sve_cifre_jednake` koja ispituje da li su sve cifre datog celog broja jednake. Funkcija treba da vrati 1 ako su sve cifre broja jednake i 0 u suprotnom.

Testirati napisane funkcije na unetom celom broju i ispisati odgovarajuće poruke.

Zadatak 2.3.27 Napisati funkciju `int je_stepen(unsigned x, unsigned n)` koja za dva uneta neoznačena broja x i n utvrđuje da li je x neki stepen broja n . Ukoliko jeste, funkcija vraća izložilac stepena, a u suprotnom vraća -1. Napisati program koji testira rad ove funkcije.

Zadatak 2.3.28 Napisati funkciju `double e_na_x(double x, double eps)` koja računa vrednost e^x kao parcijalnu sumu reda $\sum_{n=0}^{\infty} \frac{x^n}{n!}$, pri čemu se sumiranje vrši dok je razlika sabiraka u redu po apsolutnoj vrednosti manja od ε . Napisati program koji testira rad ove funkcije.

Zadatak 2.3.29 Za dati broj može se formirati niz tako da je svaki sledeći član niza dobijen kao suma cifara prethodnog člana niza. Broj je srećan ako se dati niz završava jedinicom. Napisati funkciju `int srećan(int x)` koja vraća 1 ako je broj srećan, a 0 u suprotnom. Napisati program koji za uneti prirodan broj n ispisuje sve srećne brojeve od 1 do n .

Zadatak 2.3.30 Napisati funkciju `int konverzija(int c)` koja prebacuje veliko slovo u ekvivalentno malo i obrnuto. Napisati program koji testira ovu funkciju na karakterima koji se unose sa standardnog ulaza do pojave EOF.

Zadatak 2.3.31 Napisati funkciju `int zapis(int x, int y)` koja proverava da li se brojevi x i y zapisuju pomoću istih cifara. Funkcija treba da vrati vrednost 1 ako je uslov ispunjen, odnosno 0 ako nije. Napisati i program koji učitava dva cela broja i ispisuje rezultat primene funkcije.

Primer 1

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

Primer 2

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

Primer 3

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


Zadatak 2.3.32 Napisati funkciju *int faktorijel(int n)* koja računa faktoriyel broja n . Napisati i program koji učitava dva cela broja x i y iz intervala $[0, 12]$ i ispisuje vrednost zbira $x! + y!$.

Primer 1

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite dva broja: 4 5
|| 144
```

Primer 2

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite dva broja: 18 -5
|| Greska: pogresan unos!
```

Primer 3

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite dva broja: 6 0
|| 721
```

Zadatak 2.3.33 Napisati funkciju *int rastuce(int n)* koja ispituje da li su cifre datog celog broja u rastućem poretku. Funkcija treba da vrati vrednost 1 ako cifre ispunjavaju uslov, odnosno 0 ako ne ispunjavaju uslov. Napisati i program koji učitava ceo broj i ispisuje rezultat primene funkcije.

Primer 1

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite broj: 2689
|| Cifre su u rastucem poretku!
```

Primer 2

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite broj: 559
|| Cifre su u rastucem poretku!
```

Primer 3

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite broj: 628
|| Cifre nisu u rastucem poretku!
```

Zadatak 2.3.34 Broj je Armstrongov ako je jednak sumi nekog stepena svojih cifara.

- Napisati funkciju *int stepen(int x, int n)* koja izračunava n -ti stepen broja x .
- Napisati funkciju *int armstrong(int x)* koja vraća 1 ako je broj Armstrongov, odnosno 0 ako nije.

Napisati program koji za ceo broj koji se unosi sa standardnog ulaza proverava da li je Armstrongov.

Primer 1

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

Primer 2

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite broj: 1634
|| Broj je Armstrongov!
```

Primer 3

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite broj: 118
|| Broj nije Armstrongov!
```

Zadatak 2.3.35 Napisati funkciju *int par_nepar(int n)* koja ispituje da li su cifre datog celog broja naizmenično parne i neparne. Funkcija treba da vrati vrednost 1 ako cifre ispunjavaju uslov, odnosno 0 ako ne ispunjavaju uslov. Napisati i program koji učitava ceo broj i testira rad funkcije.

Primer 1

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

Primer 2

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

Primer 3

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

Zadatak 2.3.36 Napisati funkciju *int prebrojavanje(float x)* koja prebrojava koliko puta se broj x pojavljuje u nizu brojeva koji se unose sa standardnog ulaza sve do pojave nule. Napisati program koji učitava vrednost broja x i testira rad napisane funkcije.

Primer 1

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

Primer 2

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

Zadatak 2.3.37 Fibonačijev niz je niz za koji važi: $F_0 = 1$, $F_1 = 1$, $F_{n+2} = F_{n+1} + F_n$ za $n \geq 0$. Napisati funkciju *long int fibonacci(int n)* koja računa n -ti element Fibonačijevog niza. Napisati i program koji učitava ceo broj n ($0 \leq n \leq 50$) i ispisuje traženi Fibonačijev broj.

Primer 1

```

INTERAKCIJA SA PROGRAMOM:
Unesite broj n: 7
21

```

Primer 2

```

INTERAKCIJA SA PROGRAMOM:
Unesite broj n: 65
Greska: nedozvoljena vrednost!

```

Zadatak 2.3.38 Napisati funkciju *char sifra(char c, int k)* koja za dati karakter *c* određuje šifru na sledeći način: ukoliko je *c* slovo, šifra je karakter koji se nalazi *k* pozicija ispred njega u abecedi. Karakteri koji nisu slova se ne šifruju. Šifrovanje treba da bude kružno, što znači da je, na primer, šifra za karakter *b* i pomeraj 2 karakter *z*. Napisati program koji učitava karakter po karakter do kraja ulaza i ispisuje šifrovani tekst.

Primer 1

```

INTERAKCIJA SA PROGRAMOM:
Unesite broj k: 2
Unesite tekst (CTRL+D za prekid):
c
a
8
8
+
+
Z
X

```

2.4 Rešenja

Rešenje 1.1.1

```

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

12
14  Unutar tela funkcije navode se naredbe. Unutar navedenog programa
   postoji jedna naredba koja predstavlja poziv funkcije printf.
   Funkcija printf služi za ispis teksta na standardni izlaz (obicno
16  ekran). Deklaracija ove funkcije data je u zaglavlju stdio.h

```

2 Kontrola toka

```
18   koje je potrebno ukljuciti direktivom #include na pocetku
    samog programa.

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

34   int main()
    {
36   /* printf: funkcija pomocu koje se vrši ispis */
    /* Specijalni karakter \n : prelazak u novi red */
38   /* Svaka naredba zavrsava se karakterom ; */
    printf("Zdravo svima!\n");
40
    /* Povratna vrednost 0 se obicno koristi da oznaci
42     da je prilikom izvorsavanja programa sve proslo
        u redu. */
44     return 0;
    }
```

Rešenje 1.1.2

```
#include <stdio.h>

2
int main()
4
{
    /*
6     Svaka promenljiva u programu mora biti deklarirana na
        pocetku main funkcije. Deklaracija se sastoji iz naziva
8     promenljive (u ovom slucaju n) ispred kog se navodi tip
        promenljive (u ovom slucaju celobrojni tip, int).
10    */

12    int n;

14    /*
        Vrednost promenljive se ucitava pomocu funkcije scanf koja
16        je, kao i funkcija printf, sastavni deo standardne biblioteke.
        Argumenti funkcije scanf koji se navode u zagradama
18        ( i ) i razdvajaju zarezima, oznacavaju sledece:
```

```

20     "%d" - format za tip podatka koji ce biti ucitan
        (%d za int, svaki tip ima svoj format)
22     &n - adresa promenljive x (o adresama ce biti vise
        reci u narednim zadacima).

24     Ucitavanje se vrsi sa standardnog ulaza (obicno tastatura).
    */
26     printf("Unesite ceo broj: ");
    scanf("%d", &n);

28

30     /*
        Funkcija printf ispisuje tekst "Uneti broj: ", a nakon toga,
        umesto formata %d, ispisuje vrednost promenljive n.
32     */
    printf("Uneti broj: %d\n", n);
34     /* Umesto formata %d, ispisuje vrednost izraza n*n. */
    printf("Kvadrat: %d\n", n*n);
36     /* Umesto formata %d, ispisuje vrednost izraza n*n*n. */
    printf("Kub: %d\n", n*n*n);

38
40     return 0;
}

```

Rešenje 1.1.3

```

2  #include<stdio.h>

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

8

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

12

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

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

20     /*
        Mozemo ispisivati direktno vrednost izraza x-y i bez
        njegovog dodeljivanja posebnoj promenljivoj
22     */
    printf("%d - %d = %d\n",x,y,x-y);
    printf("%d * %d = %d\n",x,y,x*y);
24
26

```

```
28  /*
30     Kada bilo koju aritmetičku operaciju primenimo na dve
    promenljive istog tipa (u ovom slučaju dva celobrojne
32     promenljive), rezultat će biti tog istog tipa. Specijalno,
    za operaciju deljenja: kada operator / primenimo na dva
34     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.
36 */
    printf("%d / %d = %d\n",x,y,x/y);
38
40 /*
    Operator % izračunava ostatak pri celobrojnem deljenju
    dve celobrojne promenljive. Na primer, 7%2 ima vrednost 1
42     (jer je 7=3*2+1).
    Da bismo odstampali karakter %, u naredbi printf pisemo %%
44 */
    printf("%d %% %d = %d\n",x,y,x%y);
46
48     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>
3
4  int main()
5  {
6      /*
7         S obzirom da su sve promenljive pozitivni celi brojevi,
            koristimo tip unsigned int (skraćeno unsigned)
9      */
10     unsigned cena;
11     unsigned kolicina;
12     unsigned iznos;
```

```

13     unsigned kusur;

15     /*
16        Ucitavamo potrebne podatke. Unutar jednog scanf-a mozemo
17        ucitati vise podataka odjednom. Za svaki treba navesti
18        odgovarajuci format za tip podataka koji se unosi
19        (%u za unsigned).
20     */
21     printf("Unesite cenu, kolicinu i iznos: ");
22     scanf("%u%u%u", &cena, &kolicina, &iznos);

23     /* Izracunavamo kusur: */
24     kusur=iznos - kolicina*cena;

25     /* I ispisujemo trazenu vrednost: */
26     printf("Kusur je %u dinara.\n", kusur);

27     return 0;
28 }

```

Rešenje 1.1.7

```

#include <stdio.h>

2 int main()
3 {
4
5     unsigned poletanje, poletanje_sat, poletanje_minut;
6     unsigned sletanje, sletanje_sat, sletanje_minut;
7     unsigned duzina, duzina_sat, duzina_minut;
8
9     printf("Unesite vreme poletanja: ");
10    scanf("%u%u", &poletanje_sat, &poletanje_minut);
11
12
13    printf("Unesite vreme sletanja: ");
14    scanf("%u%u", &sletanje_sat, &sletanje_minut);
15
16    /* Pretvoricemo i vreme poletanja i vreme sletanja u sekunde */
17    poletanje=poletanje_sat*3600+poletanje_minut*60;
18    sletanje=sletanje_sat*3600 + sletanje_minut*60;
19
20    /* I izracunati razliku u sekundama */
21    duzina=sletanje-poletanje;
22
23    /* Izdvajamo broj sati i broj minuta. */
24    duzina_sat=duzina/3600;
25    duzina_minut=(duzina%3600)/60;
26
27
28    /* I ispisujemo rezultat */

```

2 Kontrola toka

```
30 printf("Duzina trajanja leta je %u h i %u min\n", duzina_sat,
    duzina_minut);
32
33 return 0;
34 }
```

Rešenje 1.1.8

```
#include<stdio.h>
2 int main()
{
4     int x,y;
    int p;
6
    printf("Unesi dve celobrojne vrednosti:");
8     scanf("%d%d",&x,&y);
10
    printf("pre zamene: x=%d, y=%d\n",x,y);
12
    /* Pomocna promenljiva p je potrebna da sacuva vrednost
       promenljive x pre nego sto se ona izmeni i dobije
       vrednost promenljive y. */
14
    p=x;
16    x=y;
    y=p;
18    printf("posle zamene: x=%d, y=%d\n",x,y);
    return 0;
20 }
```

Rešenje 1.1.10

```
1
#include <stdio.h>
3 int main()
{
5     /* S obzirom da broj treba da bude pozitivan,
       koristimo tip unsigned.
7
       */
    unsigned x;
9
10    /*
11     Promenljive koje cuvaju cifre treba da budu
       najmanjeg celobrojnog tipa jer nece sadrzati
13     druge vrednosti osim jednocifrenih celih
       brojeva. Zbog toga za njih biramo tip char.
15
16     */
17     char cifra_jedinice;
```



```
19 char cifra_desetice;
20 char cifra_stotine;

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

23
24 /*
25  Na primer, neka je uneti broj 374. Potrebno je da koriscenjem
26  racunskih operacija za rad sa celim brojevima pristupimo
27  njegovoj cifri jedinice, cifri desetice i cifri stotine.

28
29  Primetimo najpre sledece:
30  374/10 = 37
31  374%10 = 4
32  Dakle, operacijama celobrojnog deljenja i ostatka pri deljenju
33  mozemo iz svakog broja izdvojiti njegovu poslednju cifru (u
34  ovom slucaju 4) i broj sastavljen od svih cifara osim poslednje
35  (u ovom slucaju 37).

36
37  Cifri jedinice sada lako pristupamo koriscenjem ostatka pri
38  deljenju sa 10. Ona iznosi upravo 4.

39
40  Pri trazanju cifre desetice mozemo ponovo primeniti princip
41  izdvajanja poslednje cifre kao ostatka pri deljenju sa 10.
42  Razlika je sto ne mozemo deseticu izdvojiti ako primenimo %10
43  na 374 (time dobijamo 4), vec %10 primenjujemo na 37, pri cemu
44  37 dobijamo kao ceo deo pri deljenju broja 374 brojem 10.
45  Dakle, cifru desetice dobijamo kao (374/10)%10.

46
47  S obzirom da znamo da je u pitanju trocifreni broj, cifru
48  stotine mozemo izdvojiti celobrojnim deljenjem sa 100: 374/100
49  iznosi upravo 3.

50
51  */
52 cifra_jedinice = x%10;
53 cifra_desetice = (x/10)%10;
54 cifra_stotine = x/100;

55
56 /*
57  Ako zelimo da odstampamo numericku vrednost promenljive
58  tipa char, koristimo format %d. Ako zelimo da odstampamo
59  karakter ciji je ASCII kod jednak vrenosti te promenljive,
60  koristimo %c (na primer, ako bismo promenljivu cija je
61  vrednost 65 stampali pomocu formata %d, ispis bi bio 65, ali
62  ako bismo je stampali pomocu formata %c, ispis bi bio A). U
63  ovom slucaju nam je neophodna numericka vrednost.

64  */
65 printf("jedinica %d, desetica %d, stotina %d\n", cifra_jedinice,
66        cifra_desetice,
67        cifra_stotine);

68
69 /*
```

2 Kontrola toka

```
2. nacin, bez uvođenja dodatnih promenljivih cifra_jedinice,  
71   cifra_desetice i cifra_stotine:  
  
73   printf("Cifre unetog broja su %d,%d,%d\n", x%10, (x/10)%10, x  
      /100);  
      */  
75   return 0;  
}
```

Rešenje 1.1.11

```
1  #include <stdio.h>  
3  
4  int main()  
5  {  
6      unsigned x;  
7      printf("Unesi cenu:");  
8      scanf("%u", &x);  
9  
10     /*  
11     Na primer, neka je uneta cena 8347 dinara.  
12     Vrednost x/5000 predstavlja broj novcanica  
13     od 5000 dinara pomocu kojih mozemo sakupiti  
14     celokupnu sumu. 8347 celobrojno deljeno sa  
15     5000 (operacija / nad celim brojevima) iznosi 1.  
16     */  
17     printf("%u=%u*5000+ ", x,x/5000);  
18     /*  
19     Potrebna nam je 1 novcanica od  
20     5000 dinara, a koliko nam je potrebno ostalih  
21     novcanica? Za to moramo pristupiti preostaloj  
22     sumi. Jedan nacin je da nadjemo ostatak pri deljenju  
23     unete vrednosti x (u primeru 8347) sa 5000 (operacija %).  
24     On iznosi 3347. Ovu vrednost dodeljujemo promenljivoj x.  
25     */  
26     x=x%5000;  
27  
28     /*  
29     Nastavljamo postupak trazenjem broja novcanica  
30     od 2000 dinara i redom za ostale monete.  
31     */  
32     printf("%u*2000 +", x/2000);  
33     x=x%2000;  
34     printf("%u*1000 +", x/1000);  
35     x=x%1000;  
36     printf("%u*500 +", x/500);  
37     x=x%500;  
38     printf("%u*200 +", x/200);  
39     x=x%200;  
40     printf("%u*100 +", x/100);
```

```

41     x=x%100;
    printf("%u*50 +", x/50);
43     x=x%50;
    printf("%u*20 +", x/20);
45     x=x%20;
    printf("%u*10 +", x/10);
47     x=x%10;
    printf("%u*1\n", x);
49     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;
    char cifra_stotine;

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

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

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

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

19    return 0;
21 }

```

Rešenje 1.1.13

```

1  #include <stdio.h>
3  int main(){
5     unsigned n, broj_obrnuto, broj_zamena;
    char j, d, s, h;
7     int proizvod_cifara, razlika_cifara, suma_kvadrata;

```

```
9  /* Ucitavamo vrednost sa ulaza */
   printf("Unesite cetvorocifreni broj: ");
11  scanf("%u", &n);

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

19  /* Izracunavamo proizvod cifara */
21  proizvod_cifara=j*d*s*h;
   printf("Proizvod cifara: %d\n", proizvod_cifara);

23  /* Izracunavamo razliku sume krajnjih i srednjih cifara */
25  razlika_cifara=(h+j)-(s+d);
   printf("Razlika sume krajnjih i srednjih: %d\n", razlika_cifara);

27  /* Izracunavamo sumu kvadrata cifara */
29  suma_kvadrata=j*j+d*d+s*s+h*h;
   printf("Suma kvadrata cifara: %d\n", suma_kvadrata);

31  /* Odredjujemo broj zapisan istim ciframa ali u obrnutom redosledu
      */
33  broj_obrnuto= j*1000+d*100+s*10+h;
   printf("Broj u obrnutom poretku: %u\n", broj_obrnuto);

35

37  /* Odredjujemo broj u kojem su cifra jedinica i
      cifra stotina zamenile mesta
39  */
   broj_zamena=h*1000+j*100+d*10+s;
41  printf("Broj sa zamenjenom cifrom jedinica i stotina: %u\n",
      broj_zamena);

43

45  return 0;
}
```

Rešenje 1.1.15

```
#include <stdio.h>

2
int main()
4 {
   unsigned broj, novibroj;
   unsigned levi, desni, m;
8
```

```

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

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

20     levi = broj/1000;
22     desni = broj%1000;

24     /*
        Kada levi deo pomnozimo sa 100 000, dobijamo 1 200 000
26     Kada m pomnozimo sa 1000, dobijamo          67 000
        Dobijene vrednosti saberemo sa desnim delom          345
28
        Konacan rezultat:                                1 267 345
    */

30     novibroj = levi*100000+m*1000+desni;

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

34     return 0;
36 }

```

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 */
      cm = in*2.54;

15     /* "%.2f" - ispis realne promenljive na 4 decimale */
      printf("%.2f in = %.2f cm\n", in, cm);

19 }

```

```
21     return 0;
    }
```

Rešenje 1.1.17 Zadatak se rešava analogno zadatku 1.1.16.

Rešenje 1.1.18 Zadatak se rešava analogno zadatku 1.1.16.

Rešenje 1.1.19 Zadatak se rešava analogno zadatku 1.1.16.

Rešenje 1.1.21

