

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	Rešenja	11
2	Kontrola toka	31
2.1	Naredbe grananja	31
2.2	Rešenja	43
2.3	Petlje	71
2.3.1	Ispis podataka	71
2.3.2	Obrada celih brojeva, rad sa ciframa broja	74
2.3.3	Unos i obrada veće količine podatka (unos i obrada niza brojeva?, nije sjajno zbog nizova)	77
2.3.4	Rad sa karakterima	82
2.3.5	Računanje sume i proizvoda	83
2.3.6	Dvostruka petlja i ispisivanje slike	89
2.4	Rešenja	96

1

Uvodni zadaci

Zadatak 1.1 Napisati program koji na standardni izlaz ispisuje tekst **Zdravo svima!**.

Primer 1

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

[Rešenje 1.1]

Zadatak 1.2 Napisati program za uneti ceo broj ispisuje taj broj, njegov kvadrat i njegov kub.

Primer 1

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

Primer 2

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

[Rešenje 1.2]

Zadatak 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
```

[Rešenje 1.3]

Zadatak 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 940
```

[Rešenje 1.4]

Zadatak 1.5 Napisati program koji za unetu količinu jabuka u kilogramima i unetu cenu po kilogramu ispisuje ukupnu vrednost date količine jabuka. NAPOMENA: *Pretpostaviti da je cena jabuka pozitivan ceo broj i da je unos korektan.*

Primer 1

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

Primer 1

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

[Rešenje 1.5]

Zadatak 1.6 Napisati program koji pomaže kasirki da obračuna kusur koji treba da vrati kupcu. Za unetu cenu artikla, količinu artikla i iznos koji je kupac dao, program treba da ispiše vrednost kusura. NAPOMENA: *Pretpostaviti da su cene svih artikala pozitivni celi brojevi, kao i da su unete vrednosti ispravne, tj. da se može vratiti kusur.*

Primer 1

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

Primer 2

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

[Rešenje 1.6]

Zadatak 1.7 Napisati program koji za uneta vremena poletanja i sletanja aviona ispisuje dužinu trajanja leta. NAPOMENA: *Pretpostaviti da su poletanje i sletanje u istom danu kao i da su sve vrednosti ispravno unete.*

Primer 1

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

Primer 2

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

[Rešenje 1.7]

Zadatak 1.8 Date su dve celobrojne promenljive. Napisati program koji razmenjuje njihove vrednosti.

Primer 1

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

Primer 2

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesi dve celobrojne vrednosti: 237 -592
|| pre zamene: x=237, y=-592
|| posle zamene: x=-592, y=237
```

[Rešenje 1.8]

Zadatak 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 1.10 Napisati program koji za uneti pozitivan trocifreni broj na standardni izlaz ispisuje njegove cifre jedinica, desetica i stotina. NAPOMENA: *Pretpostaviti da je unos ispravan.*

1 Uvodni zadaci

Primer 1

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

Primer 2

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

[Rešenje 1.10]

Zadatak 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, 200, 100, 50, 20, 10 i 1 dinar. NAPOMENA: *Pretpostaviti da je cena proizvoda pozitivan ceo broj.*

Primer 1

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

Primer 2

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

[Rešenje 1.11]

Zadatak 1.12 Napisati program koji učitava pozitivan trocifreni broj sa standardnog ulaza i ispisuje broj dobijen obrtanjem njegovih cifara. NAPOMENA: *Pretpostaviti da je unos ispravan.*

Primer 1

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

Primer 2

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesi trocifreni broj: 230  
|| Obrnuto: 32
```

[Rešenje 1.12]

Zadatak 1.13 Napisati program koji za uneti pozitivan četvorocifreni broj:

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

(e) izračunava broj koji se dobija zamenom cifre jedinice i cifre stotine

NAPOMENA: *Pretpostaviti da je unos ispravan.*

Primer 1

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

Primer 2

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

[Rešenje 1.13]

Zadatak 1.14 Napisati program koji ispisuje broj koji se dobija izbacivanjem cifre desetice 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 1.15 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
```

[Rešenje 1.15]

1 Uvodni zadaci

Zadatak 1.16 Napisati program koji učitava realnu vrednost izraženu u inčima, konvertuje tu vrednost u centimetre i ispisuje je zaokruženu na dve decimale. UPUTSTVO: *Jedan inč ima 2.54 centimetra.*

Primer 1

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

Primer 2

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

[Rešenje 1.16]

Zadatak 1.17 Napisati program koji učitava dužinu izraženu u miljama, konvertuje tu vrednost u kilometre i ispisuje je zaokruženu na dve decimale. UPUTSTVO: *Jedna milja ima 1.609344 kilometara.*

Primer 1

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

Primer 2

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

[Rešenje 1.17]

Zadatak 1.18 Napisati program koji učitava težinu izraženu u funtama, konvertuje tu vrednost u kilograme i ispisuje je zaokruženu na dve decimale. UPUTSTVO: *Jedna funta ima 0.45359237 kilograma.*

Primer 1

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

Primer 2

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

[Rešenje 1.18]

Zadatak 1.19 Napisati program koji učitava temperaturu izraženu u farenhajtima, konvertuje tu vrednost u celzijuse i ispisuje je zaokruženu na dve decimale. UPUTSTVO: *Veza između farenhajta i celzijusa je zadata narednom formulom $F = \frac{9 \cdot C}{5} + 32$*

Primer 1

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

Primer 2

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

[Rešenje 1.19]

Zadatak 1.20 Napisati program koji za unete realne vrednosti a_{11} , a_{12} , a_{21} , a_{22} ispisuje vrednost determinante matrice:

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

Pri ispisu vrednost zaokružiti na 4 decimale.

Primer 1

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

Primer 2

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

Primer 3

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

Primer 4

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

Zadatak 1.21 Napisati program koji za unete realne vrednosti dužina stranica pravougaonika ispisuje njegov obim i površinu. Ispisati tražene vrednosti zaokružene na dve decimale. NAPOMENA: *Pretpostaviti da je unos ispravan.*

Primer 1

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

Primer 2

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

[Rešenje 1.21]

Zadatak 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
```

[Rešenje 1.22]

1 Uvodni zadaci

Zadatak 1.23 Napisati program koji za unetu realnu vrednost dužine stranice jednakostraničnog trougla ispisuje njegov obim i površinu zaokružene na dve decimalne. NAPOMENA: *Pretpostaviti da je unos ispravan.*

Primer 1

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

Primer 2

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

[Rešenje 1.23]

Zadatak 1.24 Napisati program koji za unete realne vrednosti dužina stranica trougla ispisuje njegov obim i površinu zaokružene na dve decimalne. NAPOMENA: *Pretpostaviti da je unos ispravan.*

Primer 1

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

Primer 2

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

[Rešenje 1.24]

Zadatak 1.25 Pravougaonik čije su stranice paralelne koordinatnim osama zadat je svojim realnim koordinatama suprotnih temena (gornje levo i donje desno teme). Napisati program koji ispisuje njegov obim i površinu zaokružene na dve decimalne.

Primer 1

```
INTERAKCIJA SA PROGRAMOM:  
Unesite koordinate gornjeg levog temena: 4.3 5.8  
Unesite koordinate donjeg desnog temena: 6.7 2.3  
Obim: 5.90  
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: 9.37  
Povrsina: 19.56
```

Zadatak 1.26 Napisati program koji za tri uneta cela broja ispisuje njihovu aritmetičku sredinu zaokruženu na dve decimalne.

Primer 1

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

Primer 2

```
INTERAKCIJA SA PROGRAMOM:
Unesite tri cela broja: 3 -8 13
Aritmeticka sredina unetih brojeva je 2.67
```

[Rešenje 1.26]

Zadatak 1.27 Napisati program koji pomaže moleru da izračuna površinu zidova prostorije koju treba da okreči. Za unete dimenzije sobe u metrima (dužinu, širinu i visinu), program treba da ispiše površinu zidova za krečenje pod pretpostavkom da na vrata i prozore otpada oko 20%. Omogućiti i da na osnovu unete cene usluge po kvadratnom metru program izračuna ukupnu cenu krečenja. Sve realne vrednosti ispisati zaokružene na dve 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
```

[Rešenje 1.27]

Zadatak 1.28 Napisati program koji za unete pozitivne prirodne brojeve x , c i p 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, c i p: 140 2 1
Rezultat je: 1420
```

Primer 1

```
INTERAKCIJA SA PROGRAMOM:
Unesite redom x, c i p: 12345 9 2
Rezultat je: 129345
```

[Rešenje 1.28]

Jovana: Danijela mi je rekla gde se nalaze resenja. Dodati.
Jovana: Resenja nisam uspjela da pronadjem te sam zadatke resila.

Zadatak 1.29 Napisati program koji za uneta dva cela broja a i b dodeljuje promenljivoj `rezultat` vrednost 1 ako važi uslov:

1 Uvodni zadaci

- 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
```

Jovana: Po dogovoru na sastanku, umesto a,b,c zadatak je preformulisan na dve vrednosti - samo a i b. Prilagoditi resenja.

Jovana: Uradjeno

[Rešenje 1.29]

Zadatak 1.30 Napisati program koji za uneta dva cela broja ispisuje njihov maksimum.

Primer 1

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

Primer 2

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

[Rešenje 1.31]

Zadatak 1.31 Napisati program koji za uneta dva cela broja ispisuje njihov minimum.

Primer 1

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

Primer 2

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

[Rešenje 1.31]

Zadatak 1.32 Napisati program koji za unete realne vrednosti promenljivih x i y ispisuje vrednost sledećeg izraza:

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

zaokruženu na dve decimale.

Primer 1

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

Primer 1

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

[Rešenje 1.32]

1.1 Rešenja

Rešenje 1.1

```
2  /*
3  Navedeni program definiše funkciju koja se zove main.
4  Program može da definiše više funkcija,
5  ali obavezno mora da definiše funkciju koja se zove main i
6  izvršavanje programa uvek počinje od te funkcije. Pored naziva,
7  zapis svake funkcije cine i povratna vrednost funkcije (u ovom
8  slučaju int), lista argumenata koje funkcija koristi (u ovom
9  slučaju funkcija nema argumenata pa se navode samo prazne
10 zagrade, ()) i telo funkcije koje je ograničeno
11 viticastim zagradama ({ i }). O ovim pojmovima biće više reči
12 u narednim poglavljima.
13
14 Unutar tela funkcije navode se naredbe. Unutar navedenog programa
15 postoji jedna naredba koja predstavlja poziv funkcije printf.
16 Funkcija printf služi za ispis teksta na standardni izlaz (obično
17 ekran). Deklaracija ove funkcije data je u zaglavlju stdio.h
18 koje je potrebno uključiti direktivom #include na početku
19 samog programa.
20
21 Da bismo pokrenuli program, prvo ga moramo prevesti u izvršnu
22 datoteku. Na primer, ako je navedeni program sacuvan kao zdravo.c,
23 ako koristimo gcc kompajler koji je sastavni deo standardnih Linux
24 distribucija, prevodjenje iz komandne linije se vrši narednom
25 naredbom:
26 gcc zdravo.c
27
28 Ukoliko nije bilo gresaka prilikom prevodjenja, biće generisana
29 izvršna datoteka pod nazivom a.out koja se pokreće navodjenjem
```

1 Uvodni zadaci

```

    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 vrsi ispis */
    /* Specijalni karakter \n : prelazak u novi red */
38     /* Svaka naredba zavrшава 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 1.2

```

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

        Ucitavanje se vrsi sa standardnog ulaza (obicno tastatura).
24        */
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
    return 0;
40 }

```

Rešenje 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
18     /* Dodeljujemo vrednost promenljivoj rezultat. */
        rezultat = x+y;
        printf("%d + %d = %d\n", x,y,rezultat);
20
22     /*
        Mozemo ispisivati direktno vrednost izraza x-y i bez
        njegovog dodeljivanja posebno promenljivoj
        */
24     printf("%d - %d = %d\n",x,y,x-y);
        printf("%d * %d = %d\n",x,y,x*y);
26
28     /*
        Kada bilo koju aritmeticku operaciju primenimo na dve
        promenljive istog tipa (u ovom slucaju dva celobrojne
        promenljive), rezultat ce biti tog istog tipa. Specijalno,
        za operaciju deljenja: kada operator / primenimo na dva
        celobrojna argumenta x i y, kao rezultat dobijemo ceo deo
        pri deljenju broja x brojem y, a ne kolicnik.
        Na primer, rezultat primene operatora / na 7 i 2
        je 3, a ne 3.5.
        */
36     printf("%d / %d = %d\n",x,y,x/y);

```

1 Uvodni zadaci

```
38
/*
40   Operator % izracunava ostatak pri celobrojnomo 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
return 0;
48 }
```

Rešenje 1.4

Rešenje ovog zadatka svodi se na rešenje zadatka 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 1.5

Rešenje ovog zadatka svodi se na rešenje zadatka 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 1.6

```
1
#include <stdio.h>
3
int main()
5 {
   /*
7    S obzirom da su sve promenljive pozitivini celi brojevi,
   koristimo tip unsigned int (skraceno unsigned)
9   */
   unsigned cena;
11  unsigned kolicina;
   unsigned iznos;
13  unsigned kusur;

15  /*
   Ucitavamo potrebne podatke. Unutar jednog scanf-a mozemo
17  ucitati vise podataka odjednom. Za svaki treba navesti
   odgovarajuci format za tip podataka koji se unosi
19  (%u za unsigned).
   */
21  printf("Unesite cenu, kolicinu i iznos: ");
   scanf("%u%u%u", &cena, &kolicina, &iznos);
23 }
```

```
25  /* Izracunavamo kusur: */
    kusur=iznos - kolicina*cena;

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

29
    return 0;
31 }
```

Rešenje 1.7

```
#include <stdio.h>

2
int main()
4 {

6     unsigned poletanje, poletanje_sat, poletanje_minut;
    unsigned sletanje, sletanje_sat, sletanje_minut;
8     unsigned duzina, duzina_sat, duzina_minut;

10     printf("Unesite vreme poletanja: ");
    scanf("%u%u", &poletanje_sat, &poletanje_minut);

12

14     printf("Unesite vreme sletanja: ");
    scanf("%u%u", &sletanje_sat, &sletanje_minut);

16

    /* Pretvoricemo i vreme poletanja i vreme sletanja u sekunde */
18     poletanje=poletanje_sat*3600+poletanje_minut*60;
    sletanje=sletanje_sat*3600 + sletanje_minut*60;

20

    /* I izracunati razliku u sekundama */
22     duzina=sletanje-poletanje;

24     /* Izdvajamo broj sati i broj minuta. */
    duzina_sat=duzina/3600;
26     duzina_minut=(duzina%3600)/60;

28

    /* I ispisujemo rezultat */
30     printf("Duzina trajanja leta je %u h i %u min\n", duzina_sat,
        duzina_minut);

32
    return 0;
34 }
```

Rešenje 1.8

1 Uvodni zadaci

```
1 #include<stdio.h>
2 int main()
3 {
4     int x,y;
5     int p;
6
7     printf("Unesi dve celobrojne vrednosti:");
8     scanf("%d%d",&x,&y);
9
10    printf("pre zamene: x=%d, y=%d\n",x,y);
11
12    /* Pomocna promenljiva p je potrebna da sacuva vrednost
13       promenljive x pre nego sto se ona izmeni i dobije
14       vrednost promenljive y. */
15    p=x;
16    x=y;
17    y=p;
18    printf("posle zamene: x=%d, y=%d\n",x,y);
19    return 0;
20 }
```

Rešenje 1.10

```
1 #include <stdio.h>
2 int main()
3 {
4     /* S obzirom da broj treba da bude pozitivan,
5        koristimo tip unsigned.
6     */
7     unsigned x;
8
9     /*
10     Promenljive koje cuvaju cifre treba da budu
11     najmanjeg celobrojnog tipa jer nece sadrzati
12     druge vrednosti osim jednocifrenih celih
13     brojeva. Zbog toga za njih biramo tip char.
14     */
15
16     char cifra_jedinice;
17     char cifra_desetice;
18     char cifra_stotine;
19
20     printf("Unesi trocifreni broj:");
21     scanf("%u", &x);
22
23     /*
24     Na primer, neka je uneti broj 374. Potrebno je da koriscenjem
25     racunskih operacija za rad sa celim brojevima pristupimo
26     njegovoj cifri jedinice, cifri desetice i cifri stotine.
27     */
28 }
```

```

29     Primitimo 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     /*
70     2. nacin, bez uvođenja dodatnih promenljivih cifra_jedinice,
71     cifra_desetice i cifra_stotine:
72
73     printf("Cifre unetog broja su %d,%d,%d\n", x%10, (x/10)%10, x
74         /100);
75     */
76     return 0;
77 }

```

Rešenje 1.11

```
1  #include <stdio.h>
3
5  int main()
6  {
7      unsigned x;
8      printf("Unesi cenu:");
9      scanf("%u", &x);
10
11     /*
12      Na primer, neka je uneta cena 8347 dinara.
13      Vrednost x/5000 predstavlja broj novcanica
14      od 5000 dinara pomocu kojih mozemo sakupiti
15      celokupnu sumu. 8347 celobrojno deljeno sa
16      5000 (operacija / nad celim brojevima) iznosi 1.
17     */
18     printf("%u=%u*5000+ ", x,x/5000);
19     /*
20      Potrebna nam je 1 novcanica od
21      5000 dinara, a koliko nam je potrebno ostalih
22      novcanica? Za to moramo pristupiti preostaloj
23      sumi. Jedan nacin je da nadjemo ostatak pri deljenju
24      unete vrednosti x (u primeru 8347) sa 5000 (operacija %).
25      On iznosi 3347. Ovu vrednost dodeljujemo promeljivoj x.
26     */
27     x=x%5000;
28
29     /*
30      Nastavljamo postupak trazanjem broja novcanica
31      od 2000 dinara i redom za ostale monete.
32     */
33     printf("%u*2000 +", x/2000);
34     x=x%2000;
35     printf("%u*1000 +", x/1000);
36     x=x%1000;
37     printf("%u*500 +", x/500);
38     x=x%500;
39     printf("%u*200 +", x/200);
40     x=x%200;
41     printf("%u*100 +", x/100);
42     x=x%100;
43     printf("%u*50 +",x/50);
44     x=x%50;
45     printf("%u*20 +", x/20);
46     x=x%20;
47     printf("%u*10 +", x/10);
48     x=x%10;
49     printf("%u*1\n", x);
50     return 0;
51 }
```


Rešenje 1.12

```
1  #include <stdio.h>
3  int main()
4  {
5      unsigned x;
6      unsigned obrnuto_x;
7
8      char cifra_jedinice;
9      char cifra_desetice;
10     char cifra_stotine;
11
12     printf("Unesi trocifreni broj:");
13     scanf("%u", &x);
14
15     cifra_jedinice = x%10;
16     cifra_desetice = (x/10)%10;
17     cifra_stotine = x/100;
18
19     obrnuto_x = cifra_jedinice*100 +
20                cifra_desetice*10 +
21                cifra_stotine;
22
23     printf("Obrnuto: %u\n", obrnuto_x);
24
25     return 0;
26 }
```

Rešenje 1.13

```
1  #include <stdio.h>
3  int main(){
4
5      unsigned n, broj_obrnuto, broj_zamena;
6      char j, d, s, h;
7      int proizvod_cifara, razlika_cifara, suma_kvadrata;
8
9      /* Ucitavamo vrednost sa ulaza */
10     printf("Unesite cetvorocifreni broj: ");
11     scanf("%u", &n);
12
13     /* Izdvajamo cifre broja i to redom: j -jedinice,
14        d - desetice, s - stotine i h - hiljade */
15     j=n%10;
16     d=(n/10)%10;
```

1 Uvodni zadaci