```
2  #include <stdio.h>
4  int main()
5  {
6      float a, b;
7      float obim, povrsina;
8
9      /* Ucitavamo potrebne podatke */
10     printf("Unesite duzine stranica pravougaonika: ");
11     scanf("%f%f", &a, &b);
12
13     /* Obim */
14     obim=2*(a+b);
15
16     /* Povrsina */
17     povrsina=a*b;
18
19     /* Ispisujemo trazene vrednosti */
20     printf("Obim: %.2f\n", obim);
21     printf("Povrsina: %.2f\n", povrsina);
22
23     /* Završavamo sa programom */
24     return 0;
25 }
```

Rešenje 1.1.22

```
1  #include <stdio.h>
2  #include <math.h>
3
4  /* Zaglavlje math.h sadrži deklaracije velikog broja
5   matematičkih funkcija i konstanti. U ovom zadatku se
6   koristi zbog konstante pi (M_PI)
7  */
```

```
9      Ukoliko se koristi i neka funkcija matematicke
      biblioteke, za prevodjenje je neophodno ukljuciti
      opciju -lm
11     npr. gcc primer.c -lm
      */
13     int main()
      {
15         float r;
16         float O;
17         float P;
18         printf("Unesite duzinu poluprecnika kruga:");
19         scanf("%f", &r);

21         O=2*r*M_PI;
22         P=r*r*M_PI;

23         printf("Obim: %.2f, površina: %.2f\n",O,P);

25         return 0;
27     }
```

Rešenje 1.1.23

```
#include <stdio.h>
2 #include <math.h>
int main()
4 {
    float a;
    float P,O;

6     printf("Unesi duzinu stranice trougla:");
    scanf("%f",&a);

10     O = 3*a;
11     P = (a*a*sqrt(3))/4;

14     printf("Obim: %.2f\n",O);
    printf("Površina: %.2f\n",P);
16     return 0;
}
```

Rešenje 1.1.24

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

4 int main(){
    float a, b, c;
6     float obim, s, površina;
```

2 Kontrola toka

```
8  /* Ucitavamo potrebne podatke */
   printf("Unesite duzine stranica trougla: ");
10 scanf("%f%f%f", &a, &b, &c);

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

14
16 /* Povrsina - koristicemo Heronov obrazac*/
   s=obim/2;
   povrsina=sqrt(s*(s-a)*(s-b)*(s-c));

18
20 /* Ispisujemo trazene vrednosti */
   printf("Obim: %.2f\n", obim);
   printf("Povrsina: %.2f\n", povrsina);

22
24 return 0;
}
```

Rešenje 1.1.26

```
1  #include<stdio.h>

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

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

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

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

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

       */
29     as=(a+b+c)/3.0;
```



```
31  /*
32  Trazeni kolicnik mozemo dobiti na razne nacine:
33  as=1.0*(a+b+c)/3;
34  ili
35  as=(0.0+a+b+c)/3;
36  ili
37  as=((float)(a+b+c))/3;
38  itd.
39  */
40
41  printf("Aritmeticka sredina unetih brojeva je %.2f\n", as);
42  return 0;
43  }
```

Rešenje 1.1.27

```
1  #include <stdio.h>
2
3  int main(){
4      unsigned duzina, sirina, visina;
5      unsigned cena;
6      float povrsina_za_krecenje;
7      float ukupna_cena;
8
9      /* Ucitavamo duzinu, sirinu i visinu sobe */
10     printf("Unesite dimenzije sobe: ");
11     scanf("%u%u%u", &duzina, &sirina, &visina);
12
13     /* Ucitavamo cenu krecenja */
14     printf("Unesite cenu po m2: ");
15     scanf("%u", &cena);
16
17     /* Povrsina za krecenje odgovara povrshini kvadra -
18        bez poda jer se on ne kreci */
19     povrsina_za_krecenje=0.8*(duzina*sirina+
20                               2*duzina*visina+
21                               2*sirina*visina);
22     ukupna_cena=povrsina_za_krecenje*cena;
23
24     /* Ispisujemo trazene podatke */
25     printf("Moler treba da okreći %.2f m2\n",
26           povrsina_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
5  int main()
6  {
7      unsigned x, p;
8      char c;
9      unsigned levo, desno;
10     unsigned novo_x;
11
12     /*
13      Ucitavamo potrebne vrednosti. Sa unosom podataka tipa
14      char moramo biti pazljivi i o tome ce vise biti reci
15      u narednim poglavljima kod zadatka za rad sa funkcijama
16      getchar i putchar. Zbog toga cemo ovde za ucitavanje
17      podataka zatraziti da podatke razdvajamo blanko znakovima
18      (a ne znakom za novi red, zarezom ili nekim drugim separatorom).
19      Ovaj zahtev navodimo u format stringu funkcije scanf tako sto
20      specifikatore promenljivih razdvajamo blanko znakovima.
21
22      Ukoliko specifikatore promenljivih u format stringu pisemo
23      spojeno, tada ih prilikom unosa mozemo razdvojiti bilo kojim
24      karakterom. Zbog toga blanko znakove u format stringu funkcije
25      scanf treba izbegavati i ovo je redak slucaj kada je njihova
26      upotreba opravdana.
27
28      Ako zelimo da odstampamo znak " , u format stringu
29      funkcije printf navodimo \".
30     */
31     printf("Unesite vrednosti u formatu \"x p c\": ");
32     scanf("%u %u %c", &x, &p, &c);
33
34     /*
35      Kada ucitavamo karaktersku promenljivu, njena numericka
36      vrednost je jednaka ASCII kodu unetog karaktera. Na primer,
37      ako karakter '0' ucitamo u promenljivu c, njena numericka
38      vrednost bice 48. Da bismo pretvorili ovu numericku vrednost
39      u numericku vrednost koja odgovara cifri, od nje oduzimamo
40      ASCII kod karakterske konstante '0' koji iznosi upravo 48.
41     */
42     c=c-'0';
43
44
45     /* Odredjujemo deo broja koji se nalazi desno od pozicije p */
46     desno=x%(unsigned)pow(10, p);
47
48     /* Odredjujemo deo broja koji se nalazi levo od pozicije p */
49     levo=x/(unsigned)pow(10, p);
```

```

51  /* Odredjujemo novi broj */
    novo_x=levo*(unsigned)pow(10, p+1) +c*(unsigned)pow(10, p) + desno;
53
    /* Ispisujemo dobijenu vrednost */
55    printf("Rezultat je: %u\n", novo_x);
57
    /* Završavamo sa programom */
    return 0;
59
}
```

Rešenje 1.1.29

```

1  #include <stdio.h>
3  int main()
4  {
5      int a,b,resultata,resultatb,resultatc;
7
8      printf("Unesite dva cela broja:");
9      scanf("%d%d", &a,&b);
10
11     /*
12      Izraz a!=b ima vrednost 1 ako je ova
13      relacija tacna, a 0 ako je netacna
14     */
15     rezultatata = a!=b;
16
17     /*
18      Izraz a%2==0 && b%2==0 je konjunkcija
19      koja se sastoji od dve relacije jednakosti.
20      Izraz a%2==0 ima vrednost 1 ako je
21      ova relacija tacna, a 0 u suprotnom.
22     */
23     rezultatb = (a%2==0 && b%2==0);
24
25     /*
26      Izraz a>0 && a<=100 && b>0 && b<=100 konjunkcija
27      koja se sastoji od cetiri konjunkata. Svaki od
28      konjunkata je izraz koji sadrzi relacioni operator
29      i ima vrednost 1 ako relacija vazi a 0 ako ne vazi
30     */
31     rezultatc = (a>0 && a<=100 && b>0 && b<=100);
32
33     printf("a) rezultat=%d\n", rezultatata);
34     printf("b) rezultat=%d\n", rezultatb);
35     printf("c) rezultat=%d\n", rezultatc);
36
37     return 0;
```

39 }

Rešenje 1.1.30

```
1  #include <stdio.h>
3  int main()
4  {
5      int a,b,max;
6      printf("Unesite dva cela broja:");
7      scanf("%d%d", &a,&b);
9
10     /*
11      Ternarni operator uslova :? koristi se u sledecem
12      obliku:
13
14          izraz1 ? izraz2 : izraz3;
15
16      Izraz izraz1 se izracunava prvi. Ako je njegova
17      vrednost razlicita od nule (tj. ako ima istinitosnu
18      vrednost tacno), onda se izracunava vrednost izraza
19      izraz2 i to je vrednost citavog uslovnog izraza. U
20      suprotnom, izracunava se vrednost izraz3 i to je
21      vrednost citavog uslovnog izraza
22
23     */
24
25     max = (a>b) ? a : b;
26
27     printf("Maksimum je %d\n", max);
28 }
```

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     /*
11     Odredjujemo minimalnu i maksimalnu vrednost
12     unetih brojeva
13
14     */
15 }
```

```

13  */
15  min = (a<b) ? a : b;
17  max = (a>b) ? a : b;

17  /*
19  Racunamo vrednost promenljive rez
21  */
23  rez = (min+0.5)/(1+max*max);

25  printf("Rezultat je %.2f\n", rez);

return 0;
}

```

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);

9      /* Promenljiva min dobija vrednost promenljive a. */
10     min=a;

11     /*
13     Ako je b<a, promenljiva min ce promeniti vrednost
15     tj. bice joj dodeljena vrednost promenljive b. U
17     suprotnom, vrednost ostaje ista.
19     */

21     if (b<a)
22         min = b;

23     printf("Minimum je %d\n",min);

return 0;
}

```

Rešenje 2.1.2 Rešenje je analogno rešenju broj 2.1.1.

Rešenje 2.1.3

```

1  #include<stdio.h>

3  int main()

```

2 Kontrola toka

```
{
5     float x;
      float apsolutno_x;

7

      printf("Unesite jedan realan broj:");
9      scanf("%f",&x);

11     apsolutno_x=x;
      if (x<0)
13         apsolutno_x=-x;

15     printf("Njegova apsolutna vrednost je %.2f\n",apsolutno_x);

17

18     /*
19      2. nacin, pomocu funkcije fabs za koju je neophodno
      ukljuciti zaglavlje math.h:
21      apsolutno_x=fabs(x);
      */
23     return 0;
}
```

Rešenje 2.1.4

```
1  #include <stdio.h>

3  int main()
  {
5      int x;
      float rx;

7

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

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

17     */

19     /*
      Proveravamo da li je uneti broj jednak nuli. Ako
21     jeste, prekidamo sa daljim izvršavanjem programa
      navodjenjem naredbe return. Argument -1 u naredbi
23     return oznacava da program nije uspesno završen

      */
25     if (x==0)
      {
27         printf("Nedozvoljeno deljenje nulom\n");
      }
```

```

    return -1;
29 }

31 /*
    Primenom operatora / na argumente 1 i x dobijamo
33 rezultat celobrojnog deljenja ovih argumenata. Da
    bismo dobili kolicnik, koji je realna vrednost,
35 neophodno je da jedan od argumenata zapisemo kao
    realnu vrednost, npr celobrojnu vrednost 1 zapisemo kao
37 realnu vrednost 1.0. Ovakav postupak se naziva
    implicitna konverzija.
39 */

41 rx = 1.0/x;
    printf("Recipročna vrednost unetog broja: %.4f\n", rx);
43
    return 0;
45 }

```

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     /*
        U naredbi dodele s=s+a vrednost izraza sa desne strane
15 znaka jednakosti dodeljujemo promenljivoj sa leve
        strane znaka jednakosti. Staru vrednost promenljive s
17 saberemo sa vrednoscu promenljive a i dobijenu vrednost
        upisemo u promenljivu s.
19     */

21     if (a>0)
        s=s+a;

23     /* s+=b je skraceni zapis za s=s+b */

25     if (b>0)
        s+=b;

27     if (c>0)
        s+=c;

29

```

```
31     printf("Suma unetih pozitivnih brojeva: %d\n",s);
33     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;

    printf("Unesite cene tri artikla:");
    scanf("%u%u%u",&a,&b,&c);

    12 /*
        Racunamo minimum tri broja. Dodeljujemo promenljivoj
    14 min vrednost prvog broja.
    */
    16 min=a;

    18 /*
        Ako je drugi broj manji od minimuma, to znaci da
    20 promenljiva min ne sadrzi najmanji broj. Dodeljujemo
        joj vrednost drugog broja.
    22 */
    24 if(min>b)
        min=b;

    26 /*
        Ako je treci broj manji od minimuma, to znaci da
    28 promenljiva min ne sadrzi najmanji broj. Dodeljujemo
        joj vrednost treceg broja.
    30 */
    32 if(min>c)
        min=c;

    34 cena_bez_popusta=a+b+c;
    cena_sa_popustom = cena_bez_popusta - min + 1;

    36 printf("Cena sa popustom: %u\nUsteda: %u\n",
    38         cena_sa_popustom, cena_bez_popusta-cena_sa_popustom);

    40     return 0;
}
```


Rešenje 2.1.7

```
1  #include <stdio.h>
3  int main()
4  {
5      int n, j, d, s, h, max;
7      /* Ucitavamo broj */
8      printf("Unesite cetvorocifreni broj: ");
9      scanf("%d", &n);
11
12     /*
13      Za slucaj da je broj negativan, uzimamo apsolutnu
14      vrednost unetog broja
15     */
16     n = abs(n);
17     /*
18      Ako uneti broj nije cetvorocifren, ispisujemo poruku
19      o gresci i prekidamo izvršavanje programa.
20     */
21     if(n<1000 || n>9999)
22     {
23         printf("Greska: Niste uneli cetvorocifren broj!\n");
24         return -1;
25     }
27     /*
28      Ako je broj cetvorocifren, izdvajamo cifre broja:
29      j -jedinice, d - desetice, s - stotine i h - hiljade
30     */
31     j=n%10;
32     d=(n/10)%10;
33     s=(n/100)%10;
34     h=n/1000;
35
36     /* Odredjujemo maksimalnu cifru */
37     max=j;
38
39     if(d>max)
40         max=d;
41
42     if(s>max)
43         max=s;
44
45     if(h>max)
46         max=h;
47
48     /* II nacin:
49     * if(j>d && j>s && j>h)
50     *     max=j;
```

```
51     * if(d>j && d>s && d>h)
52     *   max=d;
53     * if(s>j && s>d && s>h)
54     *   max=s;
55     * if(h>j && h>d && h>s)
56     *   max=h;
57     */
58
59     /* Ispisujemo rezultat */
60     printf("Najveca cifra je: %d\n", max);
61
62     return 0;
63 }
```

Rešenje 2.1.8

```
#include<stdio.h>
2
3 int main()
4 {
5     int sati;
6     int minuti;
7     int preostali_sati;
8     int preostali_minuti;
9
10    /*
11     Ukoliko naredbu printf zelimo da napisemo u dva
12     reda, i tom prilikom prekidamo deo pod navodnicima,
13     to mozemo uraditi navodjenjem navodnika na kraju
14     prvog i na pocetku narednog reda:
15    */
16
17    printf("Unesite vreme (broj sati u intervalu [0,24),\n"
18           "broj minuta u intervalu [0,60)):");
19    scanf("%d%d",&sati,&minuti);
20
21    /*
22     U slucaju da je unos neispravan, ispisujemo
23     poruku o gresci i prekidamo dalje izvršavanje
24     programa.
25
26     Uslov u if naredbi je disjunkcija (operator ||)
27     sastavljena od 4 disjunkata. Svaki od njih je
28     izraz sa relacionim operatorom i ima vrednost
29     1 ako je izraz tacan i 0 u suprotnom. Da bi
30     disjunkcija bila tacna, bar jedan od disjunkata
31     mora da bude tacan. Zbog lenjog izracunavanja,
32     vrednost disjunkata ce biti racunata do vrednosti
33     prvog disjunkta koji je tacan. To je znak da
34     je uslov u if naredbi ispunjen i nema potrebe
35     racunati vrednosti drugih disjunkata.
36    */
37 }
```

```

36  */
38  if (sati>24 || sati<0 || minuti>59 || minuti<0)
39  {
40      printf("Neispravan unos.\n");
41      return -1;
42  }
44  preostali_sati = 24-sati-1;
45  preostali_minuti = 60-minuti;
46
47  if (preostali_minuti==60)
48  {
49      preostali_sati++;
50      preostali_minuti=0;
51  }
52
53  /*
54   Ukoliko naredbu printf zelimo da napisemo u dva
55   reda i pritom ne prekidamo deo pod navodnicima,
56   to mozemo uraditi bez navodjenja dodatnih karaktera:
57  */
58  printf("Do ponoci je ostalo %d sati i %d minuta\n",
59        preostali_sati, preostali_minuti);
60
61  return 0;
62 }

```

Rešenje 2.1.9

```

1  #include <stdio.h>
2
3  int main()
4  {
5      char c;
6      printf("Unesite jedan karakter:");
7      scanf("%c", &c);
8
9      printf("Uneti karakter: %c, njegov ASCII kod: %d\n", c, c);
10
11     /*
12      Razlika izmedju ASCII koda svakog malog i odgovarajuceg
13      velikog slova je konstanta koja se moze sracunati
14      izrazom 'a'-'A' (i iznosi 32)
15     */
16
17     if (c>='A' && c<='Z')
18     printf("odgovarajuce malo slovo: %c, njegov ASCII kod: ",
19           "%d\n", c+('a'-'A'), c+('a'-'A'));
20
21     if (c>='a' && c<='z')

```

2 Kontrola toka

```
22     printf("odgovarajuće veliko slovo: %c, njegov ASCII kod: ",
23            "%d\n", c-('a'-'A'), c-('a'-'A'));
24
25     return 0;
26 }
```

Rešenje 2.1.10

```
1  #include <stdio.h>
2
3  int main(){
4
5      char c1, c2, c3, c4, c5;
6      int broj_malih_slova=0;
7
8
9      /* Citamo karaktere */
10     printf("Unesite karaktere: ");
11     scanf("%c %c %c %c %c", &c1, &c2, &c3, &c4, &c5);
12
13     /* Proveravamo da li je prvi karakter malo slovo */
14     if(c1>='a' && c1<='z'){
15         /* I ako jeste, uvecavamo broj malih slova */
16         broj_malih_slova++;
17     }
18
19     /* Proveravamo da li je drugi karakter malo slovo */
20     if(c2>='a' && c2<='z'){
21         /* I ako jeste, uvecavamo broj malih slova */
22         broj_malih_slova++;
23     }
24
25     /* Proveravamo da li je treci karakter malo slovo */
26     if(c3>='a' && c3<='z'){
27         /* I ako jeste, uvecavamo broj malih slova */
28         broj_malih_slova++;
29     }
30
31     /* Proveravamo da li je cetvrti karakter malo slovo */
32     if(c4>='a' && c4<='z'){
33         /* I ako jeste, uvecavamo broj malih slova */
34         broj_malih_slova++;
35     }
36
37     /* Proveravamo da li je peti karakter malo slovo */
38     if(c5>='a' && c5<='z'){
39         /* I ako jeste, uvecavamo broj malih slova */
40         broj_malih_slova++;
41     }
42
43     /* Ispisujemo rezultat */
```

```
45     printf("Broj malih slova: %d\n", broj_malih_slova);  
47     return 0;  
}
```

Rešenje 2.1.11

```
1  #include <stdio.h>  
2  #include <ctype.h>  
3  
4  int main()  
5  {  
6      /*  
7       Broj pojavljivanja slova a i A se  
8       inicijalizuje na 0  
9      */  
10     int br_a = 0;  
11  
12     /*  
13     Funkcija getchar ucitava jedan karakter.  
14     Njena povratna vrednost je ASCII kod  
15     ucitanog karaktera.  
16  
17     Funkcija tolower za dati karakter vraca:  
18     - odgovarajuce malo slovo, ako je dati  
19       karakter veliko slovo  
20     - taj isti karakter, u suprotnom  
21     Ova funkcija je definisana u biblioteci ctype.h  
22  
23     U slucaju da je uslov ispunjen, uvecavamo  
24     brojac br_a za jedan pomocu operatora  
25     inkrementacije ++  
26     */  
27     if (tolower(getchar()) == 'a')  
28         br_a++;  
29     if (tolower(getchar()) == 'a')  
30         br_a++;  
31     if (tolower(getchar()) == 'a')  
32         br_a++;  
33     if (tolower(getchar()) == 'a')  
34         br_a++;  
35     if (tolower(getchar()) == 'a')  
36         br_a++;  
37     printf("%d\n", br_a);  
38  
39     return 0;  
41 }
```

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 vraća 1 ako je
10     dati karakter cifra i 0 u suprotnom.
11     Nalazi se u biblioteci ctype.h.
12     */
13     if (isdigit(getchar()))
14         br_cif++;
15     if (isdigit(getchar()))
16         br_cif++;
17     if (isdigit(getchar()))
18         br_cif++;
19     if (isdigit(getchar()))
20         br_cif++;
21     if (isdigit(getchar()))
22         br_cif++;
23
24     printf("%d\n", br_cif);
25
26     return 0;
27 }
```