```
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
    /* Odredjujemo broj u kojem su cifra jedinica i
       cifra stotina zamenile mesta
37  */
39  broj_zamena=h*1000+j*100+d*10+s;
    printf("Broj sa zamenjenom cifrom jedinica i stotina: %u\n",
           broj_zamena);
41
43
    return 0;
45 }
```

Rešenje 1.15

```
#include <stdio.h>
2
int main()
4 {
    unsigned broj, novibroj;
    unsigned levi, desni, m;
    6
    printf("Unesite pozitivan prirodan broj: ");
    8
    scanf("%u", &broj);
    printf("Unesite pozitivan dvocifreni broj:");
    10
    scanf("%u", &m);
    12
    /*
    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
22      levi = broj/1000;
      desni = broj%1000;
24
    /*
      Kada levi deo pomnozimo sa 100 000, dobijamo 1 200 000
      Kada m pomnozimo sa 1000, dobijamo          67 000
      Dobijene vrednosti saberemo sa desnim delom          345
28                                     -----
      Konacan rezultat:                                1 267 345
30
    */
32      novibroj = levi*100000+m*1000+desni;
34      printf("Novi broj je %u\n", novibroj);
36      return 0;
}

```

Rešenje 1.16

```

1  #include <stdio.h>
3  int main()
4  {
5      /* float - realni tip jednostruke tacnosti */
      float in;
7      float cm;
9      printf("Unesi broj inca: ");
11     /* "%f" - format za unos/ispis float promenljivih */
      scanf("%f", &in);
13
      /* 1 inch = 2.54 cm */
15     cm = in*2.54;
17
      /* "%.2f" - ispis realne promenljive na 4 decimale */
      printf("%.2f in = %.2f cm\n", in, cm);
19
      return 0;
21 }

```

Rešenje 1.17 Zadatak se radi analogno zadatku 1.16.

Rešenje 1.18 Zadatak se radi analogno zadatku 1.16.

Rešenje 1.19 Zadatak se radi analogno zadatku 1.16.

Rešenje 1.21

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

Rešenje 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
8   Ukoliko se koristi i neka funkcija matematičke
9   biblioteke, za prevodjenje je neophodno uključiti
10   opciju -lm
11   npr. gcc primer.c -lm
12  */
13  int main()
14  {
15      float r;
16      float O;
17      float P;
```

```
19 printf("Unesite duzinu poluprecnika kruga:");
scanf("%f", &r);

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

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

25 return 0;

27 }
```

Rešenje 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;
    P = (a*a*sqrt(3))/4;

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

14     return 0;

16 }
```

Rešenje 1.24

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

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

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

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

12     /* Površina - koristice Heronov obrazac */
    s=obim/2;

14 }
```

1 Uvodni zadaci

```
18     povrsina=sqrt(s*(s-a)*(s-b)*(s-c));  
    /* Ispisujemo trazene vrednosti */  
20     printf("Obim: %.2f\n", obim);  
    printf("Povrsina: %.2f\n", povrsina);  
22  
    return 0;  
24 }
```

Rešenje 1.26

```
1  #include<stdio.h>  
3  
4  int main()  
5  {  
6      int a, b, c;  
7      float as;  
9      printf("Unesite tri cela broja:");  
    scanf("%d%d%d",&a,&b,&c);  
11  
    /* pogresan nacin: as = (a+b+c)/3;  
13  
    Ukoliko podelimo zbir a+b+c sa 3, to ce biti primena  
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 decimale. 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  
    /*  
33    Trazeni kolicnik mozemo dobiti na razne nacine:  
    as=1.0*(a+b+c)/3;  
35    ili  
    as=(0.0+a+b+c)/3;  
37    ili  
    as=((float)(a+b+c))/3;  
39    itd.  
    */
```

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

Rešenje 1.27

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

Rešenje 1.28

```
1  #include <stdio.h>
3  #include <math.h>
5  int main()
6  {
```

```
7  unsigned x, p;
   char c;
9  unsigned levo, desno;
   unsigned novo_x;

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

21
      Ukoliko specifikatore promenljivih u format stringu pisemo
23      spojeno, tada ih prilikom unosa mozemo razdvojiti bilo kojim
      karakterom. Zbog toga blanko znakove u format stringu funkcije
25      scanf treba izbegavati i ovo je redak slucaj kada je njihova
      upotreba opravdana.

27
      Ako zelimo da odstampamo znak ", u format stringu
29      funkcije printf navodimo \".
   */
31  printf("Unesite vrednosti u formatu \"x p c\": ");
   scanf("%u %u %c", &x, &p, &c);
33
   /*
35      Kada ucitavamo karaktersku promenljivu, njena numericka
      vrednost je jednaka ASCII kodu unetog karaktera. Na primer,
37      ako karakter '0' ucitamo u promenljivu c, njena numericka
      vrednost bice 48. Da bismo pretvorili ovu numericku vrednost
39      u numericku vrednost koja odgovara cifri, od nje oduzimamo
      ASCII kod karakterske konstante '0' koji iznosi upravo 48.
41
   */
   c=c-'0';
43

45  /* Odredjujemo deo broja koji se nalazi desno od pozicije p */
   desno=x%(unsigned)pow(10, p);
47
   /* 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 1.29

```
1  #include <stdio.h>
3  int main()
4  {
5      int a,b,resultata,resultatb,resultatc;
7      printf("Unesite dva cela broja:");
8      scanf("%d%d", &a,&b);
9
10     /*
11      Izraz a!=b ima vrednost 1 ako je ova
12      relacija tacna, a 0 ako je netacna
13      */
14     resultata = a!=b;
15
16     /*
17      Izraz a%2==0 && b%2==0 je konjunkcija
18      koja se sastoji od dve relacije jednakosti.
19      Izraz a%2==0 ima vrednost 1 ako je
20      ova relacija tacna, a 0 u suprotnom.
21      */
22
23     resultatb = (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
32     resultatc = (a>0 && a<=100 && b>0 && b<=100);
33
34     printf("a) rezultat=%d\n", resultata);
35     printf("b) rezultat=%d\n", resultatb);
36     printf("c) rezultat=%d\n", resultatc);
37
38     return 0;
39 }
```

Rešenje 1.31

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

1 Uvodni zadaci

```
3 int main()
4 {
5     int a,b,max;
6     printf("Unesite dva cela broja:");
7     scanf("%d%d", &a,&b);
8
9     /*
10      Ternarni operator uslova :? koristi se u sledecem
11      obliku:
12
13          izraz1 ? izraz2 : izraz3;
14
15      Izraz izraz1 se izracunava prvi. Ako je njegova
16      vrednost razlicita od nule (tj. ako ima istinitosnu
17      vrednost tacno), onda se izracunava vrednost izraza
18      izraz2 i to je vrednost citavog uslovnog izraza. U
19      suprotnom, izracunava se vrednost izraz3 i to je
20      vrednost citavog uslovnog izraza
21
22      */
23
24     max = (a>b) ? a : b;
25
26     printf("Maksimum je %d\n", max);
27 }
```

Rešenje 1.31

Rešenje 1.32

```
1 #include <stdio.h>
2
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
16     min = (a<b) ? a : b;
17     max = (a>b) ? a : b;
18
19    /*
20     Racunamo vrednost promenljive rez
21
22     */
23
24     rez = (min+0.5)/(1+max*max);
```

```
21     printf("Rezultat je %.2f\n", rez);  
23     return 0;  
25 }
```


2

Kontrola toka

2.1 Naredbe grananja

Zadatak 2.1 Napisati program koji za dva uneta cela broja ispisuje njihov minimum.

Primer 1

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

Primer 2

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

[Rešenje 2.1]

Zadatak 2.2 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 2.3 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
```

[Rešenje 2.3]

Zadatak 2.4 Napisati program koji za uneti ceo broj ispisuje njegovu recipročnu vrednost zaokruženu na četiri decimala.

Primer 1

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

Primer 2

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

Primer 3

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

Primer 4

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

[Rešenje 2.4]

Zadatak 2.5 Napisati program koji učitava tri cela broja i ispisuje zbir pozitivnih.

Primer 1

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

Primer 2

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

Primer 3

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

Primer 4

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

[Rešenje 2.5]

Zadatak 2.6 U prodavnici je organizovana akcija da svaki kupac dobije najjeftiniji od tri artikla za jedan dinar. Napisati program koji za unete cene tri artikla izračunava ukupnu cenu, kao i koliko dinara se uštedi zahvaljujući popustu. NAPOMENA: *Pretpostaviti da su cene artikala pozitivni celi brojevi.*

Primer 1

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

Primer 2

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

Primer 3

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

Primer 4

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

[Rešenje 2.6]