Rešenje 2.1.13

```
1 #include <stdio.h>
2
3 int main()
4 {
5     int x;
6     printf("Unesite godinu:");
7     scanf("%d", &x);
8
9     if ((x%4==0 && x%100!=0) || x%400==0)
10         printf("Godina je prestupna\n");
11     else
12         printf("Godina nije prestupna\n");
13
14     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
14      uneti broj negativan
15     */
16     n = abs(n);
17
18     /*
19      Ako broj nije trocifren, izdajemo poruku o gresci i
20      prekidamo dalje izvršavanje programa
21     */
22     if(n<100 || n>999)
23     {
24         printf("Greska: Niste uneli trocifren broj!\n");
25         return -1;
26     }
27
28     /*
29      Izdvajamo cifre broja:
30      j -jedinice, d - desetice, s - stotine
31     */
32     j=n%10;
33     d=(n/10)%10;
34     s=n/100;
35
36     /* Proveravamo da li je broj Armstrongov */
37     if(n==j*j*j+d*d*d+s*s*s)
38         printf("Broj je Armstrongov.\n");
39     else
40         printf("Broj nije Armstrongov.\n");
41
42     return 0;
43 }
```

Rešenje 2.1.15

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

```
3 int main()
4 {
5     int n, j, d, s, h;
6     int broj_parnih, proizvod_parnih;
7
8     printf("Unesite cetvorocifreni broj: ");
9     scanf("%d", &n);
10
11     n = abs(n);
12
13     if(n<1000 || n>9999)
14     {
15         printf("Broj nije cetvorocifren.\n");
16         return -1;
17     }
18
19     /* Izdvajamo cifre broja:
20      j -jedinice, d - desetice, s - stotine i h - hiljade
21     */
22     j=n%10;
23     d=(n/10)%10;
24     s=(n/100)%10;
25     h=n/1000;
26
27     /* Inicijalizujemo broj parnih cifara na 0 */
28     broj_parnih=0;
29     /* Postavljamo proizvod parnih cifara na 1 (neutral za mnozenje)
30     */
31     proizvod_parnih=1;
32
33     /* Proveravamo da li je cifra jedinica parna */
34     if(j%2==0){
35         proizvod_parnih=proizvod_parnih*j;
36         broj_parnih++;
37     }
38
39     /* Proveravamo da li je cifra desetica parna */
40     if(d%2==0){
41         proizvod_parnih=proizvod_parnih*d;
42         broj_parnih++;
43     }
44
45     /* Proveravamo da li je cifra stotina parna */
46     if(s%2==0){
47         proizvod_parnih=proizvod_parnih*s;
48         broj_parnih++;
49     }
50
51     /* Proveravamo da li je cifra hiljada parna */
52     if(h%2==0){
53         proizvod_parnih=proizvod_parnih*h;
54         broj_parnih++;
55     }
```



```

    }

55     /* Proveravamo da li u zapisu broja ima parnih cifara i
    ispisujemo
57     rezultat */
    if(broj_parnih==0){
59         printf("Nema parnih cifara.\n");
    }
61     else{
        printf("Proizvod parnih cifara: %d\n", proizvod_parnih);
63     }

65     return 0;
}

```

Rešenje 2.1.16

```

1  #include <stdio.h>

3  int main()
4  {
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.");
13         return -1;
    }

15

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

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

27

        if (b > max)
29         {
            max = b;
31             stepenMax = 10;
        }

33         if (b < min)
        {

```

2 Kontrola toka

```
35     min = b;
36     stepenMin = 10;
37 }
38
39 if (c > max)
40 {
41     max = c;
42     stepenMax = 100;
43 }
44 if (c < min)
45 {
46     min = c;
47     stepenMin = 100;
48 }
49
50 if (d > max)
51 {
52     max = d;
53     stepenMax = 1000;
54 }
55 if (d < min)
56 {
57     min = d;
58     stepenMin = 1000;
59 }
60
61 int rez;
62 /* Ideja:
63    4179, mesta menjamo tako sto oduzmemo 9 i dodamo 1,
64    ^ ^   odnosno oduzemo 100 i dodamo 900 */
65
66 if (broj > 0)
67     rez = broj - max*stepenMax + min*stepenMax
68             - min*stepenMin + max*stepenMin;
69 else
70     rez = broj + max*stepenMax - min*stepenMax
71             + min*stepenMin - max*stepenMin;
72
73 printf("%d\n",rez);
74
75 return 0;
76 }
77 }
```

Rešenje 2.1.20

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

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

9  if (x==0)
    printf("Broj je jednak nuli\n");
11 else if (x<0)
    printf("Broj je manji od nule\n");
13 else
    printf("Broj je veci od nule\n");

15 return 0;
17 }
```

Rešenje 2.1.21

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

3
   int main()
5  {
     float a,b,c;
     float D;
     float x1,x2;
     printf("Unesite koeficijente A, B i C:");
     scanf("%f%f%f",&a,&b,&c);

11
     /*
13      Proveravamo da li je kvadratna jednacina
        korektno zadata.
15     */
     if (a==0)
17         if (b==0)
            /* slucaj a==0 && b==0 && c==0 */
19             if(c==0)
                printf("Jednacina ima beskonacno mnogo resenja\n");
                /* slucaj a==0 && b==0 && c!=0 */
21             else
                printf("Jednacina nema resenja\n");
            /* slucaj a==0 && b!=0 */
23             else
            {
25                 {
27                     x1=-c/b;
                        printf("Jednacina ima jedinstveno realno resenje %.2f\n",
29                             x1);
                    }
                }
31             /* slucaj a!=0 */
            else
            {
33                 D=b*b-4*a*c;
35                 if (D<0)
                    printf("Jednacina nema realnih resenja\n");
```

2 Kontrola toka

```
37     else if (D>0)
38     {
39         /* funkcija sqrt nalazi se u biblioteci math.h
40            (prevodjenje sa -lm opcijom) */
41         x1 = (-b+sqrt(D))/(2*a);
42         x2 = (-b-sqrt(D))/(2*a);
43         printf("Jednacina ima dva razlicita realna resenja %.2f ",
44                "i %.2f\n",x1,x2);
45     }
46     else
47     {
48         x1 = (-b)/(2*a);
49         printf("Jednacina ima jedinstveno realno resenje %.2f\n",x1);
50     }
51 }
52
53 return 0;
54 }
```

Rešenje 2.1.22

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

int main()
{
    int x;
    char c1;
    char c10;
    char c100;
    char c1000;

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

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

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

    /* Izdvajamo cifre broja. */
    c1 = x%10;
    c10 = (x/10)%10;
```

```

32  c100 = (x/100)%10;
    c1000 = (x/1000)%10;

34  if (c1000<=c100 && c100<=c10 && c10<=c1)
    printf("Cifre su uredjene neopadajuće \n");
36  else if (c1000>=c100 && c100>=c10 && c10>=c1)
    printf("Cifre su uredjene nerastuće \n");
38  else
    printf("Cifre nisu uredjene\n");
40
42  return 0;
}

```

Rešenje 2.1.23

```

1  #include <stdio.h>

3  int main(){

5      char c;

7      printf("Unesite karakter: ");
      scanf("%c", &c);

9      if(c>='a' && c<='z')
11         printf("%c\n", c-'a'+ 'A');
      else if(c>='A' && c<='Z')
13         printf("%c\n", c-'A'+ 'a');
      else if(c>='0' && c<='9')
15         printf("cifra\n");
17     /*
        Ako nijedan od prethodnih uslova nije ispunjen,
        bice izvršena naredba u else grani
19     */
      else
21         printf("%%%c\n",c);

23     return 0;
}

```

Rešenje 2.1.24

```

1  #include <stdio.h>

3  int main(){
    int k, n, broj;

5

7      printf("Unesite k: ");
      scanf("%d", &k);

```

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

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

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

           9 oduzimamo jer je 9 broj cifara potrebnih za
           zapis jednocifrenih brojeva
25          */
           n=(k-9)/2;
27          /* Broj o kojem se radi je */
           broj=9+n;
29          /* Ujedno, za neparno k se trazi cifra jedinica
           izdvojenog broja */
           printf("Na %d-toj poziciji je broj %d.\n", k, broj%10);
31          }
33          else{
           /* Ako je k paran broj, zapisan je ceo broj
           dvocifrenih brojeva i zapoceto je sa zapisom sledeceg
           9 oduzimamo jer je 9 broj cifara potrebnih za zapis
           jednocifrenih brojeva
35          */
           n=(k-9)/2 +1;
37          /* Broj o kojem se radi je */
           broj= 9 + n;
39          /* Ujedno, za parno k se trazi cifra desetica
           izdvojenog broja */
           printf("Na %d-toj poziciji je broj %d.\n", k, broj/10);
41          }
43          }
45          }
47          }
49          }
51          }
53          }
55          }
57          printf("Greska: Nedozvoljena vrednost broja k!\n");
59          }
```

```

    return 0;
61 }

```

Rešenje 2.1.25

```

1  #include <stdio.h>
   #include <math.h>
3
   int main(){
5     float x;
   int k;
7     float F;

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

11     /* Proveravamo vrednost za k */
13     if(k<1 || k>3){
   printf("Greska: nedozvoljena vrednost za k!\n");
15     return 0;
   }
17     printf("F(%f,%d)= ", x, k);

19     /* Analiziramo moguće slučajeve */
   if(k==1){
21         F=2*cos(x)-x*x*x;
   }
23     else{
   if(k==2){
25         x=2*cos(x)-x*x*x;
         F=2*cos(x)-x*x*x;
27     }
   else{
29         x=2*cos(x)-x*x*x;
         x=2*cos(x)-x*x*x;
31         F=2*cos(x)-x*x*x;
   }
33 }

35     /* Ispisujemo rezultat */
   printf("%f\n", F);
37
   return 0;
39 }

```

Rešenje 2.1.26

```

1  #include <stdio.h>

```

2 Kontrola toka

```
3 int main(){
5     int dan;
7     printf("Unesite broj: ");
9     scanf("%d", &dan);
11    switch(dan)
12    {
13        case 1:
14            printf("ponedeljak\n");
15            break;
16        case 2:
17            printf("utorak\n");
18            break;
19        case 3:
20            printf("sreda\n");
21            break;
22        case 4:
23            printf("cetvrtak\n");
24            break;
25        case 5:
26            printf("petak\n");
27            break;
28        case 6:
29            printf("subota\n");
30            break;
31        case 7:
32            printf("nedelja\n");
33            break;
34        default:
35            printf("Greska: nedozvoljeni unos!\n");
36    }
37    return 0;
39 }
```

Rešenje 2.1.27

```
1 #include <stdio.h>
3 int main()
4 {
5     char c;
6     printf("Unesite jedan karakter:");
7     scanf("%c", &c);
9     /*
10      Da bismo utvrditi da li je karakter samoglasnik,
11      neophodno je proveriti da li odgovara nekom od
```



```

13     sledecih karaktera: A,E,I,O,U,a,e,i,o,u
14 */
15     switch(c)
16     {
17         case 'A' :
18         case 'E' :
19         case 'I' :
20         case 'O' :
21         case 'U' :
22         case 'a' :
23         case 'e' :
24         case 'i' :
25         case 'o' :
26         case 'u' : printf("Uneti karakter je samoglasnik\n");
27                     break;
28         default : printf("Uneti karakter nije samoglasnik\n");
29                     break;
30     }
31
32     return 0;
33 }

```

Rešenje 2.1.28

```

1  #include <stdio.h>
2
3  int main()
4  {
5
6      char op;
7      int x, y;
8
9      printf("Unesite operator i dva cela broja: ");
10     scanf("%c %d %d", &op, &x, &y);
11
12     switch(op){
13         case '+':
14             printf("Rezultat je: %d\n", x+y);
15             break;
16         case '-':
17             printf("Rezultat je: %d\n", x-y);
18             break;
19         case '*':
20             printf("Rezultat je: %d\n", x*y);
21             break;
22         case '/':
23             if(y==0)
24                 printf("Greska: deljenje nulom nije dozvoljeno!\n");
25             else
26                 printf("Rezultat je: %f\n", x*1.0/y);
27             break;

```

```
    case '!':
29         printf("Rezultat je: %d\n", x%y);
        break;
31     default:
        printf("Greska: nepoznat operator!\n");
33 }
35 return 0;
}
```

Rešenje 2.1.29

```
1  #include <stdio.h>
3  int main()
4  {
5      int d,m;
        printf("Unesite dan i mesec");
7      scanf("%d%d",&d,&m);
9      /*
11         Argument u naredbi switch mora biti celobrojna promenljiva,
            dok argument u naredbi case mora biti celobrojna
            konstanta.
13     */
        switch(m)
15     {
17         /* Ispitujemo da li vazi m==1 ili m==2*/
            case 1:
            case 2:
19                 printf("zima\n");
                    break;
21             case 3:
                    if (d<21)
23                         printf("zima\n");
                    else
25                         printf("prolece\n");
                    break;
27             case 4:
            case 5:
29                 printf("prolece\n");
                    break;
31             case 6:
                    if (d<21)
33                         printf("prolece\n");
                    else
35                         printf("leto\n");
                    break;
37             case 7:
            case 8:
39                 printf("leto\n");
```

```
        break;
41     case 9:
        if (d<23)
43         printf("leto\n");
        else
45         printf("jesen\n");
        break;
47     case 10:
48     case 11:
        printf("jesen\n");
        break;
51     case 12:
        if (d<22)
53         printf("jesen\n");
        else
55         printf("zima\n");
    }
57
    return 0;
59 }
```

Rešenje 2.1.30

```
1  #include <stdio.h>
   #include <stdlib.h> // Potrebno za exit funkciju
3
4  int main()
5  {
6      int godina;
7      int mesec;
8
9      int prestupna;
10
11     printf("Unesite godinu: ");
12     scanf("%d",&godina);
13
14     if(godina < 0)
15     {
16         printf("Lose uneta godina!\n");
17         return -1;
18     }
19
20     /* Provera da li je godina prestupna, zbog februara */
21     if((godina % 4 == 0 && godina % 100 != 0)
22        || godina % 400 == 0)
23         prestupna = 1;
24     else
25         prestupna = 0;
26
27     printf("Unesite redni broj meseca: ");
28     scanf("%d",&mesec);
```

```
29     switch(mesec)
30     {
31         case 1: printf("Januar, 31 dan\n");
32                 break;
33         case 2:
34             if(prestupna)
35                 printf("Februar, 29 dana\n");
36             else
37                 printf("Februar, 28 dana\n");
38             break;
39
40         case 3: printf("Mart, 31 dan\n");
41                 break;
42         case 4: printf("April, 30 dana\n");
43                 break;
44         case 5: printf("Maj, 31 dan\n");
45                 break;
46         case 6: printf("Jun, 30 dana\n");
47                 break;
48         case 7: printf("Jul, 31 dan\n");
49                 break;
50         case 8: printf("Avgust, 31 dan\n");
51                 break;
52         case 9: printf("Septembar, 30 dana\n");
53                 break;
54         case 10: printf("Oktobar, 31 dan\n");
55                 break;
56         case 11: printf("Novembar, 30 dana\n");
57                 break;
58         case 12: printf("Decembar, 31 dan\n");
59                 break;
60         default: printf("Lose unet redni broj meseca!\n");
61     }
62
63     return 0;
64 }
```

Rešenje 2.1.31

```
1  #include <stdio.h>
2
3  int main(){
4      int dan, mesec, godina, dozvoljen_broj_dana;
5
6      /* Citamo datum */
7      printf("Unesite datum: ");
8      scanf("%d.%d.%d", &dan, &mesec, &godina);
9
10     /* Proveravamo godinu */
11     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
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
50          * novembar je 30 */
51         dozvoljen_broj_dana=30;
52         break;
53 }
54
55 /* Proveravamo dan */
56 if(dan<0 || dan>dozvoljen_broj_dana){
57     printf("Datum nije korektan (neispravan dan)!\n");
58     return 0;
59 }
60
61 /* Sve provere su ispunjene pa zakljucujemo
62    da je datum korektan */
63 printf("Ispravan datum!\n");
64
65 return 0;
```

```
}

```

Rešenje 2.1.32

```
1  #include <stdio.h>
2
3  int main(){
4      int dan, mesec, godina;
5      int prethodni_dan, prethodni_mesec, prethodni_godina;
6
7      /* Citamo datum */
8      printf("Unesite datum: ");
9      scanf("%d.%d.%d", &dan, &mesec, &godina);
10
11     /* Racunamo dan, mesec i godinu prethodnog dana */
12     prethodni_dan=dan-1;
13     prethodni_mesec=mesec;
14     prethodni_godina=godina;
15
16     /* I po potrebi vrsimo korekcije */
17
18     /* Ako je u pitanju prvi u mesecu */
19     if(prethodni_dan==0){
20         /* Treba korigovati mesec */
21         prethodni_mesec=mesec-1;
22         /* Ako je u pitanju januar */
23         if(prethodni_mesec==0){
24             /* Treba korigovati i godinu */
25             prethodni_mesec=12;
26             prethodni_godina=godina-1;
27         }
28
29         /* Analiziramo redni broj meseca kako bi odredili
30            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;

```

```

49         case 4:
50         case 6:
51         case 9:
52         case 11:
53             prethodni_dan=30;
54     }
55 }
56
57 /* Ispisujemo datum koji smo izracunali */
58 printf("Prethodni datum: %d.%d.%d\n",
59        prethodni_dan, prethodni_mesec, prethodni_godina);
60
61 return 0;
62 }

```

Rešenje 2.1.33 Rešenje je analogno rešenju zadatka 2.1.32.

Rešenje 2.1.35

```

1 #include<stdio.h>
2 #include<math.h>
3
4 int main()
5 {
6     char c;
7     float x1,y1,x2,y2,x3,y3,x4,y4;
8     float kab,kbc,kad;
9     float dab,dad;
10    float delta, deltap, deltaq;
11    float 0;
12    float k,n;
13
14    printf("Unesi jedan karakter:");
15    scanf("%c",&c);
16
17    printf("Unesi realne koordinate 4 tacke:");
18    scanf("%f%f%f%f%f%f%f",&x1,&y1,&x2,&y2,&x3,&y3,&x4,&y4);
19
20    switch (c)
21    {
22        case 'k':
23            if (y1==y2 && y3==y4 && x1==x4 && x2==x3)
24            {
25                dab = sqrt(pow(x1-x2,2)+pow(y1-y2,2));
26                dad = sqrt(pow(x1-x4,2)+pow(y1-y4,2));
27                0 = 2*dab + 2*dad;
28                printf("Obim pravougaonika je %f\n",0);
29            }
30        else
31            printf("Tacke ne cine pravougaonik sa stranicama ",

```

```

32         "koje su paralelne koordinatnim osama\n");
33     break;
34     case 'h':
35         /*
36          * Ukoliko se tacke A(x1,y1) i B(x2,y2) ne nalaze
37          * na pravoj koja je paralelna x osi, izracunamo k,n
38          * za pravu odredjenu tackama A(x1,y1) i B(x2,y2)
39          */
40         if ((x1-x2)!=0)
41         {
42             k = (y1-y2)/(x1-x2);
43             n = y1-k*x1;
44             /*
45              * Proverimo da li tacke C(x3,y3) i D(x4,y4) nalaze
46              * na toj pravoj
47              */
48             if (y3==x3*k+n && y4==x4*k+n)
49                 printf("Tacke su kolinearne, pripadaju pravoj ",
50                        "y=%f*x+%f\n",k,n);
51             else
52                 printf("Tacke nisu kolinearne\n");
53         }
54         /*
55          * Ukoliko se A i B nalaze na pravoj koja je paralelna
56          * x osi, proverimo da li tacke C(x3,y3) i D(x4,y4)
57          * nalaze na toj pravoj
58          */
59         else
60             if (x3==x1 && x4==x1)
61                 printf ("Tacke su kolinearne, pripadaju pravoj ",
62                        "x=%f\n",x1);
63             else
64                 printf("Tacke nisu kolinearne\n");
65         break;
66     case 'j':
67         delta = x1*y2-x2*y1;
68         deltap = x2*(y4-y3)-y2*(x4-x3);
69         deltaq = x1*(y4-y3)-y1*(x4-x3);
70         if (delta!=0)
71             printf("Sistem ima jedinstveno resenje %.2f, %.2f\n",
72                    deltap/delta, deltaq/delta);
73         else if (deltap==0 && deltaq==0)
74             printf("Sistem je neodredjen ili nema resenja.\n");
75         else
76             printf("Sistem nema resenja\n");
77         break;
78     default:
79         printf("Nekorektan unos\n");
80 }
81
82 return 0;
83 }

```


Rešenje 2.2.1

```
1  #include <stdio.h>
2
3  int main()
4  {
5      /* Promenljiva i kontrolise koliko puta ce se petlja izvesti i
6       naziva se brojac petlje.
7       Njenu pocetnu vrednost postavljamo na 0 jer se u pocetku petlja
8       nije ni jednom izvela. */
9      int i = 0;
10
11     /* Pre ulaska u telo petlje proverava se da li je
12      ispunjen uslov ulaska u petlju.
13      */
14     while(i < 5)
15     {
16         /* Ukoliko uslov ulaska u petlju jeste ispunjen, izvela se telo
17          petlje. */
18         /* Ispisujemo trazeni tekst. */
19         printf("Mi volimo da programiramo.\n");
20
21         /* Uvecava se brojac za jedan jer je jednom izvelo telo petlje.
22          Ako bi ova vrednost ostala nepromenjena, petlja bi se
23          izvela beskonacno. */
24         i++;
25
26         /* Nakon poslednje naredbe tela petlje ponovo se ispituje uslov
27          petlje.*/
28     }
29
30     return 0;
31 }
```

Rešenje 2.2.2

```
1  #include<stdio.h>
2
3  int main()
4  {
5      /* Brojac u petlji. */
6      int i=0;
7      /* Promenljiva koja oznacava koliko puta cemo ispisati trazeni
8       tekst. */
9      int n;
10
11     printf("Unesite ceo broj: ");
12     scanf("%d", &n);
```

2 Kontrola toka

```
/* Pre ulaska u telo petlje proverava se da li je ispunjen uslov
   ulaska u petlju. */
14 while (i<n)
   {
16     printf("Mi volimo da programiramo.\n");
       i++;
18   }
   return 0;
20 }
```

Rešenje 2.2.3

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

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

13
15    /* U slucaju neispravnih podataka ispisujemo odgovarajucu poruku
       i izlazimo iz programa. */
       if (n < 0)
17     {
19         printf("Neispravan unos. Promenljiva mora biti pozitivna!\n");
           return -1;
       }

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

Rešenje 2.2.4

```
1  /* Resenje pod a). */
3  #include <stdio.h>
5  int main()
6  {
7      /* Promenljive koje oznacavaju granice intervala. */
8      int n,m;
9      /* Promenljiva koja oznacava trenutno ispisani broj intervala. */
10     int i;
12
13     printf("Unesi dva cela broja: ");
14     scanf("%d%d",&n,&m);
16
17     if (m < n)
18     {
19         printf("Neispravan unos. Nisu dobro zadate granice intervala!\n");
20         return -1;
21     }
22
23     /* Na pocetku ispisujemo prvi broj intervala, a to je n. */
24     i=n;
25     /* uslov petlje se proverava pre ulaska u telo petlje */
26     while (i<=m)
27     {
28         printf("%d ", i);
29         i++;
30     }
31
32     printf("\n");
33
34     return 0;
35 }
```

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

2 Kontrola toka

```
14     scanf("%d%d",&n,&m);
16     if (m < n)
17     {
18         printf("Neispravan unos. Nisu dobro zadate granice intervala!\n");
19     };
20     return -1;
21 }
22
23                                     /* naredba i=n se izvršava jednom, pre prve
24     iteracije */
25     for(i=n;i<=m;i++)               /* uslov petlje i<=m se proverava pre svake
26     iteracije */
27     printf("%d ", i);               /* naredba i++ se izvršava nakon svake
28     iteracije */
29     printf("\n");
30
31     return 0;
32 }
```

```
1  /* Resenje pod c). */
2
3  #include <stdio.h>
4
5  int main()
6  {
7
8      /* Promenljive koje oznacavaju granice intervala. */
9      int n,m;
10     /* Promenljiva koja oznacava trenutno ispisani broj intervala. */
11     int i;
12
13     printf("Unesi dva cela broja: ");
14     scanf("%d%d",&n,&m);
15
16     if (m < n)
17     {
18         printf("Neispravan unos. Nisu dobro zadate granice intervala!\n");
19     };
20     return -1;
21 }
22
23
24     /* Uslov petlje se proverava na kraju svake iteracije. */
25     /* Zbog toga se do while petlja izvršava bar jednom, cak i u
26     slucaju
27     da uslov petlje nikada nije ispunjen. */
28     i=n;
```

```

29     do                                /* Petlja se zapocinje bez provere uslova. */
    {
        printf("%d ",i); /* Stampa se vrednost promenljive i. */
31         i++;           /* Uvecava se vrednost promenljive i. */
    }
33     while(i<=m);           /* Proverava se uslov i ukoliko je ispunjen,
        nastavlja se sa sledecom iteracijom. */
                               /* U suprotnom, petlja se završava i program
        se nastavlja od prve naredbe koja sledi za petljom. */
35     printf("\n");
37     return 0;
}

```

Rešenje 2.2.5

```

1  #include<stdio.h>
3  int main()
    {
5      int x;
        /* U promenljivoj f se pamti izracunati faktoriyel.
7         * Kako faktoriyel je jako veliki broj, za tip podataka
        * se uzima unsigned long, da bi mogla da se upise
9         * sto veca vrednost. */
        unsigned long f;
11         int i;
        int original;
13
        printf("Unesite pozitivan broj: ");
15         scanf("%d",&x);
17
        if (x < 0)
        {
19             printf("Nekorektan unos\n");
            return -1;
21         }
23
        if (x >= 22)
        {
25             printf("Broj je veliki, dolazi do prekoracenja.\n");
            return -1;
27         }
29
        original=x;
        f=1;
31
        while (x>1)
33         {
            f=f*x;
35             x--;
        }
    }

```

2 Kontrola toka

```
    }  
37     printf("Faktorijel = %lu\n", f);  
39  
41  
43     return 0;  
}
```

Rešenje 2.2.6

```
1  #include <stdio.h>  
3  int main(){  
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 );  
13  
14    if (n < 0)  
15    {  
16        printf("Neispravan unos.\n");  
17        return -1;  
18    }  
19  
20    /* Pocetna vrednost stepena koji se racuna. */  
21    vrednost=1;  
22  
23    for(exp=1; exp<=n; exp++)  
24        vrednost=vrednost*x;  
25  
26    printf("%f\n",vrednost);  
27  
28    return 0;  
}
```

Rešenje 2.2.7

```
1  #include <stdio.h>  
3  int main(void){  
5     int n, n_abs;  
6     float x;  
7     float vrednost;
```

```

9      unsigned exp;

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

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

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

18     /* Ukoliko je stepen bio negativan treba
19      * odrediti 1/x^n, sto je zapravo 1/vrednost. */
20     if(n<0){
21         printf("%.3f\n",1/vrednost);
22     }
23     else{
24         printf("%.3f\n",vrednost);
25     }

26     return 0;
27 }

```

Rešenje 2.2.8

```

1  #include<stdio.h>
2  #include<math.h>
3
4  int main()
5  {
6      int x;
7      /* Brojac u petlji. */
8      int i;
9
10     /* Ucitavamo broj. */
11     printf("Unesi ceo broj veci od 0: ");
12     scanf("%d", &x);
13
14     if (x<=0)
15     {
16         printf("Neispravan unos.\n");
17         return -1;
18     }
19
20     /* 1. nacin */
21     printf("----- 1. nacin ----- \n");
22     for(i=2;i<x;i++)
23     {
24         /* Proverava se da li i deli broj x i

```

2 Kontrola toka

```
25     * ako je to slucaj ispusje se i.
26     */
27     if (x%i==0)
28         printf("%d \n",i);
29 }

31 /* 2. nacin (brzi) -- Ne proveravaju se svi brojevi od 2 do x, vec
32    * se petlja izvrsava dok ne stignemo do korena broja.
33    */
34 printf("----- 2. nacin ----- \n");
35 for(i=2;i<=sqrt(x);i++)
36 {
37     /* Proveravamo da li i deli broj x. */
38     if (x%i==0)
39         /* U slucaju kada je delilac koren broja, npr. 4 za 16,
40            ispisujemo ga jednom. */
41         if (i==x/i)
42             printf("%d \n",i);
43         /* U suprotnom, npr. 2 za 16, ispisujemo i 2 i 8. */
44         else
45             printf("%d %d \n",i,x/i);
46 }
47 return 0;
48 }
```

Rešenje 2.2.9

```
1 #include <stdio.h>

3 int main(){
4     int n;

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

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

19         /* Ispisujemo rezultat */
20         printf("%d\n", n);
21     }
22 }
23
```



```
25     return 0;
}
```

Rešenje 2.2.10

```
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 vrši za slučaj kada je unet
12      * negativan broj kako bismo osigurali da će nam izdvojene cifre
13      * biti pozitivne.
14      */
15     x = abs(x);
16
17     /* Kako uklanjamo cifre broja (pogledati telo petlje)
18      * u nekom trenutku broj će postati 0 jer smo uklonili
19      * sve njegove cifre. Tada prekidamo rad petlje.
20      */
21     while(x>0)
22     {
23         /* Izdvajamo poslednju cifru broja x. */
24         cifra=x%10;
25         printf("%d\\n", cifra);
26         /* Uklanjamo poslednju cifru broja x. */
27         x/=10;
28     }
29
30     return 0;
31 }
```

Rešenje 2.2.11

```
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;
```

2 Kontrola toka

```
11 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
17 * od 0
   */
19
21 if (n <= 0)
22 {
23     printf("Neispravan unos.\n");
24     return -1;
25 }

27 /* Potrebno je koristiti pomocnu promenljivu jer u telu petlje
   * se odstranjuju cifre broja i na taj nacin uneseni broj
   * se menja. Nakon rada petlje potrebno je ponovo koristiti
29 * uneseni broj, a to znaci da treba sacuvati neizmenjen
   * broj.
   */
31 pom_n = n;

33 while(pom_n != 0)
34 {
35     /* Na sumu dodajemo poslednju cifru. */
37     suma += pom_n % 10;
38     /* Sa broja skidamo poslednju cifru. */
39     pom_n /= 10;
40 }

41
43 if (n % suma == 0)
44     printf("Deljiv je sumom svojih cifara.\n");
45 else
46     printf("Nije deljiv sumom svojih cifara.\n");
47
48 return 0;
49 }
```

Rešenje 2.2.12

```
1 #include<stdio.h>

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;

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

15

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

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

29  printf("Unesite brojeve: ");
   while(i<n)
31  {
       scanf("%d", &x);

33

35       if (x<0)
           suma_neg+=x;
       else
37           suma_poz+=x;

39       i++;
   }

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

```

Rešenje 2.2.13

```

1  #include <stdio.h>

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

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

```

```
15  if (n < 0)
16  {
17      printf("Neispravan unos.\n");
18      return -1;
19  }
20
21  printf("Unesite n brojeva: ");
22
23  /* Inicijalizuje se broj sa kojim se kontrolise broj
24   * učitavanja - treba da ih bude tacno n.
25   */
26  i=0;
27  while(i<n){
28      /* Ucitava se broj. */
29      scanf("%d", &x);
30
31      /* Proverava se da li broj negativan i neparan. */
32      if(x<0 && x%2!=0){
33          /* Ako jeste, dodajemo ga na zbir. */
34          zbir=zbir+x;
35      }
36
37      /* Uvecava se broj iteracija. */
38      i++;
39  }
40
41  /* Ispisuje se rezultat. */
42  printf("%d\n", zbir);
43
44  return 0;
45 }
```

Rešenje 2.2.14

```
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     {
15         printf("Neispravan unos.\n");
16         return -1;
17     }
```

```
19 printf("Unesite brojeve: ");
20 for(i=0; i<n; i++)
21 {
22     scanf("%d", &broj);
23
24     if (broj % 5 == 0 && broj % 7 != 0)
25         suma += broj;
26 }
27
28 printf("Suma je %d.\n", suma);
29
30 return 0;
31 }
```

Rešenje 2.2.15

```
1 #include <stdio.h>
2
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;
10
11     printf("Unesite koliko novaca ima Nikola: ");
12     scanf("%f", &m);
13
14     if (m < 0)
15     {
16         printf("Nikola ne moze imati negativno novaca.\n");
17         return -1;
18     }
19
20     printf("Unesite broj artikala: ");
21     scanf("%d", &n);
22
23     if (n < 0)
24     {
25         printf("Broj artikala ne moze biti negativan.\n");
26         return -1;
27     }
28
29     printf("Unesite cene artikala: ");
30
31     i=0;
32     while(i<n){
33         /* Ucitava se cena artikla. */
34         scanf("%f", &cena);
```

2 Kontrola toka

```
35     if (cena <= 0)
36     {
37         printf("Cena ne moze biti negativna.\n");
38         return -1;
39     }
40
41     /* Provera da li je cena manji od zadatog broja m. */
42     if(cena < m){
43         /* Ako jeste, uvecava se brojac brojeva za 1. */
44         broj_brojeva++;
45     }
46
47     i++;
48 }
49
50 printf("%d\n", broj_brojeva);
51
52 return 0;
53 }
```

Rešenje 2.2.16

```
1  #include <stdio.h>
2  int main()
3  {
4      int x;
5      /* U promenljivoj p se cuva proizvod. */
6      int p;
7      /* Promenljiva u služi za proveru da li
8       * su brojevi uopšte uneseni. Na početku se
9       * pretpostavlja da nisu i postavlja se na 0.
10     */
11     int u = 0;
12
13     /* Promenljiva unesen_pozitivan služi za proveru da li
14     * su pozitivni brojevi uopšte uneseni. Na početku se
15     * pretpostavlja da nisu i postavlja se na 0.
16     */
17     int unesen_pozitivan = 0;
18
19     p=1;
20     /* Izraz 1 je konstantan, razlicit je od nule
21     * sto znaci da je to tacan izraz. Uslov petlje je
22     * uvek tacan!
23     */
24     printf("Unesite brojeve:");
25     while (1)
26     {
27         scanf("%d", &x);
28         /* Proveravanje da li je uneta nula. */
```

```

29     if (x==0)
        /* Naredba break prekida petlju. Izvršavanje
31         * se nastavlja od prve naredbe nakon petlje.
        */
33         break;

35     /* Ako je makar 1 broj razlicit od 0 promenljiva u
        * ce biti postavljena na 1.
37     */
    u = 1;
39     /* Ako je unet negativan broj, taj broj se ne mnozi
        * sa ukupnim proizvodom p; zato se nastavlja dalje.
41     */
    if (x<0)
43        /* Naredba continue prekida trenutnu iteraciju petlje
        * tako sto preskace sve naredbe koje nakon njega slede.
45        * Izvršavanje se nastavlja od provere uslova petlje.
        */
47        continue;

49     /* Ako je makar jedan broj pozitivan, promenljiva
        * unesen_pozitivan se postavlja na 1. */
51     unesen_pozitivan = 1;
    p=p*x;
53 }

55 if (u == 0)
    printf("Nisu uneseni brojevi.\n");
57 else if (unesen_pozitivan == 0)
    printf("Nisu uneseni pozitivni brojevi. \n");
59 else
    printf("Proizvod pozitivnih unetih brojevi je %d.\n",p);
61 return 0;
63 }

```

Rešenje 2.2.17

```

1  #include <stdio.h>
   #include <stdlib.h>
3
   int main(){
5       int n, cifra;
       int indikator=0;
7
       /* Ucitavamo broj. */
9       printf("Unesite broj: ");
       scanf("%d", &n);
11
       if (n < 0)
13         n = abs(n);

```

```
15  /* Sve dok imamo cifara u zapisu broja. */
16  while(n>0){
17
18      /* Izdvajamo poslednju cifru broja. */
19      cifra=n%10;
20
21      /* Proveravamo da li je bas ona jednaka broju 5 */
22      if(cifra==5){
23          /* Ako jeste postavljamo indikator na vrednost 1 tako da
24             znamo
25             * da smo pronasli peticu i prekidamo sa izvršavanjem petlje.
26             */
27             indikator=1;
28             break;
29         }
30         /* Ako izdvojena cifra nije jednaka broju 5, broj delimo sa 10
31            * kako bi mogli da izdvojimo i preostale cifre broja na isti
32            * nacin.
33            */
34         n=n/10;
35     }
36
37     /* Ispisujemo rezultat */
38     if(indikator==0){
39         printf("Cifra 5 se ne nalazi u zapisu!\n");
40     }
41     else{
42         printf("Cifra 5 se nalazi u zapisu!\n");
43     }
44
45     return 0;
46 }
```

Rešenje 2.2.18

```
1  #include <stdio.h>
2  #include <math.h>
3
4
5  int main(){
6
7      int x;
8      int broj_brojeva;
9      int suma;
10
11     broj_brojeva=0;
12     suma=0;
13
14     printf("Unesite brojeve: ");
15 }
```



```

17 while(1){
    /* Ucitavanje broja. */
    scanf("%d", &x);

19
    /* Ako je unesena 0, prekida se petlja. */
21    if(x==0)
        break;

23

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

27
    /* I uvecava se broj ucitanih brojeva. */
29    broj_brojeva++;
}

31
33 if (broj_brojeva == 0)
    printf("Nisu uneseni brojevi.\n");
else
35    /* Prilikom deljenja celih brojeva kao rezultat se dobija ceo
        broj.
        * Kako je aritmeticka sredina realan broj, potrebno je izvršiti
37    * konverziju prilikom deljenja da bi se dobio ispravan rezultat.
        */
39    printf("Aritmeticka sredina: %.4f\n", (double)suma/broj_brojeva);

41    return 0;
}

```

Rešenje 2.2.19

```

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

3

5  int main(){

   float cena;
7   int broj_artikla;
   float suma;

9

   broj_artikla=0;
11  suma=0;

13

   printf("Unesite cene: ");

15

   while(1){
17       scanf("%f", &cena);

19       if(cena == 0)
           break;

```

```
21     if (cena < 0)
22     {
23         printf("Cena ne moze biti negativna.\n");
24         return -1;
25     }
26
27     suma+=cena;
28
29     /* I uvecava se broj ucitanih brojeva. */
30     broj_artikla++;
31 }
32
33 if (broj_artikla == 0)
34     printf("Nisu unesene cene.\n");
35 else
36     printf("Aritmeticka sredina: %.4f\n", suma/broj_brojeva);
37
38 return 0;
39 }
```