Zadatak 2.7 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
```

[Rešenje 2.7]

Zadatak 2.8 Napisati program koji za uneto vreme (broj sati iz intervala $[0, 24)$ i broj minuta iz intervala $[0, 60)$) ispisuje koliko je sati i minuta ostalo do ponoći.

Primer 1

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

Primer 2

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

Primer 3

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

Primer 4

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

[Rešenje 2.8]

Zadatak 2.9 Napisati program koji za učitani karakter ispisuje uneti karakter i njegov ASCII kod. Ukoliko je uneti karakter malo (veliko) slovo, ispisati i odgovarajuće veliko (malo) slovo i njegov ASCII kod.

Primer 1

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

Primer 2

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

Primer 3

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

Primer 4

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

[Rešenje 2.9]

Zadatak 2.10 Napisati program koji za unetih pet karaktera ispisuje koliko je među njima malih slova.

Primer 1

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

Primer 2

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

[Rešenje 2.10]

Zadatak 2.11 Program učitava pet karaktera. Napisati koliko se puta pojavilo veliko ili malo slovo a.

Primer 1

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

Primer 2

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

Primer 3

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

Primer 4

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

[Rešenje 2.11]

Zadatak 2.12 Program učitava pet karaktera. Ispisati koliko puta su se pojavile cifre.

Primer 1

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

Primer 2

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

Primer 3

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

Primer 4

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

[Rešenje 2.98]

Zadatak 2.13 Napisati program koji za unetu godinu ispisuje da li je prestupna.

Primer 1

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

Primer 2

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

Primer 3

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

Primer 4

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

[Rešenje 2.13]

Zadatak 2.14 Broj je Armstrongov ako je jednak zbiru kubova svojih cifara. Napisati program koji za dati trocifren broj proverava da li je Armstrongov.

Primer 1

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

Primer 2

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

Primer 3

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

Primer 4

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

[Rešenje 2.14]

Zadatak 2.15 Napisati program koji ispisuje proizvod parnih cifara unetog četvorocifrenog broja.

Primer 1

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

Primer 2

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

Primer 3

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

Primer 4

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

[Rešenje 2.15]

Zadatak 2.16 Napisati program koji učitava četvorocifreni broj i ispisuje broj koji se dobija kada se unetom broju razmene najmanja i najveća cifra. NAPOMENA: U slučaju da se najmanja ili najveća cifra pojavljuju na više pozicija, uzeti prvo pojavljivanje.

Primer 1

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

Primer 2

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

Primer 3

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

Primer 4

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

[Rešenje 2.16]

Zadatak 2.17 Napisati program koji ispituje da li se tačke $A(x_1, y_1)$ i $B(x_2, y_2)$ nalaze u istom kvadrantu i ispisuje odgovor DA ili NE.

[Rešenje 2.17]

Zadatak 2.18 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 2.19 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 2.20 Napisati program koji za uneti ceo broj x ispisuje njegov znak, tj da li je broj jednak nuli, manji od nule ili veći od nule.

Primer 1

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

Primer 2

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

Primer 3

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

Primer 4

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

[Rešenje 2.20]

2 Kontrola toka

Zadatak 2.21 Napisati program koji za unete koeficijente kvadratne jednačine ispisuje koliko realnih rešenja jednačina ima i ako ih ima, ispisuje ih zao-kružene na dve decimale.

Primer 1

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

Primer 2

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

[Rešenje 2.21]

Zadatak 2.22 Napisati program koji za uneti četvorocifreni broj prove-rava da li su njegove cifre uređene rastuće, opadajuće ili nisu uređene i štampa odgovarajuću poruku.

Primer 1

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

Primer 2

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

Primer 3

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

Primer 4

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesite četvorocifreni broj: 88  
|| Uneti broj nije četvorocifren.
```

[Rešenje 2.22]

Zadatak 2.23 Napisati program koji učitava karakter i:

- a) ako je *c* malo slovo, ispisuje odgovarajuće veliko
- b) ako je *c* veliko slovo, ispisuje odgovarajuće malo
- c) ako je *c* cifra, ispisuje poruku *cifra*
- d) u ostalim slučajevima, ispisuje karakter *c* između dve zvezdice.

Primer 1

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

Primer 2

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

Primer 3

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

Primer 4

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

[Rešenje 2.23]

Zadatak 2.24 U nizu 12345678910111213....9899 ispisani su redom brojevi od 1 do 99. Napisati program koji za uneti ceo broj k ($1 \leq k \leq 189$) ispisuje cifru koja se nalazi na k -toj poziciji datog niza.

Primer 1

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

Primer 2

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

[Rešenje 2.24]

Zadatak 2.25 Data je funkcija $f(x) = 2 \cdot \cos(x) - x^3$. Napisati program koji za učitane vrednosti realne promenljive x i vrednosti celobrojne promenljive k koje može biti 1, 2 ili 3 izračunava vrednost funkcije $F(k, x) = f(f(\dots f(x)))$ gde je funkcija f primenjena k -puta i ispisuje je zaokruženu na dve decimalne. U slučaju neispravnog ulaza, odštampati odgovarajuću poruku o grešci.

Primer 1

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

Primer 2

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

Primer 3

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

Primer 4

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

[Rešenje 2.25]

Zadatak 2.26 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
```

[Rešenje 2.26]

Zadatak 2.27 Napisati program koji za uneti karakter ispituje da li je samoglasnik.

Primer 1

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

Primer 2

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

Primer 3

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

Primer 4

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

[Rešenje 2.27]

Zadatak 2.28 Napisati program koji učitava dva cela broja i jedan od karaktera +, -, *, / ili % i ispisuje vrednost izraza dobijenog primenom date operacije na date argumente. U slučaju pogrešnog unosa ispisati odgovarajuću poruku.

Primer 1

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

Primer 2

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

Primer 3

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

Primer 4

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

[Rešenje 2.28]

Zadatak 2.29 Napisati program koji za uneti dan i mesec ispisuje godišnje doba kojem pripadaju. NAPOMENA: *Podrazumevati da je unos korektan.*

Primer 1

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

Primer 2

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

Primer 3

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

Primer 4

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

[Rešenje 2.29]

Zadatak 2.30 Napisati program koji za unetu godinu i mesec ispisuje naziv meseca kao i koliko dana ima u tom mesecu te godine.

[Rešenje 2.30]

Zadatak 2.31 Napisati program koji za uneti datum u formatu *dan.mesec.godina.* proverava da li je korektan.

Primer 1

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

Primer 2

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

[Rešenje 2.31]

Zadatak 2.32 Napisati program koji za korektno unet datum u formatu *dan.mesec.godina.* ispisuje datum prethodnog dana.

Primer 1

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

Primer 2

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

[Rešenje 2.32]

Zadatak 2.33 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 2.34 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 2.35** 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 unosi redom A, B, C, D , pri čemu $ABCD$ opisuje pravougaonik čije su stranice AB, BC, CD, DA , a dijagonale AC i BD . Na primer, tačke $(1, 1)$, $(2, 1)$, $(2, 2)$, $(1, 2)$ čine pravougaonik čije su stranice paralelne koordinatnim osama i čiji je obim 4 a tačke $(1, 1)$, $(2, 2)$, $(3, 3)$, $(4, 4)$ ne čine pravougaonik.
- ukoliko je uneti karakter h - proverava da li su unete tačke kolinearne i ukoliko jesu, ispisuje jednačinu prave kojoj pripadaju. Na primer, tačke $(1, 2)$, $(2, 3)$, $(3, 4)$, $(4, 5)$ su kolinearne i pripadaju pravoj $y = x + 1$, tačke $(1, 1)$, $(1, 2)$, $(1, 3)$, $(1, 4)$ su kolinearne i pripadaju pravoj $x = 1$, a tačke $(1, 1)$, $(2, 1)$, $(2, 2)$, $(1, 2)$ nisu kolinearne.

- ukoliko je uneti karakter j - Kramerovim pravilom proverava da li je sistem jednačina $x_1 * p + x_2 * q = x_4 - x_3$, $y_1 * p + y_2 * q = y_4 - y_3$ određen, neodređen ili nema rešenja, i u slučaju da je određen ispisuje rešenja.

[Rešenje 2.35]

Zadatak 2.36 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) ugrozava polje (m, n)
- c) da li konj sa (k, l) ugrozava polje (m, n)

2.2 Rešenja

Rešenje 2.1

```

1  #include <stdio.h>
3  int main()
4  {
5      int a,b,min;
6      printf("Unesite dva cela broja: ");
7      scanf("%d%d",&a,&b);
8
9      /* Promenljiva min dobija vrednost promenljive a. */
10     min=a;
11
12     /*
13      Ako je b<a, promenljiva min ce promeniti vrednost
14      tj. bice joj dodeljena vrednost promenljive b. U
15      suprotnom, vrednost ostaje ista.
16     */
17
18     if (b<a)
19         min = b;
20
21     printf("Minimum je %d\n",min);
22
23     return 0;
24 }
```

Rešenje 2.2 Rešenje je analogno rešenju broj 2.1.

Rešenje 2.3

```
1  #include<stdio.h>
3
5  int main()
6  {
7      float x;
8      float apsolutno_x;
9
10     printf("Unesite jedan realan broj:");
11     scanf("%f",&x);
12
13     apsolutno_x=x;
14     if (x<0)
15         apsolutno_x=-x;
16
17     printf("Njegova apsolutna vrednost je %.2f\n",apsolutno_x);
18
19     /*
20      2. nacin, pomocu funkcije fabs za koju je neophodno
21      ukljuciti zaglavlje math.h:
22      apsolutno_x=fabs(x);
23     */
24     return 0;
25 }
```

Rešenje 2.4

```
1  #include <stdio.h>
3
5  int main()
6  {
7      int x;
8      float rx;
9
10     printf("Unesite jedan ceo broj:");
11     scanf("%d",&x);
12
13     /*
14      Obratiti paznju:
15      x=0 - relacija jednakosti
16           (da li je promenljiva x jednaka nuli)
17      x=0 - naredba dodele
18           (promenljiva x dobija vrednost nula)
19     */
20 }
```

```

19  */
21  /*
22     Proveravamo da li je uneti broj jednak nuli. Ako
23     jeste, prekidamo sa daljim izvršavanjem programa
24     navodjenjem naredbe return. Argument -1 u naredbi
25     return oznacava da program nije uspesno završen
26  */
27  if (x==0)
28  {
29      printf("Nedozvoljeno deljenje nulom\n");
30      return -1;
31  }
32
33  /*
34     Primenom operatora / na argumente 1 i x dobijamo
35     rezultat celobrojnog deljenja ovih argumenata. Da
36     bismo dobili kolicnik, koji je realna vrednost,
37     neophodno je da jedan od argumenata zapisemo kao
38     realnu vrednost, npr celobrojnu vrednost 1 zapisemo kao
39     realnu vrednost 1.0. Ovakav postupak se naziva
40     implicitna konverzija.
41  */
42
43  rx = 1.0/x;
44  printf("Recipročna vrednost unetog broja: %.4f\n", rx);
45
46  return 0;
47  }

```

Rešenje 2.5

```

1  #include<stdio.h>
2
3  int main()
4  {
5      int a,b,c;
6      int s;
7      printf("Unesite tri cela broja:");
8      scanf("%d%d%d",&a,&b,&c);
9
10     /* inicijalizujemo promenljivu s na nulu */
11     s=0;
12
13     /*
14     U naredbi dodele s=s+a vrednost izraza sa desne strane
15     znaka jednakosti dodeljujemo promenljivoj sa leve
16     strane znaka jednakosti. Staru vrednost promenljive s
17     saberemo sa vrednoscu promenljive a i dobijenu vrednost
18     upisemo u promenljivu s.
19     */

```