Rešenje 2.2.20

```
1  #include <stdio.h>
2
3  int main()
4  {
5      int n;
6      /* Ucitavaju se dva broja, broj i sledbenik, i proverava se da li
7       * razlicitog znaka. */
8      double broj, sledbenik;
9      /* Brojac. */
10     int i;
11     int broj_promena = 0;
12
13     printf("Unesite broj n ");
14     scanf("%d", &n);
15
16     if (n < 0)
17     {
18         printf("Neispravan unos.\n");
19         return -1;
20     }
21
22     /* Prvo se proveara da li uopste ima unosa, i ako
23      * unosa nema, ispisuje se odgovarajuca poruka i
24      * izlazi iz programa. */
25     if (n == 0)
26     {
27         printf("Broj promena je 0.\n");
28     }
```

```

    return 0;
29 }

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

37 /* Kako je vec jedan broj unesen, brojac se postavlja
   * na 1, a ne na 0. */
39 for(i=1; i<n; i++)
{
41     /* Učitava se sledbenik. */
    scanf("%lf", &sledbenik);
43
45     /* Ako su razlicitog znaka proizvod je manji od 0. */
    if (sledbenik * broj < 0)
        broj_promena++;
47     /* Problem je ako je proizvod jednak 0. Tada mora
       * proveriti da li je jedan od brojeva negativan jer tada
49     * postoji promena znaka. */
    else if (sledbenik*broj==0 && (sledbenik < 0 || broj < 0))
51         broj_promena++;

53     /* Tekuci sledbenik postaje tekuci broj, a
       * u sledecoj iteraciji petlje se učitava
55     * sledeci sledbenik. */
    broj = sledbenik;
57 }

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

61 return 0;
}

```

Rešenje 2.2.21

```

1 #include <stdio.h>
int main()
3 {
   /* Broj artikala. */
5   int n;
   /* Brojac. */
7   int i;
   /* Cena trenutno unetnog artikla. */
9   float cena;
   /* Minimalna cena. */
11  float min_cena;

13  printf("Unesite broj artikala:");

```

```
scanf("%d", &n);
15
if (n<=0)
17
{
    printf("Neispravan unos\n");
19
    return -1;
}
21

/* Prva cena se unosi iznad petlje
 * kako bi bio njegova vrednost bila dodeljena promenljivoj
min_cena.
23
 * Neophodno je da promenljiva min bude inicijalizovana pre
ulaska u petlju
25
 * da bi uslov x<min mogao da bude ispitan u prvoj iteraciji.
 */
27
printf("Unesite cenu artikala:");
scanf("%f", &cena);
29
/* Proveravamo da li je cena isprano uneta vrednost. */
if (cena <= 0)
31
{
33
printf("Cena ne moze biti negativna.\n");
return -1;
35
}

min_cena = cena;
37

i=0;
39
while(i< n-1)
41
{
43
    scanf("%f", &cena);

    if (cena <= 0)
45
    {
47
printf("Cena ne moze biti negativna.\n");
return -1;
49
    }

    /* Provera da li je uneta cena manja od tekuće minimalne cene.
    */
51
    if(cena<min_cena)
53
        min_cena = cena;
        i++;
55
    }

57
printf("Minimalna cena je: %f\n", min_cena);
return 0;
59
}
```

Rešenje 2.2.22

```
1  #include <stdio.h>
2  #include <stdlib.h>
3
4  int main(){
5
6      int n;
7      int x, x_desetica;
8      int max_desetica, broj;
9      int i;
10
11     printf("Unesite broj n: ");
12     scanf("%d", &n);
13
14     if (n < 0)
15     {
16         printf("Neispravan unos.\n");
17         return -1;
18     }
19
20     if (n == 0)
21     {
22         printf("Nisu uneseni brojevi.\n");
23         return 0;
24     }
25
26     /* Maksimalna cifra desetice se postavlja na 0 jer
27      * 0 je svakako najmanja cifra pa je pocetna vrednost
28      * neutralna tj.
29      * ne moze da utice na izracunavanje maksimuma. Ipak, treba
30      * biti pazljiv
31      * jer nije uvek zgodno pretpostaviti da je maksimalna
32      * vrednost 0.
33      * Na primer, ako je zadatak naci maksimum celih brojeva, a
34      * korisnik
35      * unese -32 -7 i -22, maksimalni je broj -7, sto je manje od
36      * 0.
37      */
38     max_desetica=0;
39
40     printf("Unesite brojeve: ");
41     for(i=0; i<n; i++){
42         scanf("%d", &x);
43
44         /* Izdvajanje cifre desetice procitanog broja. */
45         x_desetica=(abs(x)/10)%10;
46
47         /* Proverava da li je izdvojena cifra veca od
48          * trenutne maksimalne cifre desetica. */
49         if(x_desetica>max_desetica)
50         {
51             /* Ako jeste vece, pamti se nova najveca cifra,
```

2 Kontrola toka

```
47         * kao i broj u kom se pojavila. */
        max_desetica=x_desetica;
49     broj=x;
    }
51 }

53 printf("Broj sa najvecom cifrom desetica je %d\n", broj);
55 return 0;
57 }
```

Rešenje 2.2.23

```
1  #include <stdio.h>
2  #include <stdlib.h>
3
4  int main(){
5
6      int n;
7      int x, x_kopija, br_cifara;
8      int max_br_cifara, broj;
9      int i;
10
11     printf("Unesite broj n: ");
12     scanf("%d", &n);
13
14     if (n < 0)
15     {
16         printf("Neispravan unos.\n");
17         return -1;
18     }
19
20     if (n == 0)
21     {
22         printf("Nisu uneseni brojevi.\n");
23         return 0;
24     }
25
26     /* Maksimalan broj cifara se postavlja na 0,
27        * svaki broj ima vise od 0 cifara pa je ova vrednost
28        * neutralna. */
29     max_br_cifara=0;
30
31     printf("Unesite n brojeva: ");
32     for(i=0; i<n; i++){
33         scanf("%d", &x);
34
35         /* Odredjivanje broja cifara unetog broja x. */
36         x_kopija=abs(x);
37         br_cifara=0;
```

```

39     while(x_kopija!=0){
        x_kopija=x_kopija/10;
        br_cifara++;
41     }

43     /* Ako je broj cifara unetog broja veci od maksimalnog */
    if(br_cifara>max_br_cifara){
45         /* Cuvamo ga */
        max_br_cifara=br_cifara;
47         /* I zbog ispisa rezultata, cuvamo i originalni broj */
        /* Zbog ovoga smo morali i da racunamo broj cifara nad kopijom
        broja x kako ne bismo promenili njegovu vrednost */
49         broj=x;
        }
51     }

53     printf("Najvise cifara ima broj %d\n", broj);
55     return 0;
57 }

```

Rešenje 2.2.24

```

/* Sa standardnog ulaza se unosi ceo pozitivan broj n, a zatim i n
   celih brojeva. Napisati program koji ispisuje
2 broj sa najvecom vodecom cifrom. Vodeca cifra je prva cifra iz zapisa
   broja. Ukoliko ima vise takvih, ispisati
   prvi. */
4
#include <stdio.h>
6 #include <math.h>

8 int main(){
10     int n;
    int x, x_kopija;
12     int broj;
    int vodeca_cifra, max_vodeca_cifra;
14     int i;

16     /* Citamo vrednost sa ulaza */
    printf("Unesite broj n: ");
18     scanf("%d", &n);

20     /* Postavljamo maksimalnu vodecu cifru na 0 - cifre broja su vece
       ili jednake od 0 pa je ova vrednost neutralna */
    max_vodeca_cifra=0;
22
    /* Ucitavamo broj po broj */

```

2 Kontrola toka

```
24 printf("Unesite n brojeva: ");
25 for(i=0; i<n; i++){
26     scanf("%d", &x);

27     /* Odredjujemo vodecu cifru broja */
28     x_kopija=abs(x);
29     while(x_kopija>10){
30         x_kopija=x_kopija/10;
31     }
32     vodeca_cifra=x_kopija;

33     /* Ako je izdvojena cifra veca od maksimalne vodece cifre */
34     if(vodeca_cifra>max_vodeca_cifra){
35         /* Cuvamo je */
36         max_vodeca_cifra=vodeca_cifra;
37         /* I zbog ispisa, cuvamo i broj u kojem se ona pojavljuje */
38         /* Zbog ovoga smo morali i da racunamo vodecu cifru nad kopijom
39         broja x kako ne bismo promenili njegovu vrednost */
40         broj=x;
41     }
42 }
43
44 /* Ispisujemo rezultat */
45 printf("%d\n", broj);
46
47 return 0;
48 }
49
50 }
```

Rešenje 2.2.25

```
1 #include <stdio.h>
2 #include <math.h>
3
4 int main(){
5
6     int x;
7     int min, max;

8
9     printf("Unesite brojeve: ");

10
11     /* Prvi broj se ucitava izvan petlje zbog
12     * inicijalizacije maksimuma i minimuma. */
13     scanf("%d", &x);
14     max=x;
15     min=x;

16
17     /* Sve dok se ne unese 0, ucitavaju se brojevi u petlji. */
18     while(x!=0){
19
20     }
```



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

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

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

Rešenje 2.2.26

```
1  #include <stdio.h>
    #include <stdlib.h>
3
    int main(){
5
        int n;
        int d;
            /* Uzastopni brojevi za koje se racuna rastojanje. */
9        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)
            {
19             printf("Neispravan unos.\n");
                return -1;
21         }

23       broj_parova=0;

25       printf("Unesite n brojeva: ");

27       /* Prvi broj se ucitava pre petlje. */
        scanf("%d", &x);
29
        for(i=1; i<n; i++){
```

2 Kontrola toka

```
31     scanf("%d", &y);

33     /* Provera da li su brojevi na rastojanju d. */
    if(abs(y-x)==d)
35         broj_parova++;

37     /* Broj iz tekuće iteracije se čuva kako bi
                                   * mogao da se upotrebljava u narednoj iteraciji. */
39     x=y;
    }

41     printf("Broj parova: %d\n", broj_parova);

43     return 0;

45 }
```

Rešenje 2.2.27

```
1  #include <stdio.h>

3  int main()
{
5     int x;
    /* Tezina trenutne pozicije u broju. Može biti 1, 10, 100,
7     * 1000 itd.
    */
9     int pozicija;
    /* Trenutna izdvojena cifra iz broja x. */
11    int cifra;
    /* Broj dobijen nakon transformacije. */
13    unsigned int y;

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

17

19    if(x <= 0)
    {
21        printf("Nekorektan unos.\n");
        return -1;
23    }

25    /* Posto pocinjemo sa izdvajanjem cifara od cifre jedinica,
        * postavlja se tezinu (stepen) pozicije na 1.
    */
27    pozicija = 1;
29    y = 0;

31    /* Provera da li ima cifara u zapisu broja. */
    while(x > 0) {
```

```

33      /* Izdvaja se poslednja cifra iz zapisa. */
35      cifra = x % 10;

37      /* Provera da li je cifra parna. */
39      if(cifra % 2 == 0){
41          /* I ako jeste, uvecava se. */
43          cifra++;
45      }

47      /* Novi broj se formira tako sto se izdvojena cifra pomnozi
      * odgovarajucom tezinom (stepenom) pozicije.
      */
49      y += cifra*pozicija;

51      /* Priprema se broj za izdvajanje naredne cifre, uklanja se
      * poslednja cifra broja.
      */
53      x /= 10;

55      /* Uvecava se tezinu (stepen) pozicije. */
57      pozicija *= 10;
59      printf("%d\n", y);

61      return 0;
63  }

```

Rešenje 2.2.28

```

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

5  int main() {

7      int x;
9      /* Tezina trenutne pozicije u broju. Moze biti 1, 10, 100,
      * 1000 itd.
      */
11     int stepen_deset;
13     /* Trenutna izdvojena cifra iz broja x. */
15     int cifra;
17     /* Redni broj cifre koja se trenutno obradjuje, gledano s desna
      * na levo.
      */
19     int rbr;

```

```
19  /* Broj dobijen nakon transformacije. */
20  int y;
21  /* Promenljiva znak cuva znak unesenog broja. Moze biti
22   * -1 za negativnu vrednost ili 1 za pozitivnu vrednost.
23   */
24  int znak = 1;
25
26  /* Ucitavanje broja. */
27  printf("Unesite broj: ");
28  scanf("%d", &x);
29
30  if(x <= 0)
31  {
32      x = abs(x);
33      znak = -1;
34  }
35
36  /* Postavlja se vrednost stepena na 0 - to znaci da se prvo mnozi
37   sa
38   * 10^0=1.
39   */
40  stepen_deset = 0;
41
42  /* Postavlja se vrednost broja koji se formira na 0. */
43  y = 0;
44  /* Postavlja se redni broj pozicije na 0. */
45  rbr = 0;
46
47  /* Provera da li ima cifara u zapisu broja. */
48  while(x > 0) {
49
50      /* Izdvajanje cifre. */
51      cifra = x%10;
52
53      /* Proverava se da li je pozicija izdvojene cifre parna -
54       * cifre na parnim pozicijama se zadržavaju.
55       */
56      if(rbr % 2 == 0) {
57          /* Ako jeste parna
58           * izdvojena cifra se dodaje novom broju.
59           * Neophodno je izvršiti promenu tipova, jer je double
60           povratni
61           * tip funkcije pow.
62           */
63          y += cifra * ((int) pow(10, stepen_deset));
64
65          /* Uvecava se stepen zbog naredne cifre. */
66          stepen_deset++;
67      }
68
69      /* Azurira se redni broj cifre. */
70      rbr++;
71      /* I priprema se broj za naredno izdvajanje. */
72  }
```

```
        x /= 10;
69    }

71    y = znak*y;

73    /* Ispisuje se rezultat. */
    printf("%d\n", y);
75

77    return 0;
}
```

Rešenje 2.2.29

```
1  #include <stdio.h>

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

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

11

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

17

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

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

23

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

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

37        /* Azurira se vrednost stepena. */
```

```

    stepen=stepen*10;
39
    /* Provera da li je srednja cifra jednaka zbiru leve i
41     * desne cifre.
    */
43     if(cifra_levo+cifra_desno==cifra_sredina){
45         /* Treba izbaciti srednju cifru, pa broj n se azurira
            * tako sto se podeli sa 100.
            */
47         n=n/100;
49     }
    else{
51         /* Inace, zadrzava se srednja cifra i odbacuje
53         * se samo poslednja.
            */
55         n=n/10;
    }
57 }

59 /* Na novi broj se dodaje preostali dvocifreni
    * ili jednocifreni broj.
    */
61 novo_n=n*stepen+novo_n;
63
65 /* Ispisivanje rezultata. */
    printf("%d\n", novo_n);
67
69     return 0;
}

```

Rešenje 2.2.30

```

1  #include <stdio.h>
   #include <math.h>
3
   int main() {
5
       int x;
7       int broj_cifara;
       int min_stepen, max_stepen;
9       int pom;
       int leva_cifra, desna_cifra;
11      int indikator;

13      printf("Unesite broj: ");
       scanf("%d", &x);
15

```

```
17  /* Ako je korisnik uneo negativan broj, analizira se njegova
    * apsolutna vrednost.
    */
19  if(x < 0)
    x=-x;
21
23  /* Odredjuje se broj cifara u zapisu broja x da bi moglo da
    * se izdvajaju istovremeno cifre i sa leve i sa desne strane.
    */
25  broj_cifara = 0;
27  pom = x;
    while(pom > 0) {
29      pom /= 10;
        broj_cifara++;
31  }
33  /* Odredjuje se stepen koji stoji uz krajnju levu cifru broja. */
    max_stepen = (int) pow (10, broj_cifara-1);
35
37  /* Indikator je promenljiva koja ukazuje da li je broj
    * palindrom ili ne.
    */
39  indikator=1;
    while(x!=0 && indikator==1){
41      /* Izdvaja se leva cifra. */
        leva_cifra=x/max_stepen;
43      /* Izdvaja se desna cifra. */
        desna_cifra=x%10;
45      /* Ako su cifre razlicite, odmah se moze zakljuciti da
        * broj nije palindrom i prekida se izvršavanje petlje.
        */
47      if(leva_cifra!=desna_cifra){
49          indikator=0;
            break;
51      }
        /* Formira se nova vrednost broja x tako sto se odbacuje
53      * krajnja leva i krajnja desna cifra.
        */
55      x=(x/max_stepen-x%10)/10;
        /* Koriguje se maksimalan stepen tako dobijenog broja -
57      * deli se sa 100 jer su odbacene 2 cifre.
        */
59      max_stepen=max_stepen/100;
    }
61
63  /* Ispisuje se rezultat. */
    if(indikator==1)
        printf("Broj je palindrom!\n");
65  else
        printf("Broj nije palindrom!\n");
67
```

```
69     return 0;
    }
```

Rešenje 2.2.31

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



```
39     i++;
40 }
41
42 printf("Trazeni broj je: %d\n", drugi);
43
44 return 0;
45 }
```

Rešenje 2.2.32

```
#include<stdio.h>
2 int main()
3 {
4     int a0;
5     int an,an1;
6
7     printf("Unesi pocetni clan niza brojeva:");
8     scanf("%d",&a0);
9
10    if (a0 <= 0)
11    {
12        printf("Nekorektan unos. Broj mora biti pozitivan.\n");
13        return -1;
14    }
15
16    printf("%d\n", a0);
17
18    an=a0;
19    while(an!=1)
20    {
21        if (an%2) /* Ukoliko je vrednost izraza an%2 razlicita od nule,*/
22        {          /* izraz se tumaci kao tacan i izvrsavaju se naredbe iz
23            if grane. */
24                an1=(3*an+1)/2;
25            }
26        else /* U suprotnom, ukoliko je vrednost izraza an%2 jednaka nuli
27            , izraz */
28        {          /* se tumaci kao netacan i izvrsavaju se naredbe iz else
29            grane. */
30                an1=an/2;
31            }
32        printf("%d\n",an1);
33        an=an1;
34    }
35
36    return 0;
37 }
```

Rešenje 2.2.33

```
1 #include <stdio.h>
2 #include <math.h>
3
4 int main()
5 {
6     int format;
7     /* Pomocna promenljiva koja služi kao brojac u petlji. */
8     int i;
9     /* Trenutne vrednosti za sirinu i visinu i
10      * pomocna promenljiva za promene u petlji.
11      */
12     double sirina, duzina, nova_duzina;
13     unsigned int konacna_sirina, konacna_duzina;
14
15     printf("Uneti format papira: ");
16     scanf("%d", &format);
17
18     if (format <= 0)
19     {
20         printf("Neispravan unos.\n");
21         return -1;
22     }
23
24     /* duzina/sirina = 1 : sqrt(2) duzina*sirina = 1000x1000mm^2
25      * Na osnovu ovih odnosa dobijamo pocetnu vrednost za sirinu
26      * i duzinu, odnosno vrednosti za papir A0.
27      */
28     duzina = sqrt(1000*1000/sqrt(2));
29     sirina = sqrt(2)*duzina;
30
31     /* Kako vec imamo odredjenu sirinu i duzinu za papir A0,
32      * petlju krecemo od izracunavanja za papir A1, pa
33      * brojac i postavljamo na 1.
34      */
35     for(i=1; i<=format; i++)
36     {
37         nova_duzina = sirina/2;
38         sirina = duzina;
39         duzina = nova_duzina;
40     }
41
42     /* Duzina i sirina celi brojevi. */
43     konacna_sirina = (unsigned int)sirina;
44     konacna_duzina = (unsigned int)duzina;
45
46     printf("%u %u\n", konacna_duzina, konacna_sirina);
47
48     return 0;
49 }
```

Rešenje 2.2.34

```

1  #include <stdio.h>
3  int main()
4  {
5      char c;

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

11     while((c=getchar())!='.')
12     {
13         if (c>='A' && c<='Z')
14         /* Razlika izmedju ascii koda svakog malog i odgovarajuceg
15          * velikog slova je konstanta koja se moze racunati izrazom
16          * 'a'-'A' (i iznosi 32). */
17             putchar(c+'a'-'A');
18         else if (c>='a' && c<='z')
19             putchar(c-'a'+'A');
20         else
21             putchar(c);
22     }

23     return 0;
24 }

```

Rešenje 2.2.35

```

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

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

14
15     /* Petlja se zavrsava kada korisnik ne unese karakter,
16      * vec zada konstantu EOF .
17      * Ova konstanta se zadaje kombinacijom tastera CTRL+D.
18      * U tom slucaju, getchar() vraca -1. */
19     while((c=getchar())!=EOF)

```

2 Kontrola toka

```
20 {
21     if (c>='A' && c<='Z')
22         br_v++;
23     else if (c>='a' && c<='z')
24         br_m++;
25     else if (c>='0' && c<='9')
26     {
27         br_c++;
28         /* Kada od promenljive tipa char oduzimamo karakter
29          * (ili neku drugu promenljivu tipa char), zapravo se
30          * vrsi oduzimanje njihovih ascii vrednosti i dobija se
31          * broj. */
32         suma=suma+c-'0';
33     }
34     else if (c=='\t' || c=='\n' || c==' ')
35         br_b++;
36
37     br_k++;
38 }
39
40 printf("velika: %d, mala: %d, cifre: %d, beline: %d \n", br_v, br_m
41       , br_c, br_b);
42 printf("suma cifara: %d\n", suma);
43
44 return 0;
45 }
```

Rešenje 2.2.36

```
#include <stdio.h>
2
3 int main()
4 {
5     /* Promenljiva i je brojac. */
6     int n, i;
7     /* Brojaci za svaki od samoglasnika. */
8     int broj_a = 0, broj_e = 0, broj_i = 0, broj_o = 0, broj_u = 0;
9     /* Promenljiva c je tekuci ucitani karakter. */
10    char c, belina;
11
12    printf("Unesite broj n: ");
13    scanf("%d", &n);
14
15    if (n < 0)
16    {
17        printf("Neispravan unos.\n");
18        return -1;
19    }
20
21    for(i=0; i<n; i++)
22    {
```

```

24      /* Prvo se cita belina koja se nalazi nakon prethodnog
      * unosa, pa tek posle procitane beline se cita uneseni
      * karakter. */
26      scanf("%c%c", &belina, &c);

28      /* Provera da li je ucitani karakter samoglasnik. */
      switch(c)
30      {
          case 'a': case 'A': broj_a++; break;
          case 'e': case 'E': broj_e++; break;
          case 'i': case 'I': broj_i++; break;
          case 'o': case 'O': broj_o++; break;
          case 'u': case 'U': broj_u++; break;
36      }
38      }

      printf("samoglasnik a: %d\n", broj_a);
      printf("samoglasnik e: %d\n", broj_e);
      printf("samoglasnik i: %d\n", broj_i);
      printf("samoglasnik o: %d\n", broj_o);
      printf("samoglasnik u: %d\n", broj_u);
44      return 0;
46  }

```

Rešenje 2.2.37

```

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

3

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

7  int main(){

9      int n;
      int broj_Z, broj_i, broj_m, broj_a;
11     char novi_red, c;
      int i;

13

      broj_Z=0;
      broj_i=0;
      broj_m=0;
      broj_a=0;

15

17

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

23     /* Ucitavanje karakter po karakter. */

```

```
for(i=0; i<n; i++){
25     printf("Unestite %d. karakter: ", i+1);
    /* Prvo se cita belina koja se nalazi nakon prethodnog
27         * unosa, pa tek posle procitane beline se cita
        uneseni
        * karakter. */
29     scanf("%c%c", &novi_red, &c);

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

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

57    return 0;
59 }
```

Rešenje 2.2.38

```
1  #include <stdio.h>

3  int main()
    {
5      int n;
        /* Brojac. */
        int i;
7      /* Promenljiva u kojoj se cuva suma kubova. */
        int s;
9
11 
```

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

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

21 i=1;
s=0;

23 for(i=1;i<=n;i++)
25     s+=i*i*i;

27 printf("Suma kubova od 1 do %d: %d\n", n, s);
29 return 0;
}
```

Rešenje 2.2.39

```
#include <stdio.h>

2 int main()
4 {
    int n;
    /* Brojac. */
    int i;
    /* Promenljiva u kojoj se cuva suma kubova. */
    int s;

10

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

14

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

20

22 i=1;
s=0;

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

2 Kontrola toka

```
30     return 0;
    }
```

Rešenje 2.2.40

```
1  #include <stdio.h>
3  int main(){
    int n, i;
5   float x, S, stepen;

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

9   if (n < 0)
11  {
        printf("Neispravan unos.\n");
13     return -1;
    }

15     /* Inicijalizacija sume. */
17     S=0;

19     /* Stepen promenljiva ce sadrzati vrednosti stepena x^n.
        * Pocetna vrednost joj je 1 jer je x^0 = 1. */
21     stepen=1;

23     for(i=1; i<=n; i++)
    {
25         stepen=stepen*x;
        S=S+i*stepen;
27     }

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

31     return 0;
}
```

Rešenje 2.2.41

```
1  #include <stdio.h>
2  int main(){
3   unsigned n, i;
    float x, S, stepen;

5   printf("Unesite redom brojeve x i n: ");
7   scanf("%f %u", &x, &n);

9   S=1;
```



```

    stepen=1;
11  for(i=1; i<=n; i++)
    {
13      stepen=stepen*x;
      S=S+1/stepen;
15  }

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

19  return 0;
}

```

Rešenje 2.2.42

```

#include <stdio.h>
2  #include <math.h>
int main(){
4      /* Promenljiva i je brojac, promenljiva S cuva izracunatu sumu,
      * a promenljiva clan je tekuci clan niza. */
6      int i;
      float S;
8      float x, eps;
      float clan;

10     printf("Unesite x: ");
12     scanf("%f", &x);

14     printf("Unesite tacnost eps: ");
16     scanf("%f", &eps);

18     S=0;
      /* Prvi clan sume je 1. */
      clan = 1;
20     i=1;
      while(clan > eps){
22         S=S+ clan;
          clan = clan*x / i;
24         i++;
      }

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

28     return 0;
30 }

```

Rešenje 2.2.43

```

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

```

```
int main(){
4      /* Promenljiva i je brojac, promenljiva S cuva izracunatu
      sumu,
      * promenljiva znak moze bito 1 ili -1 i odredjuje znak
6      * trenutnog clana sume, a promenljiva clan je tekuci clan
      niza. */
      int i, znak;
8      float S;
      float x, eps, clan;

10     printf("Unesite x: ");
12     scanf("%f", &x);

14     printf("Unesite tacnost eps: ");
16     scanf("%f", &eps);

18     S=0;
      clan = 1;
20     i=1;
      znak=1;

22     /* Kako clanovi sume mogu biti negativni, potrebno je
      posmatrati
24     * apsolutnu vrednost clana. */
      while(fabs(clan)>eps)
26     {
          S = S + clan;

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

          i++;
34     }

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

38     return 0;
40 }
```

Rešenje 2.2.44

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

```
double stepen = 1;
double proizvod = 1;

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

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

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

printf("Proizvod = %lf\n", proizvod);

return 0;
}
```

Rešenje 2.2.45

```
#include <stdio.h>

int main()
{
    int n, i;
    double Razlomak;

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

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

    Razlomak = n;

    /* Razlomak se izracunava "od nazad", odnosno, krece se
     * od najnizeg razlomka 1/n i od njega se nadalje formira
     * sledeci, "visi" razlomak itd. Završava se kada se stigne
     * do koraka 0 + 1/R. */
    for(i=n-1; i>= 0; i--)
        Razlomak = i + 1/Razlomak;

    printf("Razlomak = %lf\n", Razlomak);
}
```

```
27     return 0;
29 }
```

Rešenje 2.2.46

```
1  #include <stdio.h>
2  #include <math.h>
3  int main(){
4      /* Promenljiva i je brojac, promenljiva S cuva izracunatu
5       sumu,
6       * promenljiva znak moze bito 1 ili -1 i odredjuje znak
7       * trenutnog clana sume, a promenljiva clan je tekuci clan
8       niza. */
9      int i, znak, n;
10     float S;
11     float x, clan;
12
13     printf("Unesite x i n: ");
14     scanf("%f%d", &x, &n);
15
16     if (n <= 0)
17     {
18         printf("Neispravan unos.\n");
19         return -1;
20     }
21
22     S=0;
23     clan = 1;
24     i=1;
25     znak=1;
26
27     /* Kako clanovi sume mogu biti negativni, potrebno je
28     posmatrati
29     * apsolutnu vrednost clana. */
30     while(i <= 2*n)
31     {
32         S = S + clan;
33
34         /* Promena znaka. */
35         znak = - znak;
36         /* Svaki clan suma se od prethodnog clana razlikuje za x
37         ^2/(i*(i+1)). */
38         clan = clan*x*x / (i*(i+1));
39         clan *= znak;
40
41         i+=2;
42     }
43
44     printf("S=%f\n", S);
```

```
43     return 0;
}
```

Rešenje 2.2.47

```
1  #include <stdio.h>
   #include <math.h>
3
   int main()
5  {
       int n, i;
       /* Promenljiva clan je deo proizvoda i predstavlja 1/i!. */
       double clan;
       double S = 1;
11
       printf("Unesite prirodan broj: ");
       scanf("%d", &n);
13
       if (n <= 1)
15     {
           printf("Neispravan unos.\n");
           return -1;
17     }
19
       clan = 1;
       for(i=2; i<=n; i++)
21     {
           clan = clan/i;
           S *= 1 + clan;
23     }
25
       printf("S = %lf\n", S);
27
       return 0;
29 }
}
```

Rešenje 2.2.48

```
1  #include <stdio.h>
   #include <math.h>
3
   int main()
5  {
       int n, i, znak = -1;
       /* Promenljiva clan je deo proizvoda i predstavlja 1*3*5*...*i.
          */
       long int clan;
       long int S = 0;
9
```

```
11     printf("Unesite prirodan broj: ");
12     scanf("%d", &n);
13
14     if (n < 5 || n % 2 == 0)
15     {
16         printf("Neispravan unos.\n");
17         return -1;
18     }
19
20     clan = 1*3;
21     for(i=5; i<=n; i+=2)
22     {
23         clan = znak*clan*i;
24         S += clan;
25     }
26
27     printf("S = %ld\n", S);
28
29     return 0;
30 }
```

Rešenje 2.2.49

```
1  #include <stdio.h>
2  #include <math.h>
3
4  int main()
5  {
6      int n, i;
7      double P;
8      double x, a;
9
10     printf("Unesite dva realna broja x i a: ");
11     scanf("%lf%lf", &x, &a);
12
13     printf("Unesite prirodan broj: ");
14     scanf("%d", &n);
15
16     if (n <= 0)
17     {
18         printf("Neispravan unos.\n");
19         return -1;
20     }
21
22     P = x;
23     for(i=0; i<n; i++)
24         P = (P+a)*(P+a);
25
26     printf("Izraz = %lf\n", P);
27 }
```

```
28     return 0;
    }
```

Rešenje 2.3.1

```
#include <stdio.h>

2  int kvadrat(int x)
4  {
    /* promenljive u listi argumenata funkcije, kao i one
6     deklarisanе u samoj funkciji, lokalne su za tu funkciju
       sto znaci da se promenljive x i y neće "videti" nigde izvan
8     funkcije kvadrat (ni u funkciji main ni u funkciji kub)
       */

10     int y;
12     y = x*x;
14     return y;
}

16 int kub(int a)
18 {
    /*
       u listi argumenata funkcije mozemo, a ne moramo, imati
       promenljivu
20     istog naziva kao promenljiva koja je deklarisanа u main
       funkciji
       (u ovom slucaju promenljiva a); ova promenljiva se razlikuje
22     od promenljive a deklarisanе u main funkciji i vidljiva je
       samo unutar funkcije kub
24     */
26     return a*a*a;
}

28 int main()
30 {
    int a, kv, kb;
    printf("Unesi ceo broj:");
32     scanf("%d", &a);

34     kv = kvadrat(a); /* promenljivoj kv dodeljujemo povratnu vrednost
       funkcije kvadrat */
    kb = kub(a);      /* promenljivoj kb dodeljujemo povratnu vrednost
       funkcije kub */

36     printf("Kvadrat broja %d je %d, a njegov kub je %d\n", a, kv, kb);
38     return 0;
}
```

Rešenje 2.3.2

```
1  /*
   Napisati program koji za uneti realan broj x i ceo broj n ispisuje
3  vrednost stepena x^n. Unosenje promenljivih, racunanje stepena i
   ispis promenljivih realizovati u posebnim funkcijama.
5  */

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

9
11 float stepen(float a, int b)
   {
13     float s=1;
       int i;

15     for(i=0;i<abs(b);i++)
         s=s*a;

17     return b>0 ? s : 1/s; /* ukoliko je izlozilac b negativan,
       izracunamo a^|b| i vracamo reciprocnu vrednost
19                                izracunatog stepena */

21 }

23 int main()
   {
25     int n;
       float x;
27     float s;

29
31     printf("Unesi jedan realan i jedan ceo broj:");
       scanf("%f%d",&x,&n);

33     s = stepen(x,n);

35
37     printf("%f~d=%f\n",x,n,s);

39     return 0;
   }
```

Rešenje 2.3.3

```
1  /*
   Napisati funkciju koja za dva data cela broja odredjuje
3  najveći zajednički delilac. Napisati potom glavni program
   koji testira ovu funkciju.
5  */
```



```
7 #include <stdio.h>

9 int euklid(int x, int y)
10 {
11     int r;
12     /* Euklidov algoritam */
13     while(y) /* algoritam se zaustavlja kada vrednost */
14     {       /* promenljive y postane nula */
15         r=x%y;
16         x=y;
17         y=r;
18     }
19     return x; /* nzd je sacuvan u promenljivoj x */
20 }

23 int main()
24 {
25     int a,b;
26     int nzd;

27     printf("unesi dva cela broja:");
28     scanf("%d%d", &a,&b);

31     nzd = euklid(a,b); /* promenljivoj nzd dodeljujemo povratnu
        vrednost funkcije euklid */

33     printf("najveci zajednicki delilac za %d i %d je %d\n", a,b,nzd);

35     return 0;
36 }
```

Rešenje 2.3.4

```
2 /*
   Napisati funkciju koja za dato n vraca zbir reciprocnih vrednosti
   brojeva od 1 do n.
   Napisati program koji omogucava korisniku da unese prirodan broj n, a
   potom ispisuje zbir reciprocnih
4 vrednosti brojeva od 1 do n koristeći funkciju float zbir_reciprocnih
   (int n). Rezultat zaokruziti
   na dve decimalne.
6 */

8 #include <stdio.h>

10 float zbir_reciprocnih(int n)
11 {
12     float z=0;
13     int i;
```

```
14  for(i=1;i<=n;i++)
    z+=1.0/i; /* da bismo dobili reciprocnu vrednost broja, vazno je
              da izbegnemo celobrojno deljenje dva cela broja */
16  return z; /* tako sto ce npr deljenik biti 1.0 umesto 1 */
18  }
19
20  int main()
21  {
22      int n;
23      printf("Unesi jedan pozitivan ceo broj:\n");
24      scanf("%d", &n);
25      printf("Zbir reciprocnih vrednosti brojeva od 1 do %d je %.2f\n", n
26            , zbir_reciprocnih(n));
27      /* povratna vrednost funkcije zbir_reciprocnih je float; funkciju
28         mozemo pozvati u okviru
29         naredbe printf i umesto specifikatora %.2f bice ispisana
30         povratna vrednost funkcije
31         zbir_reciprocnih zaokruzena na dve decimale */
32      return 0;
33  }
```

Rešenje 2.3.5

```
1  /*
2  Napisati funkciju koja racuna aritmeticku sredinu cifara datog celog
3  broja.
4  Napisati potom glavni program koji omogucava korisniku da unese ceo
5  broj
6  i racuna aritmeticku sredinu njegovih cifara primenom napisane
7  funkcije. Ispisati
8  izracunatu vrednost zaokruzenu na dve decimale.
9  */
10
11 #include<stdio.h>
12 #include<stdlib.h>
13
14 float aritmeticka_sredina(int x)
15 {
16     int zbir_cifara=0;
17     int broj_cifara=0;
18     char cifra;
19
20     if (x==0) /* u slucaju da je uneta 0 */
21         return 0; /* aritmeticka sredina cifara iznosi 0 i tu vrednost
22                   vratamo */
23
24     x=abs(x); /* uzimamo apsolutnu vrednost broja za slucaj da je
25               negativan */
26
27     while(x)
```

```

25     {
        cifra=x%10;

27         broj_cifara++;
        zbir_cifara+=cifra;

29         x/=10;
31     }

33     return (0.0+zbir_cifara)/broj_cifara; /* posto su zbir_cifara i
        broj_cifara celobrojne vrednosti,
                                           neophodno je da bar
        jednu od njih konvertujemo u realnu
35                                           kako bismo izbegli
        celobrojno deljenje */
    }

37 int main()
39 {
    int x;
41     printf("Unesi jedan ceo broj:");
    scanf("%d",&x);
43     printf("Aritmeticka sredina cifara broja %d iznosi %.2f\n", x,
        aritmeticka_sredina(x));
    return 0;
45 }

```

Rešenje 2.3.6

```

1  /*
    Napisati funkciju koja za dva realna broja x i y i jedan neoznaceni
    ceo broj n
3  ispisuje vrednosti funkcije sin u n ravnomerno rasporedjenih tacaka
    intervala [x,y].
    Napisati potom glavni program koji omogucava korisniku da unese
    potrebne vrednosti
5  i poziva napisanu funkciju.
    */

7
9  #include <stdio.h>
    #include <math.h>

11 void ispis(float x, float y, int n) /* funkcija nema povratnu
    vrednost; zbog toga je povratni tip void */
    {
13         float i;
        float korak=(y-x)/(n-1);

15         for(i=x;i<=y;i+=korak)
17             printf("sin(%.4f)=%.4f\n", i,sin(i));
    }

```

2 Kontrola toka

```
19 }
21 int main()
22 {
23     float a,b;
24     int n;
25     float t;
26     printf("Unesi dva realna broja:");
27     scanf("%f%f",&a,&b);
28     printf("Unesi jedan ceo broj > 1:");
29     scanf("%u",&n);
31     if (n<=1 || a==b)
32     {
33         printf("Nekorektan unos\n");
34         return -1;
35     }
36     if (b<a) /* u slucaju da je desni kraj intervala manji od levog */
37     {       /* zamenimo im mesta */
38         t=a;
39         a=b;
40         b=t;
41     }
43
45     ispis(a,b,n);
47     return 0;
48 }
```

Rešenje ??

```
/*
2  Napisati funkciju koja broji neparne cifre u zapisu datog celog broja
   . Napisati
   potom glavni program koji unosi cele brojeve dok se ne unese nula, i
   ispisuje
4  broj neparnih cifara svakog unetog broja koriscenjem napisane
   funkcije.
*/
6
8  #include<stdio.h>
9  #include<stdlib.h>
10
12 int broj_ncifara(int x)
13 {
14     int s=0;
15     char cifra;
16     x = abs(x);
```

```

16     while(x)
17     {
18         cifra = x%10;
19         s+=(cifra%2); /* izraz cifra%2 ima vrednost 1 kada je cifra
20            neparna,                a 0 kada je cifra parna */
21         x/=10;
22     }
23
24     return s;
25 }
26
27 int main()
28 {
29     int x;
30     do
31     {
32         scanf("%d",&x);
33         printf("Broj neparnih cifara u zapisu broja %d: %d\n", x,
34            broj_ncifara(x));
35     } while(x!=0);
36
37     return 0;
38 }

```

Rešenje 2.3.8

```

1  #include <stdio.h>
2
3  /*
4   Funkcija koja racuna minimum tri cela broja
5  */
6  int min(int x, int y, int z){
7      int min;
8
9      min=x;
10
11     if(min>y)
12         min=y;
13
14     if(min>z)
15         min=z;
16
17     return min;
18 }
19
20 int main(){
21     int x,y,z;
22
23     /* Ucitavamo brojeve */
24     printf("Unesite brojeve: ");

```

2 Kontrola toka

```
25 scanf("%d%d%d", &x, &y, &z);

27 /* Pozivamo funkciju i ispisujemo rezultat */
printf("Minimum je: %d\n", min(x,y,z));

29 return 0;

31 }
```

Rešenje 2.3.9

```
1 #include <stdio.h>

3 /* Funkcija koja racuna apsolutnu vrednost */
unsigned int apsolutna_vrednost(int x){
5     /* Kako funkcija vraca unsigned, a x je tipa int, vrsimo kastovanje
    rezultata u tip unsigned */
    return (unsigned)(x<0?-x:x);
7 }

9 int main(){
    int n;

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

15     /* Ispisujemo njegovu apsolutnu vrednost */
17     printf("Apsolutna vrednost: %u\n", apsolutna_vrednost(n));

19     return 0;

    }
```