```
21  */
22
23  if (a>0)
24      s=s+a;
25
26  /* s+=b je skraceni zapis za s=s+b */
27
28  if (b>0)
29      s+=b;
30
31  if (c>0)
32      s+=c;
33
34  printf("Suma unetih pozitivnih brojeva: %d\n",s);
35  return 0;
36 }
```

Rešenje 2.6

```
2  #include <stdio.h>
3  int main()
4  {
5
6      unsigned a,b,c;
7      unsigned min;
8      unsigned cena_bez_popusta, cena_sa_popustom;
9
10     printf("Unesite cene tri artikla:");
11     scanf("%u%u%u",&a,&b,&c);
12
13     /*
14      Racunamo minimum tri broja. Dodeljujemo promenljivoj
15      min vrednost prvog broja.
16     */
17     min=a;
18
19     /*
20      Ako je drugi broj manji od minimuma, to znaci da
21      promenljiva min ne sadrzi najmanji broj. Dodeljujemo
22      joj vrednost drugog broja.
23     */
24     if(min>b)
25         min=b;
26
27     /*
28      Ako je treci broj manji od minimuma, to znaci da
29      promenljiva min ne sadrzi najmanji broj. Dodeljujemo
30      joj vrednost treceg broja.
31     */
32     if(min>c)
33         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 2.7

```

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

```

```

    max=j;
41
    if(d>max)
43        max=d;

    if(s>max)
45        max=s;

47
    if(h>max)
49        max=h;

51    /* II nacin:
    * if(j>d && j>s && j>h)
53    *     max=j;
    * if(d>j && d>s && d>h)
55    *     max=d;
    * if(s>j && s>d && s>h)
57    *     max=s;
    * if(h>j && h>d && h>s)
59    *     max=h;
    */

61    /* Ispisujemo rezultat */
63    printf("Najveca cifra je: %d\n", max);

65    return 0;
}
```

Rešenje 2.8

```

2  #include<stdio.h>

4  int main()
{
6      int sati;
      int minuti;
8      int preostali_sati;
      int preostali_minuti;

10

12     /*
        Ukoliko naredbu printf zelimo da napisemo u dva
        reda, i tom prilikom prekidamo deo pod navodnicima,
14     to mozemo uraditi navodjenjem navodnika na kraju
        prvog i na pocetku narednog reda:
16     */

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

```

22  /*
23     U slucaju da je unos neispravan, ispisujemo
24     poruku o gresci i prekidamo dalje izvršavanje
25     programa.
26
27     Uslov u if naredbi je disjunkcija (operator ||)
28     sastavljena od 4 disjunkata. Svaki od njih je
29     izraz sa relacionim operatorom i ima vrednost
30     1 ako je izraz tacan i 0 u suprotnom. Da bi
31     disjunkcija bila tacna, bar jedan od disjunkata
32     mora da bude tacan. Zbog lenjog izracunavanja,
33     vrednost disjunkata ce biti racunata do vrednosti
34     prvog disjunkta koji je tacan. To je znak da
35     je uslov u if naredbi ispunjen i nema potrebe
36     racunati vrednosti drugih disjunkata.
37
38  */
39
40  if (sati>24 || sati<0 || minuti>59 || minuti<0)
41  {
42      printf("Neispravan unos.\n");
43      return -1;
44  }
45
46  preostali_sati = 24-sati-1;
47  preostali_minuti = 60-minuti;
48
49  if (preostali_minuti==60)
50  {
51      preostali_sati++;
52      preostali_minuti=0;
53  }
54
55  /*
56     Ukoliko naredbu printf zelimo da napisemo u dva
57     reda i pritom ne prekidamo deo pod navodnicima,
58     to mozemo uraditi bez navodjenja dodatnih karaktera:
59
60  */
61  printf("Do ponoci je ostalo %d sati i %d minuta\n",
62        preostali_sati, preostali_minuti);
63  return 0;
64  }

```

Rešenje 2.9

```

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

```

2 Kontrola toka

```
8     printf("Unesite jedan karakter:");
    scanf("%c", &c);

10     printf("Uneti karakter: %c, njegov ASCII kod: %d\n", c, c);

12     /*
14      Razlika izmedju ASCII koda svakog malog i odgovarajuceg
      velikog slova je konstanta koja se moze sracunati
      izrazom 'a'-'A' (i iznosi 32)
16     */

18     if (c>='A' && c<='Z')
        printf("odgovarajuće malo slovo: %c, njegov ASCII kod: ",
20              "%d\n", c+('a'-'A'), c+('a'-'A'));

22     if (c>='a' && c<='z')
        printf("odgovarajuće veliko slovo: %c, njegov ASCII kod: ",
24              "%d\n", c-('a'-'A'), c-('a'-'A'));

26     return 0;
}
```

Rešenje 2.10

```
1
#include <stdio.h>

3
int main(){

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

7

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

13

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

19

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

25

27    /* Proveravamo da li je treci karakter malo slovo */
    if(c3>='a' && c3<='z'){
```



```

29     /* I ako jeste, uvecavamo broj malih slova */
    broj_malih_slova++;
}

31

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

37

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

43

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

47

49     return 0;
}

```

Rešenje 2.11

```

1  #include <stdio.h>
   #include <ctype.h>
3
   int main()
5  {
7      /*
        Broj pojavljivanja slova a i A se
        inicijalizuje na 0
9      */
        int br_a = 0;

11     /*
13     Funkcija getchar ucitava jedan karakter.
        Njena povratna vrednost je ASCII kod
15     ucitanog karaktera.

17     Funkcija tolower za dati karakter vraca:
        - odgovarajuće malo slovo, ako je dati
19         karakter veliko slovo
        - taj isti karakter, u suprotnom
21     Ova funkcija je definisana u biblioteci ctype.h

23     U slucaju da je uslov ispunjen, uvecavamo
        brojac br_a za jedan pomocu operatora
25     inkrementacije ++
        */

```

2 Kontrola toka

```
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
38     printf("%d\n", br_a);
39
40     return 0;
41 }
```

Rešenje 2.98

```
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 2.13

```
1
```

```
1 #include <stdio.h>
3
5 int main()
6 {
7     int x;
8     printf("Unesite godinu:");
9     scanf("%d",&x);
10
11     if ((x%4==0 && x%100!=0) || x%400==0)
12         printf("Godina je prestupna\n");
13     else
14         printf("Godina nije prestupna\n");
15
16     return 0;
17 }
```

Rešenje 2.14

```
1 #include <stdio.h>
2 #include <stdlib.h> /*abs*/
3
5 int main()
6 {
7     int n, j, d, s;
8
9     /* Ucitavamo broj */
10    printf("Unesite broj: ");
11    scanf("%d", &n);
12
13    /*
14     * Uzimamo apsolutnu vrednost broja za slucaj da je
15     * uneti broj negativan
16     */
17    n = abs(n);
18
19    /*
20     * Ako broj nije trocifren, izdajemo poruku o gresci i
21     * prekidamo dalje izvršavanje programa
22     */
23    if(n<100 || n>999)
24    {
25        printf("Greska: Niste uneli trocifren broj!\n");
26        return -1;
27    }
28
29    /*
30     * Izdvajamo cifre broja:
31     * j -jedinice, d - desetice, s - stotine
32     */
33    j=n%10;
```

```
35     d=(n/10)%10;
36     s=n/100;
37
38     /* Proveravamo da li je broj Armstrongov */
39     if(n==j*j*j+d*d*d+s*s*s)
40         printf("Broj je Armstrongov.\n");
41     else
42         printf("Broj nije Armstrongov.\n");
43
44     return 0;
45 }
```

Rešenje 2.15

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

```

37     proizvod_parnih=proizvod_parnih*j;
    broj_parnih++;
39 }

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

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

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

57 /* Proveravamo da li u zapisu broja ima parnih cifara i
    ispisujemo
59     rezultat */
    if(broj_parnih==0){
61     printf("Nema parnih cifara.\n");
    }
63 else{
    printf("Proizvod parnih cifara: %d\n", proizvod_parnih);
65 }

67

69     return 0;
71 }

```

Rešenje 2.16

```

1  #include <stdio.h>

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

7

        // Da bismo lakse odredili da li je cetvorocifren
9        int absBroj = broj < 0 ? -broj : broj;
        if ( absBroj <= 999 || absBroj >= 10000)
11     {

```

```
13     printf("Broj nije cetvorocifren.");
14     return -1;
15 }
16
17 int a = absBroj % 10;
18 int b = (absBroj / 10) % 10;
19 int c = (absBroj / 100) % 10;
20 int d = absBroj / 1000;
21
22 int max = a, min = a;
23 // cuvamo i stepen da bismo lakse zamenili cifre
24 /* Ideja:
25    4179, mesta menjamo tako sto oduzmemo 9 i dodamo 1,
26    ^ ^  odnosno oduzemo 100 i dodamo 900 */
27 int stepenMax = 1, stepenMin = 1;
28
29 if (b > max)
30 {
31     max = b;
32     stepenMax = 10;
33 }
34 if (b < min)
35 {
36     min = b;
37     stepenMin = 10;
38 }
39
40 if (c > max)
41 {
42     max = c;
43     stepenMax = 100;
44 }
45 if (c < min)
46 {
47     min = c;
48     stepenMin = 100;
49 }
50
51 if (d > max)
52 {
53     max = d;
54     stepenMax = 1000;
55 }
56 if (d < min)
57 {
58     min = d;
59     stepenMin = 1000;
60 }
61
62 int rez;
63 /* Ideja:
```

```

65     4179, mesta menjamo tako sto oduzemo 9 i dodamo 1,
        ~ ~ odnosno oduzemo 100 i dodamo 900 */
67     if (broj > 0)
        rez = broj - max*stepenMax + min*stepenMax
69             - min*stepenMin + max*stepenMin;
        else
71         rez = broj + max*stepenMax - min*stepenMax
                + min*stepenMin - max*stepenMin;
73
        printf("%d\n", rez);
75     return 0;
77 }