Rešenje 2.3.10

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

3 /* Funkcija koja vraca razlomljeni deo prosledjenog broja */
5 float razlomljeni_deo(float x){

7     /* Funkcija fabs vraca apsolutnu vrednost realnog broja
    * NAPOMENA: funkcija fabs se nalazi u zaglavlju math.h
    * NAPOMENA2: funkcija abs se nalazi u zaglavlju stdlib.h, ali se
    koristi samo za cele brojeve!
    */
11     x = fabs(x);

13     /* Razlomljeni deo broja dobijamo tako sto od samog broja oduzmemo
    njegov ceo deo*/
    }
```

```
    return x - (int)x;
15 }

17 int main(){
    float n;

19     /* Ucitavamo broj */
21     printf("Unesite broj:");
    scanf("%f", &n);

23     /* Ispisujemo rezultat */
25     printf("Razlomljeni deo: %.6f\n", razlomljeni_deo(n));

27     return 0;
}
```

Rešenje 2.3.11

```
1  #include<stdio.h>

3  /* Funkcija koja iscrtava romb */
void romb(int n){
5     int i, j;

7     /* U svakoj liniji */
    for(i=0; i<n; i++){

9         /* Prvo ispisujemo n-i-1 razmaka */
11        for(j=0; j<n-i-1; j++)
            printf(" ");

13        /* Zatim ispisujemo n zvezdica */
15        for(j=0; j<n; j++)
            printf("*");

17        /* Na kraju svake linije stoji oznaka za novi red */
19        printf("\n");
    }

21 }

23 int main(){
25     int n;

27     /* Ucitavamo broj n */
    printf("Unesite broj n: ");
29     scanf("%d", &n);

31     /* Proveravamo korektnost ulaza i ispisujemo rezultat */
    if(n<=0)
33     printf("Greska: pogresna dimenzija!\n");
}
```

```
    else
35     romb(n);

37     return 0;
}
```

Rešenje 2.3.12

```
#include<stdio.h>

2
/* Funkcija koja stampa n zvezdica za kojima sledi znak za novi red
   */
4 void stampaj_zvezdice(int n){
    int i;
6     for(i=0; i<n; i++)
        printf("*");
8
    printf("\n");
10 }

12 /* Funkcija koja crta grafikon */
void grafikon_h(int a, int b, int c, int d)
14 {
    int i;
16
    /* Prvo ispisujemo a zvezdica */
18     stampaj_zvezdice(a);

20     /* Zatim u sledecem redu b zvezdica */
    stampaj_zvezdice(b);

22     /* Zatim u sledecem redu c zvezdica */
24     stampaj_zvezdice(c);

26     /* Zatim u poslednjem redu d zvezdica */
    stampaj_zvezdice(d);
28 }

30
32 int main(){
    int a,b,c,d;

34     /* Ucitavamo vrednosti a,b,c,d */
    printf("Unesite vrednosti: ");
36     scanf("%d%d%d%d", &a, &b, &c, &d);

38     /* Proveravamo korektnost ulaza i ispisujemo rezultat */
    if(a <0 || b<0 || c<0 || d<0){
40         printf("Greska: pogresan unos!\n");
    }else{
42         grafikon_h(a,b,c,d);
    }
```



```

    }
44     return 0;
46 }

```

Rešenje 2.3.13

```

1  #include<stdio.h>

3  int maksimum(int a, int b, int c, int d){
    int max;

5     max=a;
7     if(b>max)
        max=b;
9     if(c>max)
        max=c;
11    if(d>max)
        max=d;

13    return max;
15 }

17 /* Funkcija koja iscrtava vertikalni grafikon */
void grafikon_v(int a, int b, int c, int d){
19     int i, max;

21     /* Na pocetku je potrebno pronaci najveću od ove cetiri vrednosti
        */
    max=maksimum(a, b, c, d);

23     /* Grafikon ukupno ima max horizontalnih linija */
25     for(i=0; i<max; i++){

27         /* U svakoj od horizontalnih linija se nalazi po 4 polja:
            polje za a,b,c i d uspravnu liniju.
29         U svako od polja treba da se upise ili * ili belina,
            u zavisnosti od vrednosti a i toga u kojoj liniji se trenutno
            nalazimo
31         */

33         /* Proveravamo uslov za polje a */
        if(i<max-a)
35             printf(" ");
        else
37             printf("*");

39         /* Proveravamo uslov za polje b */
        if(i<max-b)
41             printf(" ");
        else

```

2 Kontrola toka

```
43     printf("*");
45     /* Proveravamo uslov za polje c */
46     if(i<max-c)
47         printf(" ");
48     else
49         printf("*");
51     /* Proveravamo uslov za polje d */
52     if(i<max-d)
53         printf(" ");
54     else
55         printf("*");
57     /* Na kraju svake horizontalne linije stampamo novi red */
58     printf("\n");
59 }
61
62 int main(){
63     int a,b,c,d;
65     /* Ucitavamo vrednosti a,b,c,d */
66     printf("Unesite vrednosti: ");
67     scanf("%d%d%d%d", &a, &b, &c, &d);
69     /* Proveravamo korektnost ulaza i stampamo grafikon */
70     if(a < 0 || b < 0 || c < 0 || d < 0)
71         printf("Greska: pogresan unos!\n");
72     else
73         grafikon_v(a,b,c,d);
75     return 0;
76 }
```

Rešenje 2.3.14

```
1 #include<stdio.h>
3 /* Funkcija koja proverava da li je godina prestupna */
4 int prestupna(int godina){
5     if((godina %100 != 0 && godina%4 == 0) || godina%400 == 0)
6         return 1;
7     else
8         return 0;
9 }
11 /* Funkcija koja proverava da li postoji prestupna godina u datom
    intervalu */
12 int postoji_prestupna(int g1, int g2){
13     for(; g1<=g2; g1++){
```

```

15     if(prestupna(g1))
16         return 1;
17     }
18     return 0;
19 }
20
21 int main(){
22
23     int g1, g2;
24
25     /* Ucitavamo godine */
26     printf("Unesite dve godine: ");
27     scanf("%d%d", &g1, &g2);
28
29     /* Proveravamo korektnost ulaza */
30     if(g1 < 0 || g2 < 0 || g1>g2){
31         printf("Greska: pogresan unos!\n");
32     }
33     else{
34
35         /* Proveravamo da li uopste postoji prestupna godina u datom
36            intervalu */
37         if(postoji_prestupna(g1,g2)){
38             /* Ako postoje, ispisujemo ih */
39             printf("Prestupne godine su: ");
40             for(; g1<=g2; g1++){
41                 if(prestupna(g1))
42                     printf("%d ", g1);
43             }
44             printf("\n");
45         }else{
46             /* U suprotnom, stampamo odgovarajucu poruku */
47             printf("Nema prestupnih godina u ovom intervalu!\n");
48         }
49     }
50     return 0;
51 }

```

Rešenje 2.3.16

```

1  #include <stdio.h>
2
3  /* Funkcija koja racuna zbir delilaca broja x */
4  int zbir_delilaca(int x){
5      int i=0;
6
7      /* Na pocetku zbir inicijalizujemo na 0 */
8      int zbir = 0;
9
10     /* Svaki broj izmedju 1 i x koji deli broj x dodajemo u zbir. */
11     for(i=1; i<=x; i++){

```

```
13     if(x % i == 0)
        zbir += i;
    }
15
    /* Vracamo dobijeni zbir */
17     return zbir;
}
19
20 int main(){
21     int k, i;
22
23     /* Ucitavamo broj k */
24     printf("Unesite broj k:");
25     scanf("%d", &k);
26
27     /* Proveravamo korektnost ulaza */
28     if(k <= 0)
29         printf("Greska: pogresan unos!\n");
30     else{
31
32         /*Za svaki broj od 1 do k ispisujemo zbir delilaca*/
33         for(i=1; i<=k; i++)
34             printf("%d ", zbir_delilaca(i));
35
36         printf("\n");
37     }
38
39
40
41     return 0;
42 }
```

Rešenje [2.3.16](#)

Rešenje [2.3.16](#)

Rešenje [2.3.16](#)

Rešenje [2.3.16](#)

Rešenje [2.3.16](#)

Rešenje [2.3.21](#)

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

```

3  /* Funkcija koja racuna zbir delilaca broja x */
   int zbir_delilaca(int x){
5     int i=0;

7     /* Na pocetku zbir inicijalizujemo na 0 */
     int zbir = 0;

9     /* Svaki broj izmedju 1 i x koji deli broj x dodajemo u zbir. */
11    for(i=1; i<=x; i++){
        if(x % i == 0)
13        zbir += i;
    }

15    /* Vracamo dobijeni zbir */
17    return zbir;
   }

19  int main(){
21
22      int k, i;
23
24      /* Ucitavamo broj k */
25      printf("Unesite broj k:");
26      scanf("%d", &k);
27
28      /* Proveravamo korektnost ulaza */
29      if(k <= 0)
30          printf("Greska: pogresan unos!\n");
31      else{
32
33          /*Za svaki broj od 1 do k ispisujemo zbir delilaca*/
34          for(i=1; i<=k; i++)
35              printf("%d ", zbir_delilaca(i));
36
37          printf("\n");
38      }
39
40
41      return 0;
   }

```

Rešenje 2.3.22

```

1  #include <stdio.h>
2
3  /* Funkcija koja uklanja broj stotina iz broja n */
4  int ukloni_stotine(int n){
5
6      /* Ako je broj izmedju -100 i 100 nema cifru desetica pa onda
       vracamo isti taj broj */
       if(n>-100 && n<100)

```

```
8   return n;
9   else
10  {
11      /* U suprotnom vracamo broj sa uklonjenom cifrom stotina */
12
13      /* Odredjujemo znak broja */
14      int znak=(n<0)? -1 : 1;
15
16      /* I nadalje radimo sa apsolutnom vrednoscu broja */
17      n=abs(n);
18
19      return znak*((n/1000)*100 + n%100);
20  }
21
22  /* Funkcija koja vraca znak broja */
23  int znak(int broj){
24      return broj<0?-1:1;
25  }
26
27  int main(){
28
29      int broj;
30
31      while(1){
32
33          /* Ucitavamo broj sa standardnog ulaza */
34          printf("Unesite broj: ");
35          scanf("%d", &broj);
36
37          /* Broj 0 oznacava kraj rada */
38          if(broj == 0)
39              break;
40
41          /* Ispisujemo rezultat, vodeci racuna da program treba da radi
42             ispravno i za negativne brojeve */
43          printf("%d\n", znak(broj)*ukloni_stotine(abs(broj)));
44      }
45
46      return 0;
47  }
```

Rešenje 2.3.23

```
1  #include<stdio.h>
2  #include<math.h>
3
4  int rotacija(int n){
5
6      /* U promenljivoj broj pamtimo originalnu vrednost n */
```

```

8      int broj, br = 0, znak;

10     /* Odredjujemo znak broja */
    znak=(n<0) ? -1: 1;

12     /* I nadalje radimo sa apsolutnom vrednoscu broja */
    n=abs(n);

14     /* U promenljivoj broj cuvamo kopiju broja n */
    broj=n;

16     /* Ako je broj jednocifren, nema potrebe da ga rotiramo. */
    if(n>-10 && n < 10)
20         return n;

22     /* Petljom izdvajamo cifru po cifru, kako bismo dosli do krajnje
        leve cifre broja
        (one koja treba da postane krajnje desna), npr za n = 1234, treba
        da dobijemo 1,
24         zatim da "pomerimo" 234 u levo i da na kraj nalepimo 1 = 2341 */

26     /* Na kraju ove petlje, u n se nalazi najlevlja cifra broja (koja
        treba da postane krajnje desna),
        dok se u br nalazi broj cifara unetog broja */
28     while(n >=10){
        n/=10;
30         br++;
    }

32     /*
34     Levi deo (234) dobijamo kao n%(10^broj_cifara)
        Zatim levi deo pomnozimo sa 10, da bi dobili 2340
36     Zatim na levi deo dodamo desni deo (1) koja se nalazi u
        promenljivoj n
        */

38     return znak* ((broj%(int)pow(10, br))*10 + n);
40 }

42 int main(){

44     int n;
    while(1){

46         /* Ucitavamo broj */
48         printf("Unesite broj: ");
        scanf("%d", &n);

50         /* Ako je uneta 0, izlazimo iz petlje */
52         if(n == 0)
            break;

54

```

2 Kontrola toka

```
56  /* Stampamo broj rotiran za jedno mesto u levo */
    printf("%d\n", rotacija(n));
    }
58
60  return 0;
    }
```

Rešenje 2.3.24

```
1  /*
   Napisati funkciju koja ispituje da li je dati ceo broj prost.
   Funkcija treba
3  da vrati 1 ako je broj prost i 0 u suprotnom. Napisati potom glavni
   program
   koji za uneti ceo broj n ispisuje prvih n prostih brojeva.
5  */

7  #include <stdio.h>
   #include <math.h>
9
11 int prost (int x) /* 1-broj je prost, 0-broj nije prost */
   {
13     int i;

15     if (x==2 || x==3) /* brojevi 2 i 3 su prosti */
        return 1;

17     if (x%2==0)          /* parni brojevi nisu prosti */
        return 0;

19     for (i=3; i<=sqrt(x);i+=2) /* trazimo delioca */
        if (x%i==0) /* ako je pronadjen, to znaci da broj nije prost */
            return 0; /* završavamo funkciju */

23     /* ukoliko izvršavanje funkcije dodje do poslednje naredbe return,
       to znaci da broj nije ispunio nijedan od prethodnih uslova
       (nije ni 2, ni 3, ni paran, niti ima ijednog delioca), odakle
       sledi da je prost i zbog toga vratamo 1
25     */
27     return 1;
   }

31
33 int main()
   {
35     int n;
        scanf("%d",&n);
        int i,j;

37     i=1; /* kandidat za prost broj */
        j=0; /* brojac prostih brojeva */
39 }
```



```

41 while(j<n)
{
    if (prost(i))          /* ako je broj prost */
43     {
        printf("%d\n", i); /* stampamo ga i */
45         j++;             /* uvecavamo brojac prostih brojeva */
    }
47     i++; /* bilo da je i prost ili ne, uvecavamo ga za 1 i
        nastavljam sa sledecom iteracijom */
}
49
51 return 0;
}

```

Rešenje 2.3.25

```

1  /*
   Napisati funkciju koja ispituje da li se cifra c nalazi u zapisu
   celog broja x.
3  Napisati potom glavni program koji za uneti ceo broj i unetu cifru
   poziva
   napisanu funkciju i ispisuje odgovarajucu poruku.
5  */

7  #include<stdio.h>
   #include<stdlib.h>
9
11 int sadrzi(int x, int c)
{
13     char cifra;
    x=abs(x);
    while(x)
15     {
        cifra = x%10;
17         if (cifra==c)
            return 1;
19         x/=10;
    }
21     return 0;
}
23 int main()
{
25     int x;
    int c;
27     printf("Unesi jedan ceo broj i jednu cifru:");
    scanf("%d%d",&x,&c);
29     if (sadrzi(x,c))
        printf("Cifra %d se nalazi u zapisu broja %d\n",c,x);
31     else
        printf("Cifra %d se ne nalazi u zapisu broja %d\n",c,x);
33     return 0;
}

```

```
}
```

Rešenje 2.3.26

```
/*
2
a) Napisati funkciju sve_parne_cifre koja ispituje da li se dati ceo
    broj sastoji isključivo iz parnih cifara. Funkcija treba
4 da vrati 1 ako su sve cifre broja parne i 0 u suprotnom.

6 b) Napisati funkciju sve_cifre_jednake koja ispituje da li su sve
    cifre datog celog broja jednake. Funkcija treba
    da vrati 1 ako su sve cifre broja jednake i 0 u suprotnom.
8
c) Napisati potom glavni program koji na uneti ceo broj primenjuje
    napisane funkcije i ispisuje odgovarajuće poruke.
10
Na primer, za uneti broj 222, program treba da ispise:
12 Sve cifre broja su parne.
    Sve cifre broja su jednake.
14
A za uneti broj -284:
16 Sve cifre broja su parne.
    Broj sadrži različite cifre
18
*/
20 #include <stdio.h>
    #include <stdlib.h>
22
int sve_parne_cifre(int x) /* funkcija vraća 1 ako su sve cifre broja
    parne i 0 u suprotnom */
24 {
    char d;
26 x=abs(x);          /* uzimamo apsolutnu vrednost broja za slučaj da je
        broj negativan */
    while (x>0)
28 {
        d=x%10;          /* izdvajamo cifru broja */
30
        if (d%2==1)      /* u slučaju da je neparna, to znači da nisu sve
            cifre broja parne */
32             return 0;    /* vraćamo 0 */

        x/=10;          /* "uklanjamo" poslednju cifru broja celobrojnim
            deljenjem sa 10 */
34     }
36
    return 1;          /* ukoliko se while petlja završila, to znači da
        uslov d%2==1 nije
38             nijednom bio ispunjen i da su sve cifre broja
            parne; zbog toga
```

```

40         */
41     }
42 }
43
44 int sve_cifre_jednake(int x) /* funkcija vraca 1 ako su sve cifre
    broja jednake i 0 u suprotnom */
45 {
46     char d;
47     char prva_cifra;
48     x=abs(x);
49     prva_cifra = x%10; /* izdvajamo prvu cifru broja */
50     x/=10;             /* broj delimo sa 10 jer smo prvu cifru vec
        izdvojili */
51
52     while(x)
53     {
54         d = x%10;
55
56         if (d!=prva_cifra)
57             return 0;
58
59         x/=10;
60     }
61
62     return 1;
63 }
64 main()
65 {
66     int x;
67     int d;
68
69     printf("unesi ceo broj:");
70     scanf("%d", &x);
71
72     if (sve_parne_cifre(x))
73         printf("Sve cifre broja su parne\n");
74     else
75         printf("Broj sadrzi bar jednu neparnu cifru\n");
76
77     if (sve_cifre_jednake(x))
78         printf("Sve cifre broja su jednake\n");
79     else
80         printf("Broj sadrzi razlicite cifre \n");
81 }

```

Rešenje ??

```
/*
```

2 Kontrola toka

```
2  Napisati funkciju koja za dva uneta neoznacena broja x i n utvrđuje
   da li je x neki stepen
   broja n. Ukoliko jeste, funkcija vraća izlozilac stepena, a u
   suprotnom vraća -1. Napisati
4  potom glavni program koji testira ovu funkciju.
   */
6
8  #include <stdio.h>
10
12  int je_stepen(unsigned x, unsigned n) /* funkcija vraća izlozilac
   stepena ukoliko broj x jeste neki stepen broja n */
14  {
16      int i=1;
18      int s=n;
20
22      while(s<x)
24      {
26          s=s*n;
28          i++;
30      }
32
34      if (s==x)
36          return i;
38
40      return -1;
42  }
44
46  int main()
48  {
50      unsigned x;
52      unsigned n;
54      int st;
56
58      scanf("%u%u",&x,&n);
60
62      st = je_stepen(x,n);
64
66      if (st!=-1)
68          printf("%u=%u~%d\n",x,n,st);
70      else
72          printf("%u nije stepen broja %u\n",x,n);
74
76      return 0;
78  }
```

Rešenje ??