```

Rešenje 2.20

```

1  #include <stdio.h>
3  int main()
4  {
5      int x;
        printf("Unesite jedan ceo broj:");
7      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 2.21

```

1  #include <stdio.h>
3  #include <math.h>
4  int main()
5  {
6      float a,b,c;
7      float D;
8      float x1,x2;
9      printf("Unesite koeficijente A, B i C:");
10     scanf("%f%f%f",&a,&b,&c);
11
        /*

```

```
13         Proveravamo da li je kvadratna jednacina
           korektno zadata.
15     */
16     if (a==0)
17     {
18         if (b==0)
19         {
20             /* slucaj a==0 && b==0 && c==0 */
21             if(c==0)
22                 printf("Jednacina ima beskonacno mnogo resenja\n");
23             /* slucaj a==0 && b==0 && c!=0 */
24             else
25                 printf("Jednacina nema resenja\n");
26         }
27         /* slucaj a==0 && b!=0 */
28         else
29         {
30             x1=-c/b;
31             printf("Jednacina ima jedinstveno realno resenje %.2f\n",
32                   x1);
33         }
34     }
35     /* slucaj a!=0 */
36     else
37     {
38         D=b*b-4*a*c;
39         if (D<0)
40             printf("Jednacina nema realnih resenja\n");
41         else if (D>0)
42         {
43             /* funkcija sqrt nalazi se u biblioteci math.h
44              (prevodjenje sa -lm opcijom) */
45             x1 = (-b+sqrt(D))/(2*a);
46             x2 = (-b-sqrt(D))/(2*a);
47             printf("Jednacina ima dva razlicita realna resenja %.2f ",
48                   "i %.2f\n",x1,x2);
49         }
50         else
51         {
52             x1 = (-b)/(2*a);
53             printf("Jednacina ima jedinstveno realno resenje %.2f\n",x1);
54         }
55     }
56     return 0;
57 }
```

Rešenje 2.22

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

4 int main()
5 {
```



```

6   int x;
   char c1;
8   char c10;
   char c100;
10  char c1000;

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

14

16  /*
   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
18  */
   x=abs(x);
20

22  if (x<1000 || x>9999)
   {
24     printf("Uneti broj nije cetvorocifren\n");
     return -1;
26  }

28  /* Izdvajamo cifre broja. */
   c1 = x%10;
30  c10 = (x/10)%10;
   c100 = (x/100)%10;
32  c1000 = (x/1000)%10;

34  if (c1000<=c100 && c100<=c10 && c10<=c1)
     printf("Cifre su uredjene neopadajuce \n");
36  else if (c1000>=c100 && c100>=c10 && c10>=c1)
     printf("Cifre su uredjene nerastuce \n");
38  else
     printf("Cifre nisu uredjene\n");
40  }

```

Rešenje 2.23

```

1   #include <stdio.h>

3   int main(){

5       char c;

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

11      if(c>='a' && c<='z')
         printf("%c\n", c-'a'+ 'A');

```

2 Kontrola toka

```
13     else if(c>='A' && c<='Z')
14         printf("%c\n", c-'A'+'a');
15     else if(c>='0' && c<='9')
16         printf("cifra\n");
17     /*
18      * Ako nijedan od prethodnih uslova nije ispunjen,
19      * bice izvršena naredba u else grani
20      */
21     else
22         printf("%c*\n",c);
23
24     return 0;
25 }
```

Rešenje 2.24

```
1  #include <stdio.h>
2
3
4
5  int main(){
6      int k, n, broj;
7
8      printf("Unesite k: ");
9      scanf("%d", &k);
10
11     if(k<10){
12         /* Trazi se jednocifren broj */
13         printf("Na %d-toj poziciji je broj %d.\n", k, k);
14     }
15     else
16         /* Trazi se dvocifreni broj */
17         if(k>=10 && k<=189){
18
19             /* Odredjujemo broj dvocifrenih brojeva koji se mogu
20              * zapisati pomocu k cifara */
21
22             if(k%2!=0){
23                 /*
24                  * Ako je k neparan broj, zapisan je ceo broj
25                  * dvocifrenih brojeva
26
27                  * 9 oduzimamo jer je 9 broj cifara potrebnih za
28                  * zapis jednocifrenih brojeva
29                  */
30                 n=(k-9)/2;
31
32                 /* Broj o kojem se radi je */
33                 broj=9+n;
34
35                 /* Ujedno, za neparno k se trazi cifra jedinica
```

```

37         izdvojenog broja */
        printf("Na %d-toj poziciji je broj %d.\n", k, broj%10);
39     }
    else{
41         /* Ako je k paran broj, zapisan je ceo broj
           dvocifrenih brojeva i zapoceto je sa zapisom sledeceg
43
           9 oduzimamo jer je 9 broj cifara potrebnih za zapis
45         jednocifrenih brojeva
           */
47         n=(k-9)/2 +1;

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

51         /* Ujedno, za parno k se trazi cifra desetica
           izdvojenog broja */
53         printf("Na %d-toj poziciji je broj %d.\n", k, broj/10);
55     }
57 }
    else{
59     printf("Greska: Nedozvoljena vrednost broja k!\n");
61 }
63 return 0;
64 }

```

Rešenje 2.25

```

1  #include <stdio.h>
3  #include <math.h>
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){

```

2 Kontrola toka

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

Rešenje 2.26

```
1
#include <stdio.h>
3
int main(){
5
    int dan;
7
    printf("Unesite broj: ");
9    scanf("%d", &dan);
11
    switch(dan)
    {
13        case 1:
            printf("ponedeljak\n");
15            break;
        case 2:
17            printf("utorak\n");
            break;
19        case 3:
            printf("sreda\n");
21            break;
        case 4:
23            printf("cetvrtak\n");
            break;
25        case 5:
            printf("petak\n");
27            break;
        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
38 return 0;
39 }
```

Rešenje 2.27

```
1  #include <stdio.h>
2
3  int main()
4  {
5      char c;
6      printf("Unesite jedan karakter:");
7      scanf("%c", &c);
8
9      /*
10     Da bismo utvrditi da li je karakter samoglasnik,
11     neophodno je proveriti da li odgovara nekom od
12     sledecih karaktera: A,E,I,O,U,a,e,i,o,u
13
14     */
15
16     switch(c)
17     {
18         case 'A' :
19         case 'E' :
20         case 'I' :
21         case 'O' :
22         case 'U' :
23         case 'a' :
24         case 'e' :
25         case 'i' :
26         case 'o' :
27         case 'u' : printf("Uneti karakter je samoglasnik\n");
28                     break;
29         default : printf("Uneti karakter nije samoglasnik\n");
30                     break;
31     }
32
33     return 0;
34 }
```

Rešenje 2.28

```
1  #include <stdio.h>
3
5  int main()
6  {
7      char op;
8      int x, y;
9
10     printf("Unesite operator i dva cela broja: ");
11     scanf("%c %d %d", &op, &x, &y);
12
13     switch(op){
14         case '+':
15             printf("Rezultat je: %d\n", x+y);
16             break;
17         case '-':
18             printf("Rezultat je: %d\n", x-y);
19             break;
20         case '*':
21             printf("Rezultat je: %d\n", x*y);
22             break;
23         case '/':
24             if(y==0)
25                 printf("Greska: deljenje nulom nije dozvoljeno!\n");
26             else
27                 printf("Rezultat je: %f\n", x*1.0/y);
28             break;
29         case '%':
30             printf("Rezultat je: %d\n", x%y);
31             break;
32         default:
33             printf("Greska: nepoznat operator!\n");
34     }
35
36     return 0;
37 }
```

Rešenje 2.29

```
1  #include <stdio.h>
3
5  int main()
6  {
7      int d,m;
8      printf("Unesite dan i mesec");
9      scanf("%d%d",&d,&m);
```

```
9
11  /*
13     Argument u naredbi switch mora biti celobrojna promenljiva,
        dok argument u naredbi case mora biti celobrojna
        konstanta.
15  */
16  switch(m)
17  {
18      /* Ispitujemo da li vazi m==1 ili m==2*/
19      case 1:
20      case 2:
21          printf("zima\n");
22          break;
23      case 3:
24          if (d<21)
25              printf("zima\n");
26          else
27              printf("prolece\n");
28          break;
29      case 4:
30      case 5:
31          printf("prolece\n");
32          break;
33      case 6:
34          if (d<21)
35              printf("prolece\n");
36          else
37              printf("leto\n");
38          break;
39      case 7:
40      case 8:
41          printf("leto\n");
42          break;
43      case 9:
44          if (d<23)
45              printf("leto\n");
46          else
47              printf("jesen\n");
48          break;
49      case 10:
50      case 11:
51          printf("jesen\n");
52          break;
53      case 12:
54          if (d<22)
55              printf("jesen\n");
56          else
57              printf("zima\n");
58  }
59  return 0;
}
```

Rešenje 2.30

```
1 #include <stdio.h>
2 #include <stdlib.h> // Potrebno za exit funkciju
3
4
5 int main()
6 {
7     int godina;
8     int mesec;
9
10    int prestupna;
11
12
13    printf("Unesite godinu: ");
14    scanf("%d",&godina);
15
16    if(godina < 0)
17    {
18        printf("Lose uneta godina!\n");
19        return -1;
20    }
21
22    /* Provera da li je godina prestupna, zbog februara */
23    if((godina % 4 == 0 && godina % 100 != 0)
24        || godina % 400 == 0)
25        prestupna = 1;
26    else
27        prestupna = 0;
28
29    printf("Unesite redni broj meseca: ");
30    scanf("%d",&mesec);
31
32    switch(mesec)
33    {
34        case 1: printf("Januar, 31 dan\n");
35                break;
36        case 2:
37            if(prestupna)
38                printf("Februar, 29 dana\n");
39            else
40                printf("Februar, 28 dana\n");
41            break;
42
43        case 3: printf("Mart, 31 dan\n");
44                break;
45        case 4: printf("April, 30 dana\n");
46                break;
47        case 5: printf("Maj, 31 dan\n");
48                break;
49        case 6: printf("Jun, 30 dana\n");
50                break;
```



```

51     case 7: printf("Jul, 31 dan\n");
           break;
53     case 8: printf("Avgust, 31 dan\n");
           break;
55     case 9: printf("Septembar, 30 dana\n");
           break;
57     case 10: printf("Oktobar, 31 dan\n");
           break;
59     case 11: printf("Novembar, 30 dana\n");
           break;
61     case 12: printf("Decembar, 31 dan\n");
           break;
63     default: printf("Lose unet redni broj meseca!\n");
}
65
67 return 0;
}