```
2  /*
4  Napisati funkciju
```

```
double e_na_x(double x, double eps)
6
    koja racuna vrednost  $e^x$  kao parcijalnu sumu reda
8    suma( $x^n/n!$ ), gde indeks n ide od
    od 0 do beskonacno, pri cemu se sumiranje vrsi dok
10   je razlika sabiraka u redu po apsolutnoj vrednosti
    manja od eps. Napisati potom program koji omogucuje
12   korisniku da unese jedan realan broj x i ispisuje
    vrednost  $e^x$ .
14
    */
16
    #include<stdio.h>
18   #include<math.h>

20   double e_na_x(double x, double eps)
    {
22       double s=1;
        double clan=1;
24       int n=1;

26       /*
        parcijalnu sumu formiramo tako sto u svakoj iteraciji petlje
28       promenljivoj s dodamo jedan sabirak sume oblika  $(x^n)/n!$  koji
        cuvamo u promenljivoj clan

        svaki sabirak mozemo da dobijemo na osnovu prethodnog tako sto
32       ga pomnozimo sa x i podelimo sa n, koje predstavlja redni broj
        sabirka u sumi

        prvi sabirak (kome odgovara  $n=0$ ) iznosi 1; zbog toga promenljive
36       s i clan inicijalizujemo na vrednost 1

        sumiranje se sprovodi dogod je sabirak po apsolutnoj vrednosti
38       veci od trazene tacnosti eps
40       */

42       do
        {
44         clan = (clan*x)/n;
          s += clan;
46         n++;
        } while(fabs(clan)>eps);
48
        return s;
50   }

52   int main()
    {
54       double x,eps;
        printf("x=");
56       scanf("%lf", &x);
```

```
58     printf("eps=");
    scanf("%lf", &eps);

60     printf("e~%f=%f\n", x, e_na_x(x,eps));
    return 0;
62 }
```

Rešenje 2.3.29

```
/*
2  Za dati broj moze se formirati niz tako da je svaki sledeci clan niza
    dobijen
    kao suma cifara prethodnog clana niza. Broj je srecan ako se dati niz
    zavrшава sa
4  jedinicom. Napisati program koji za uneti broj odredjuje da li je
    srecan.
    Na primer:
6  - broj 1234 je srecan jer je zbir njegovih cifara 10, dalje zbir
    cifara broja 10 je 1.
    - broj 999 nije srecan jer je njegov zbir cifara 27, zbir cifara
    broja 27 je 9.
8  - broj 991 je srecan, zbir njegovih cifara je 19, zbir cifara broja
    19 je 10, zbir cifara
    broja 10 je 1.
10 - broj 372 nije srecan, zbir njegovih cifara je 12, zbir cifara broja
    12 je 3

12 Napisati funkciju koja vraca 1 ako je broj srecan, a 0 u suprotnom.

14 Napisati program koji omogucava korisniku da unese prirodan broj,
    poziva funkciju
    i ispisuje da li je dati broj srecan. Potom traziti od korisnika da
    unese prirodan
16 broj n i ispisati sve srecne brojeve od 1 do n.
*/

18 #include<stdio.h>

20 int zbir_cifara(int x)
22 {
    int s=0;
24     char cifra;
    while(x)
26     {
        cifra = x%10;
28         s+=cifra;
        x/=10;
30     }
    return s;
32 }
```

```

34 int srecan(int x)
35 {
36     int s; /* promenljiva s sadrzi sumu cifara */
37
38     do
39     {
40         s=zbir_cifara(x);
41         x=s; /* kada izracunamo sumu cifara, dodeljujemo je promenljivoj
42             x jer iz te promenljive izdvajamo cifre u funkciji zbir_cifara
43             */
44     } while(x>=10);
45
46     return (x==1);
47 }
48
49 int main()
50 {
51     unsigned n;
52     int i;
53     printf("Unesi jedan neoznaceni broj:");
54     scanf("%u",&n);
55
56     for(i=1;i<=n;i++)
57         if (srecan(i))
58             printf("%d je srecan\n", i);
59
60     return 0;
61 }

```

Rešenje 2.3.30

```

/*
2 . a) Napisati funkciju
3
4     int konverzija (int c)
5
6 koja prebacuje veliko slovo u ekvivalentno malo i obrnuto.
7
8 b) Napisati program koji omogućava korisniku da unese niz karaktera
9 sa tastature, a potom ispisuje uneseni niz konvertovanih karaktera.
10 Na primer, za uneti tekst "Kolokvijum iz Progi je 1.12." program
11 treba da ispise "KOLOVKIJUM IZ pROGI JE 1.12."
12
13 */
14 #include <stdio.h>
15
16 int konverzija(int c)
17 {
18     /* ključna rec return vraća povratnu vrednost funkcije (ako je ima)
19     */

```

```
/* i završava izvršavanje funkcije */
20
    if (c>='A' && c<='Z')
22         return c+'a'-'A';

    if (c>='a' && c<='z')
24         return c-'a'+'A';

26     return c;
28 }

30 int main()
{
32     int c;

34     while((c=getchar())!=EOF) /* korisnik unosi karakter po karakter
        do konstante EOF */
        putchar(konverzija(c)); /* funkcija putchar ispisuje jedan
36                                karakter na standardni izlaz */

38     return 0;
}
```

Rešenje 2.3.31

```
1 #include <stdio.h>

3 /* Funkcija int zapis(int x, int y) proverava da li su dva cela broja
    napisana
    * pomocu istih cifara, kao i da li se te cifre pojavljuju
    * isti broj puta.
    * Ideja je sledeca:
    * iz broja x izdvajaju se redom cifra po cifra s kraja,
    * a zatim se svaka takva cifra traži i u broju y.
    * Ukoliko postoji, eliminiše se prvi put kada se pojavi (dakle,
    * samo jednom).
    * Ukoliko su sve cifre iste (**redosled nije bitan**),
11 * na kraju će i iz x i iz y biti sve cifre eliminisane",
    * te ostaju nule u oba broja.
13 *
    * Broj novo_y formira se, zbog jednostavnosti, pomocu Heronovog
    obrasca.
15 * Ovaj postupak obradjen je u okviru funkcije int izbaci_cifru(int
    y, int cifra).
    */

17 int izbaci_cifru(int y, int cifra) {
19     int novo_y = 0;
21     int indikator = 0;
    int izdvojena_cifra;
```



```
23 while(y) {
24
25     izdvojena_cifra = y % 10;
26     /* U slucaju da se cifra razlikuje od one koju treba eliminisati,
27      * ili ukoliko je jedna cifra vec eliminisana =>
28      * tekucu cifru ukljuciti prilikom formiranja novog y
29      */
30     if(izdvojena_cifra != cifra || indikator)
31
32         /* Heronov obrazac.
33          * Menja poredak cifara, ali on u ovom slucaju i nije bitan.
34          */
35         novo_y = novo_y*10 + izdvojena_cifra;
36     else
37
38         /* U slucaju da je cifra vec eliminisana,
39          * ne treba je opet eliminisati.
40          * Za svaku pojavu cifre iz x,
41          * eliminiše se jedna odgovarajuća pojava
42          * te cifre iz y.
43          */
44         indikator = 1;
45
46     y /= 10;
47 }
48
49 return novo_y;
50 }
51
52 int zapis(int x, int y) {
53
54     /* Cifra koja se izdvaja iz x, a onda eliminiše iz y */
55     int cifra;
56
57     /* U slucaju da su prosledjeni brojevi negativni */
58     x = abs(x);
59     y = abs(y);
60
61     while(x) {
62
63         cifra = x % 10;
64         x /= 10;
65
66         y = izbaci_cifru(y, cifra);
67
68         /* otkomentarisati donju liniju radi lakseg pracenja rada
69          programa: */
70         // printf("Iz x izdvojeno: %d\n\tx = %d, y = %d\n\n", cifra, x, y
71         );
72     }
```

2 Kontrola toka

```
73     return (x == 0 && y == 0);
74 }
75
76 int main() {
77
78     int x, y;
79     printf("Unesite dva cela broja: ");
80     scanf("%d%d", &x, &y);
81
82     if(zapis(x, y))
83         printf("Uslov je ispunjen!\n");
84     else
85         printf("Uslov nije ispunjen!\n");
86
87     return 0;
88 }
```

Rešenje 2.3.32

```
1  #include <stdio.h>
2
3  /* Funkcija racuna faktorijel broja.
4   * Faktorijel formiramo mnozenjem sa trenutnom vrednoscu broja x,
5   * a zatim smanjujuci tu vrednost za 1.
6   * Ukoliko je x = 5, f = 5 * 4 * 3 * 2 * 1
7   */
8  int faktorijel(int x) {
9
10     int f = 1;
11     while(x) {
12         f *= x;
13         x--;
14     }
15     return f;
16 }
17
18 int main() {
19
20     int x, y;
21
22     printf("Unesite dva broja: ");
23     scanf("%d%d", &x, &y);
24
25     /* Provera uslova.
26     *
27     * Faktorijel je veoma brza funkcija, tj.
28     * s povecanjem broja x, drasticno brzo uvecava se i vrednost x!.
29     * Tip podatka int ima ogranicenje u velicini broja koji moze da
30     * sadrzi.
31     * Za 13! i vece, int ne bi mogao da sacuva sve cifre potrebne za
32     * zapis tako velikog broja,
```

```

31  * te bi doslo do prekoracenja.
32  *
33  * Slicno, faktorijel nije definisan nad skupom negativnih celih
    brojeva.
34  */
35  if(x < 0 || y < 0 || x > 12 || y > 12) {
    printf("Greska: pogresan unos!\n");
37  }
    else{
39  printf("%d\n", faktorijel(x) + faktorijel(y));
    }
41  return 0;
}

```

Rešenje 2.3.33

```

1  #include <stdio.h>
    #include <stdlib.h>
3
    /* Funkcija proverava da li se
5  * cifre u zapisu broja nalaze u rastucem poretku.
    *
7  * Situacija od interesa je kada za dve uzastopne cifre to nije
    slucaj.
    * Tada ne treba proveravati i za ostale cifre,
9  * vec odmah prekinuti izvršavanje funkcije.
    *
11 * Ukoliko funkcija nije ranije prekinuta,
    * to znaci da cifre jesu u rastucem poretku
13 * (odnosno, kako izdvajamo cifre od nazad, u stvari proveravamo
    opadajuci poredak),
    * te treba vratiti 1.
15 */
17 int rastuce(int n) {
19     int tekuca_cifra;
    int prethodna_cifra;
21
    n = abs(n);
23
    /* Prvu cifru (odnosno, poslednju u zapisu broja)
25     * izdvajamo izvan petlje
    * kako bismo mogli da je poredimo sa narednom
27     */
    tekuca_cifra = n % 10;
29     n /= 10;
31     while(n) {
33         /* Cifra koja je bila tekuca u prethodnoj iteraciji petlje,

```

```
35     * u novoj iteraciji postaje prethodna.
36     *
37     * Novoizdvojena cifra je tekuca.
38     */
39     prethodna_cifra = tekuca_cifra;
40     tekuca_cifra = n % 10;
41
42     /* Ukoliko smo naisli na cifre koje kvare rastuci poredak,
43     * prekidamo izvršavanje funkcije sa odgovarajucom povratnom
44     * vrednoscu 0.
45     */
46     if(prethodna_cifra < tekuca_cifra)
47         return 0;
48
49     /* Inace, nastavljamo sa izdvajanjem cifara */
50     n /= 10;
51 }
52
53 return 1;
54 }
55
56 int main() {
57
58     int x;
59     printf("Unesite broj: ");
60     scanf("%d", &x);
61
62     if(rastuce(x))
63         printf("Cifre su u rastucem poretku!\n");
64     else
65         printf("Cifre nisu u rastucem poretku!\n");
66
67     return 0;
68 }
```

Rešenje 2.3.34

```
1 #include <stdio.h>
2
3 /* Funkcija racuna broj x na n-ti stepen */
4 int stepen(int x, int n) {
5
6     int i;
7     /* Promenljiva u kojoj se cuva proizvod broja x sa samim sobom, n
8     puta */
9     int st = 1;
10
11     for(i = 1; i <= n; i++)
12         st *= x;
13
14     return st;
15 }
```

```
15 }
16
17 /* Funkcija proverava da li je broj Armstrongov. */
18 int armstrong(int x) {
19     /* u y se cuva zbir i-tih stepena cifara */
20     int y;
21     /* stepen za koji se proverava */
22     int i = 1;
23     /* prilikom izdvajanja cifara, broj x se menja,
24      * te treba imati promenljivu koja cuva pravu vrednost x
25      */
26     int original = x;
27
28     do {
29
30         y = 0;
31         /* Racunamo i-te stepene za svaku cifru,
32          * i istovremeno te stepen sabiramo.
33          * Rezultat pamtimo u promenljivoj y.
34          */
35         while(x) {
36
37             y += stepen(x % 10, i);
38             x /= 10;
39         }
40
41         /* x je sada promenjen, pa ga treba vratiti na pravu vrednost. */
42         x = original;
43         i++;
44     } while(y < x); /* Petlju vrtimo sve dok je zbir stepena cifara
45                     manji od datog broja. */
46
47     /* Ukoliko smo nasli i, takvo da je zbir i-tih stepena cifara
48      * jednak upravo broju x, takav broj je Armstrongov,
49      * te izraz x == y vraca 1.
50      *
51      * Inace, vraca 0, tj. broj nije Armstrongov.
52      */
53     return x == y;
54 }
55
56 int main() {
57
58     int x;
59     printf("Unesite broj: ");
60     scanf("%d", &x);
61
62     if(armstrong(x))
63         printf("Broj je Armstrongov!\n");
64     else
```

```
65     printf("Broj nije Armstrongov!\n");
67     return 0;
}
```

Rešenje 2.3.35

```
1  #include <stdio.h>
   #include <stdlib.h>
3
   /* Funkcija proverava da li su dve uzastopne cifre
   * razlicite parnosti.
   *
   * Interesantna situacija je ukoliko su dve uzastopne cifre
   * obe parne, odnosno obe neparne.
   * Ovaj uslov svodimo na poredjenje njihovih ostataka pri deljenju sa
   * 2:
   * ukoliko su ostaci isti, cifre su iste parnosti,
11  * te ne treba dalje proverati da li je uslov zadovoljen,
   * vec odmah prekinuti sa izvršavanjem funkcije.
13  *
   * Ukoliko dve uzastopne cifre ni u jednom slucaju nisu bile iste
   * parnosti,
15  * a izdvojene su sve cifre iz broja x,
   * uslov je ispunjen, pa funkcija vraća 1.
17  */
   int par_nepar(int x) {
19
       int prethodna_cifra;
21       int tekuca_cifra;

23       /* u slucaju da je uneti broj negativan */
       x = abs(x);

25
       /* jednu cifru izdvajamo van petlje
       * kako bismo mogli da je odmah u petlji poredimo sa narednom
       */
27
       prethodna_cifra = x % 10;
       x /= 10;
31
       while(x) {
33
           tekuca_cifra = x % 10;

35
           if(tekuca_cifra % 2 == prethodna_cifra % 2)
               return 0;
37
           /* tekuca cifra postaje prethodna cifra za narednu iteraciju */
           prethodna_cifra = tekuca_cifra;
39
           x /= 10;
41
       }
}
```

```
43     return 1;
44 }
45
46
47 int main() {
48
49     int x;
50     printf("Unesite broj: ");
51     scanf("%d", &x);
52
53     if(par_nepar(x))
54         printf("Broj ispunjava uslov!\n");
55     else
56         printf("Broj ne ispunjava uslov!\n");
57
58     return 0;
59 }
```

Rešenje 2.3.36

```
1  #include <stdio.h>
2
3  /* Funkcija broji koliko puta se realan broj x
4   * javlja u nizu unetih brojeva sa tastature.
5   *
6   * Brojevi se unose sve do pojave 0,
7   * pa treba koristiti do..while petlju,
8   * kako bi bar jedan broj bio unet (makar bio i 0).
9   */
10 int prebrojavanje(float x) {
11
12     /* y prihvata uneti broj sa tastature */
13     float y;
14     /* br_pojavljivanja je broj koji broji koliko puta se broj x
15      * javlja u unetom nizu brojeva */
16     int br_pojavljivanja = 0;
17
18     printf("Unesite brojeve: ");
19     do {
20
21         /* Unosimo broj. */
22         scanf("%f", &y);
23
24         /* Poredimo uneti broj sa datim brojem.
25          * Ukoliko je unet bas trazeni broj,
26          * uvecavamo brojac.
27          */
28         if(x == y)
29             br_pojavljivanja++;
30     } while(y); /* Sve dok nije uneta 0 */
31 }
```

```
32     return br_pojavljivanja;
33 }
34
35 int main() {
36     float x;
37     int br_pojavljivanja;
38
39     printf("Unesite broj x: ");
40     scanf("%f", &x);
41
42     br_pojavljivanja = prebrojavanje(x);
43     printf("Broj pojavljivanja broja %.2f je: %d\n", x,
44           br_pojavljivanja);
45
46     return 0;
47 }
```

Rešenje 2.3.37

```
1  #include <stdio.h>
2
3  /* Funkcija racuna n-ti clan Fibonacijevog niza.
4   * Clanovi ovog niza zadaju se rekurzivno tj. u zavisnosti od
5   * prethodnih clanova.
6   * Fibonacijevi brojevi od 0. do 47. se mogu smestiti u tip int, a
7   * kako n moze uzimati vrednosti
8   * od 1 do 50, povratni tip funkcije je long int.
9   */
10 long int fibonaci(int n) {
11
12     int i;
13
14     /* f0 i f1 su prva dva clana niza */
15     int f0 = 1;
16     int f1 = 1;
17     /* promenljiva u kojoj se cuvaju opsti clanovi: n+2, n+1. i n-ti
18     clan */
19     long int fn2, fn1, fn;
20
21     /* ukoliko treba vratiti multi ili prvi clan,
22     * njih ne treba racunati
23     * jer su vec dati.
24     */
25     if(n == 0 || n == 1)
26         return 1;
27
28     /* postavljamo prethodne clanove niza */
29     fn = f0;
30     fn1 = f1;
```



```

29  /* racunamo od drugog clana, pa dok ne dodjemo do n-tog */
    for(i = 2; i <= n; i++) {

31      /* izracunamo n+2-i clan niza sabiranjem prethodna dva clana */
        fn2 = fn1 + fn;
33      /* promenimo prethodne clanove niza, zbog naredne iteracije */
        fn = fn1;
35        fn1 = fn2;
    }

37    return fn2;
39 }

41 int main() {

43     int n;
    printf("Unesite broj n: ");
45     scanf("%d", &n);

47     /* Provera vrednosti za broj n */
    if(n < 0 || n > 50) {
49         printf("Greska: nedozvoljena vrednost!\n");
    }
51     else{
        printf("%ld\n", fibonaci(n));
53     }

55     return 0;
}

```

Rešenje 2.3.38

```

1  #include <stdio.h>

3  /* Funkcija vraca karakter koji se u abecedi
   * nalazi k mesta pre datog karaktera c
   */
5  char sifra(char c, int k) {

7      /* Ukoliko je uneto malo slovo ... */
9      if(c >= 'a' && c <= 'z')
        /* Pri tome karakter koji je k pozicija pre datog karaktera
        ispada iz opsega malih slova ... */
11         if(c-k < 'a')
            /* Treba krenuti s drugog kraja abecede, racunajuci i
            preskocena slova.
            */
13             * Na primer, ukoliko je c = 'b' i k = 2
15             * Jedan karakter pre 'b' je 'a'.
            * Dva karaktera pre 'b' je 'z' (kruzno).
17             *

```

```
19      * Karakter iz prvog dela abecede, koji je preskocen, je 'a'.
      * Broj preskocenih karaktera iz prvog dela abecede
      * racunamo tako sto izracunamo c - 'a' (rastojanje od datog
      karaktera do malog slova a)
21      * sto je u ovom slucaju 'b' - 'a' = 1.
      *
23      * Ostatak karaktera do k ispisujemo, ali gledavsi unazad od z.
      * Zato racunamo k - (c - 'a') - 1.
25      *
      * Od k oduzimamo rastojanje izmedju c i 'a',
27      * kako bismo dobili preostali broj karaktera koji treba
      preskociti.
      */
29      return 'z' - (k - (c - 'a') - 1);
      else
31      /* U suprotnom, karakter ne ispada iz opsega malih slova, te je
      dovoljno bas njega i vratiti */
      return c-k;
33
      /* Ukoliko je uneto veliko slovo ... */
35      else if(c >= 'A' && c <= 'Z')
          if(c-k < 'A')
37              return 'Z' - (k - (c - 'A') - 1);
          else
39              return c-k;
41
      return c;
43 }

45 int main() {
47     int k;
      char c;
49
      printf("Unesite broj k: ");
51     scanf("%d", &k);

53     printf("Unesite tekst (CTRL + D za prekid): ");
      while((c = getchar()) != EOF)
55         putchar(sifra(c, k));

57     return 0;
}
```