```

Rešenje 2.31

```

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

```

2 Kontrola toka

```
33      /* Dozvoljeni broj dana za januar, mart, maj, jul, avgust,
34       * oktobar i decembar je 31 */
35      dozvoljen_broj_dana=31;
36      break;
37  case 2:
38      /* Proveravamo da li je godina prestupna */
39      if(godina%4==0 && godina%100!=0 || godina%400==0)
40          /* Ako jeste, dozvoljeni broj dana za februar je 29 */
41          dozvoljen_broj_dana=29;
42      else
43          /* Ako nije, dozvoljeni broj dana za februar je 28 */
44          dozvoljen_broj_dana=28;
45      break;
46  case 4:
47  case 6:
48  case 9:
49  case 11:
50      /* Dozvoljeni broj dana za april, jun, septembar i
51       * novembar je 30 */
52      dozvoljen_broj_dana=30;
53      break;
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;
66 }
```

Rešenje 2.32

```
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;
```

```
15     prethodni_mesec=mесec;
16     prethodni_godina=година;
17
18     /* I po potrebi vrsimo korekcije */
19
20     /* Ako je u pitanju prvi u mesecu */
21     if(prethodni_dan==0){
22         /* Treba korigovati mesec */
23         prethodni_mesec=mесec-1;
24         /* Ako je u pitanju januar */
25         if(prethodni_mesec==0){
26             /* Treba korigovati i godinu */
27             prethodni_mesec=12;
28             prethodni_godina=година-1;
29         }
30
31         /* Analiziramo redni broj meseca kako bi odredili
32            tacan dan*/
33         switch(prethodni_mesec){
34             case 1:
35             case 3:
36             case 5:
37             case 7:
38             case 8:
39             case 10:
40             case 12:
41                 prethodni_dan=31;
42                 break;
43             case 2:
44                 if((prethodni_godina%4==0 && prethodni_godina%100!=0)
45                    || prethodni_godina%400==0)
46                     prethodni_dan=29;
47                 else
48                     prethodni_dan=28;
49                 break;
50             case 4:
51             case 6:
52             case 9:
53             case 11:
54                 prethodni_dan=30;
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 2.35

```

2  #include<stdio.h>
   #include<math.h>
4  int main()
   {
6      char c;
      float x1,y1,x2,y2,x3,y3,x4,y4;
8      float kab,kbc,kad;
      float dab,dad;
10     float delta, deltap, deltaq;
      float 0;
12     float k,n;

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

16

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

20     switch (c)
     {
22         case 'k':
             if (y1==y2 && y3==y4 && x1==x4 && x2==x3)
             {
24                 dab = sqrt(pow(x1-x2,2)+pow(y1-y2,2));
26                 dad = sqrt(pow(x1-x4,2)+pow(y1-y4,2));
                 0 = 2*dab + 2*dad;
28                 printf("Obim pravougaonika je %f\n",0);
             }
30             else
                 printf("Tacke ne cine pravougaonik sa stranicama ",
32                     "koje su paralelne koordinatnim osama\n");
                 break;
34         case 'h':
             /*
36             Ukoliko se tacke A(x1,y1) i B(x2,y2) ne nalaze
             na pravoj koja je paralelna x osi, izracunamo k,n
38             za pravu odredjenu tackama A(x1,y1) i B(x2,y2)
             */
40             if ((x1-x2)!=0)
             {
42                 k = (y1-y2)/(x1-x2);
44                 n = y1-k*x1;
                 /*
46                 Proverimo da li tacke C(x3,y3) i D(x4,y4) nalaze
                 na toj pravoj
                 */
48                 if (y3==x3*k+n && y4==x4*k+n)
                     printf("Tacke su kolinearne, pripadaju pravoj ",
50                         "y=%f*x+%f\n",k,n);
                     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 return 0;
82 }

```

2.3 Petlje

2.3.1 Ispis podataka

REDOSLED: Petlje se sustinski koriste za tri stvari: map, filter i reduce, kao i za kombinaciju te tri stvari.

Map — preslikavanje, dakle ceo niz necega se preslikava na neki nacin u neki novi niz (dupliranje vrednosti svih elemenata niza, dupliranje svake cifre broja, dodavanje prefiksa svim recima...)

Filter — iz niza necega biraju se neki koji zadovoljavaju neki kriterijum (svi parni brojevi, svi koji sadrze karakter "a", svi prosti brojevi, svi savrseni brojevi...)

Reduce — ceo niz se svodi na jednu vrednost (zbir svih vrednost, proizvod svih vrednosti, nadovezane sve vrednosti...)

Kombinacija — dve tehnike od prethodne tri (npr filter-reduce: zbir svih parnih brojeva) ili od svake po malo (zbir svih dupliranih brojeva koji su savrseni)

Sustinski, studenti treba da usvoje najpre ove tri tehnike, pa onda da idu ne

njihove kombinacije, i to najpre na kombinacije dve od tri, pa na kraju na zadatke koje kombinuju sve to. Ove tehnike nisu vezane za nizove, mogu se primeniti i na prirodne brojeve posmatrane kao niz brojeva ili na prirodni broj posmatran kao niz cifara...

Danijela:: ok neka bude redosled koji si predložila

Zadatak 2.37 Napisati program koji 5 puta ispisuje tekst *Mi volimo da programiramo.*

Primer 1

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

[Rešenje 2.37]

Zadatak 2.38 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 pozitivan broj 0
```

[Rešenje 2.38]

Zadatak 2.39 Napisati program koji učitava pozitivan ceo broj n a potom ispisuje sve cele brojeve od 0 do n .

Primer 1

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

Primer 2

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

[Rešenje 2.39]

Zadatak 2.40 Napisati program koji učitava dva cela broja n i m ispisuje sve cele brojeve iz intervala $[n, m]$.

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

Primer 1

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

Primer 2

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

[Rešenje 2.40]

Zadatak 2.41 Uskladiti formulaciju zadatka sa odgovarajućom formulacijom kod nizova. Fibonačijev niz počinje ciframa 1 i 1, a svaki član se dobija zbirom prethodna dva. Napisati program koji učitava ceo neoznačen broj n i određuje i na standardni izlaz ispisuje n -ti član Fibonačijevog niza. Milena: Ovde bih dodala jos dva normalna test primera.

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!
```

[Rešenje 2.41]

* **Zadatak 2.42** 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.

Milena: Ovde bih dodala jos dva normalna test primera.

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.
```

[Rešenje 2.42]

* **Zadatak 2.43** 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 neoznačen broj k ispisuje dimenzije papira A_k u milimetrima. Rezultat ispisati kao celobrojne vrednosti.

Primer 1

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesite broj n: 4  
|| 210 297
```

Primer 2

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesite broj n: 3  
|| 297 420
```

Primer 3

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesite broj n: 7  
|| 74 105
```

Primer 4

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesite broj n: 9  
|| 37 52
```

[Rešenje 2.43]

Milena: Komentari u resenjima su uglavnom ok. Jedino smo rekli da stil treba da bude bezlican, tj da umesto "ucitavamo broj"ili "transformisemo vrednost"kazemo "ucitava se broj"ili "vrednost se transformise". Ima i jedne i druge vrste komentara, cak i u istom resenju, i to treba ujednaciti.

2.3.2 Obrada celih brojeva, rad sa ciframa broja

Zadatak 2.44 Napisati program koji učitava ceo broj i ispisuje njegove cifre u obrnutom poretaku.

Primer 1

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

Primer 2

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

[Rešenje 2.44]

Zadatak 2.45 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.

Milena: Svuda gde postoji ispravan i neispravan unos, staviti da bude 3 ili 4 test primera, tako da se veci znacaj daje ispravnom unosu. Ovako kada je 1-1 de-

luje kao da su podjednako vazni, a ipak vecu tezinu treba dati ispravnim unosima.

Primer 1

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

Primer 2

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

[Rešenje 2.45]

Zadatak 2.46 Napisati program koji za pozitivan ceo broj proverava i ispisuje da li se cifra 5 nalazi u njegovom zapisu.

Milena: Da li cemo koristiti tip unsigned u resenju u svim zadacima u kojima se trazi pozitivan ceo broj ili cemo stavljati test primere za negativan broj? Ja bih ovde dozvolila da je i pozitivan i negativan broj, samo treba prilagoditi resenje.

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!
```

[Rešenje 2.46]

Zadatak 2.47 Napisati program koji za uneti prirodan broj ispisuje da li je on deljiv sumom svojih cifara.

Milena: Sta ako se unese broj 0? — Mozda staviti test primer na tu temu.

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.
```

[Rešenje 2.47]

Zadatak 2.48 Napisati program koji za uneti prirodan broj uklanja sve nule sa njegove desne strane. Novodobijeni broj ispisati na standardni izlaz.

Primer 1

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

Primer 2

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

Primer 3

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

[Rešenje 2.48]

Zadatak 2.49 Napisati program koji uneti prirodan broj transformiše tako što svaku parnu cifru u zapisu broja uveća za 1. Novodobijeni broj ispisati na standardni izlaz.

Primer 1

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

Primer 2

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

Primer 3

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

[Rešenje 2.49]

Zadatak 2.50 Milena: Neoznaceni ceo broj je termin iz programiranja. U zadacima treba da budu prisutni celi i prirodni brojevi. Zato sam svuda to menjala. Takodje, treba izbegavati formulacije koje sugerisu redosled programiranja i naredbi u programu. Dakle vise idemo na formulacije sta treba da se uradi, nego kako to treba da se uradi.

Napisati program koji formira i ispisuje broj koji se dobija izbacivanjem svake druge cifre polaznog prirodnog 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
```

[Rešenje 2.50]

* **Zadatak 2.51** 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: 242
|| 22
```

[Rešenje 2.51]

* **Zadatak 2.52** Broj je *palindrom* ukoliko se isto čita i sa leve i sa desne strane. Napisati program koji učitava prirodan broj i proverava da li je učitani broj palindrom.

Primer 1

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite broj: 25452
|| Broj je palindrom!
```

Primer 2

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite broj: 895
|| Broj nije palindrom!
```

Primer 3

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite broj: 5
|| Broj je palindrom!
```

[Rešenje 2.52]

2.3.3 Unos i obrada veće količine podatka (**unos i obrada niza brojeva?, nije sjajno zbog nizova**)

Zadatak 2.53 Napisati program koji učitava pozitivan ceo broj n , a zatim učitava n celih brojeva i na standardni izlaz ispisuje sumu pozitivnih i sumu negativnih unetih brojeva.

Primer 1

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

Primer 2

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

Primer 3

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

[Rešenje 2.53]

Zadatak 2.54 Program unosi ceo pozitivan broj n , a potom i n celih brojeva. Izračunati i ispisati zbir onih brojeva koji su neparni i negativni.

Primer 1

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

Primer 2

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

Primer 3

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

[Rešenje 2.54]

Zadatak 2.55 Napisati program koji učitava cele brojeve sve dok se ne unese nula. Nakon toga ispisati proizvod onih unetih brojeva koji su pozitivni.

Primer 1

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

Primer 2

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

Primer 2

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

[Rešenje 2.55]

Zadatak 2.56 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 na standardni izlaz 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.
```

[Rešenje 2.56]

Zadatak 2.57 Sa standardnog ulaza se unose realni brojevi sve do unosa broja nula 0. Napisati program koji izračunava i ispisuje aritmetičku sredinu unetih brojeva.

Primer 1

```
INTERAKCIJA SA PROGRAMOM:
Unesite brojeve: 8 5.2 6.11 3 0
Aritmetička sredina: 5.5775
```

Primer 2

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

[Rešenje 2.57]

Zadatak 2.58 U prodavnici se nalaze artikala čije cene su realni pozitivni brojevi. Cene artikala se unose sa standardnog unosa 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
|| Aritmeticka sredina: 5.5775
```

Primer 2

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite cene: 6.32 -9 8 0
|| Cena ne moze biti negativana.
```

Primer 3

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

[Rešenje 2.58]

Zadatak 2.59 Nikola želi da obraduje baku i da joj kupi jedan poklon u radnji. On na raspolaganju ima m novaca. U radnji se nalazi n artikala i zanima ga koliko ima artikala u radnji čija cena je manja ili jednaka m . Napisati program koji pomaže Nikoli da brzo odrediti broj atikala. 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 moze biti negativan.
```

Primer 4

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

[Rešenje 2.59]

Zadatak 2.60 Sa standardnog ulaza unosi se 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.

2 Kontrola toka

Primer 1

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

Primer 2

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

Primer 3

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

Primer 4

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

[Rešenje 2.98]

Zadatak 2.61 Sa standardnog ulaza unosi se ceo pozitivan broj n , a potom n realnih brojeva. Odrediti koliko puta je prilikom unosa došlo do promene znaka. Ispisati dobijenu vrednost na standardni izlaz.

Primer 1

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite broj n: 10
|| Unesite brojeve:
|| 7.82 4.3 -1.2 56.8 -3.4 -72.1 8.9 11.2 -11.2
|| -102.4
|| Broj promena je 5.
```

Primer 2

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

Primer 3

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

Primer 4

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

[Rešenje 2.98]

Zadatak 2.62 Sa standardnog ulaza se unosi ceo pozitivan broj n , a zatim i n celih brojeva. Napisati program koji ispisuje broj sa najvećom cifrom desetica. Ukoliko ima više takvih, ispisati prvi.

Primer 1

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

Primer 2

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

Primer 3

```

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

```

[Rešenje 2.62]

Zadatak 2.63 Sa standardnog ulaza se unosi ceo pozitivan broj n , a zatim i n celih brojeva. Napisati program koji ispisuje broj sa najvećim brojem cifara. Ukoliko ima više takvih, ispisati prvi.

Primer 1

```

|| INTERAKCIJA SA PROGRAMOM:
|| Unesite broj n: 5
|| Unesite n brojeva: 18 365 25 1 78
|| Najviše cifara ima broj 365.

```

Primer 2

```

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

```

[Rešenje 2.63]

Zadatak 2.64 Sa standardnog ulaza se unosi 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.

```

[Rešenje 2.64]

Zadatak 2.65 Sa standardnog ulaza se unose celi pozitivni brojevi 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 $d(x, y) = |y - x|$. Rezultat ispisati na standardni izlaz.

Primer 1

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

Primer 2

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

[Rešenje 2.65]

Zadatak 2.66 Vršna su merenja nadmorskih visina na određenom delu teritorije i naučnike zanima razlika između najveće i najmanje nadmorske visine. Napisati program koji učitava realne brojeve sve do unosa 0 koji označavaju nadmorske visine i ispisuje razliku najveće i najmanje nadmorske visine.

Primer 1

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

Primer 2

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

[Rešenje 2.66]

2.3.4 Rad sa karakterima

Zadatak 2.67 Napisati program koji učitava karaktere dok se ne unese karakter tačka, i ako je karakter malo slovo ispisuje odgovarajuće veliko, ako je karakter veliko slovo ispisuje odgovarajuće malo, a u suprotnom ispisuje isti karakter kao i uneti.

Primer 1

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

Primer 2

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

[Rešenje 2.67]

Zadatak 2.68 Napisati program koji učitava karaktere sve do kraja ulaza, a potom ispisuje broj velikih slova, broj malih slova, broj cifara, broj belina i zbir unetih cifara.

Primer 1

```

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

```

Primer 2

```

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

```

[Rešenje 2.68]

Zadatak 2.69 Sa standardnog ulaza unosi se ceo pozitivan broj n , a potom i n karaktera. Za svaki od samoglasnika ispisati koliko puta se pojavio među unetim karakterima. Ne praviti razliku između malih i velikih slova.

Primer 1

```

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

```

Primer 2

```

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

```

[Rešenje 2.69]

Zadatak 2.70 Sa standardnog ulaza se unosi ceo broj n , a zatim i n karaktera. Napisati program koji proverava da li se od unetih karaktera može napisati reč *Zima*.

Primer 1

```

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

```

Primer 2

```

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

```

[Rešenje 2.70]

2.3.5 Računanje sume i proizvoda

Zadatak 2.71 Napisati program koji učitava ceo pozitivan broj i izračunava njegov faktorijel. U slučaju neispravnog unosa ispisati odgovarajuću poruku.

Primer 1

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

Primer 2

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

Primer 3

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

[Rešenje 2.71]

Zadatak 2.72 Sa standardnog ulaza unose se realan broj x i ceo pozitivan broj n . Napisati program koji izračunava n -ti stepen broja x , tj. x^n .

Primer 1

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

Primer 2

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

Primer 3

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

[Rešenje 2.72]

Zadatak 2.73 Sa standardnog ulaza unose se realan broj x i ceo broj n . Napisati program koji izračunava n -ti stepen broja x .

Primer 1

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

Primer 2

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

[Rešenje ??]

Zadatak 2.74 Napisati program koji učitava ceo pozitivan broj n i ispisuje vrednost sume kubova brojeva od 1 do n , odnosno $s = 1 + 2^3 + 3^3 + \dots + n^3$. U slučaju greške pri unosu podataka ispisati odgovarajuću poruku.

Primer 1

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

Primer 2

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

[Rešenje 2.74]

Zadatak 2.75 Napisati program koji učitava ceo pozitivan broj n i ispisuje sumu kubova, $s = 1 + 2^3 + 3^3 + \dots + k^3$, za svaku vrednost $k = 1, \dots, n$. U slučaju greške pri unosu podataka ispisati odgovarajuću poruku.

Primer 1

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

Primer 2

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

[Rešenje 2.75]

Zadatak 2.76 Sa standardnog ulaza unose se realan broj x i ceo neoznačen broj n . Napisati program koji izračunava i na standardni izlaz ispisuje sumu $S = x + 2 \cdot x^2 + 3 \cdot x^3 + \dots + n \cdot x^n$.

Primer 1

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

Primer 2

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

[Rešenje 2.76]

Zadatak 2.77 Sa standardnog ulaza unose se realan broj x i ceo neoznačen broj n . Napisati program koji izračunava i na standardni izlaz ispisuje sumu $S = 1 + \frac{1}{x} + \frac{1}{x^2} + \dots + \frac{1}{x^n}$.

Primer 1

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

Primer 2

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

[Rešenje 2.77]

*** Zadatak 2.78** Napisati program koji učitava realane brojeve x i eps i sa zadatom tačnošću eps izračunava i na standardni izlaz 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*

2 Kontrola toka

izračunati član sume u računanju sledećeg člana sume. Naime, ako je izračunat član sume $\frac{x^n}{n!}$ na osnovu njega se lako može dobiti član $\frac{x^{n+1}}{(n+1)!}$. Nikako ne računati stepen i faktorijel odvojeno zbog neefikasnosti takvog rešenja i zbog mogućnosti prekoračenja.

Primer 1

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

Primer 2

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

[Rešenje 2.78]

* **Zadatak 2.79** Napisati program koji učitava realane brojeve x i eps i sa zadatom tačnošću eps izračunava i na standardni izlaz ispisuje sumu $S = 1 - x + \frac{x^2}{2!} - \frac{x^3}{3!} + \frac{x^4}{4!} - \frac{x^5}{5!} \dots$ NAPOMENA: Voditi računa o efikasnosti rešenja i o mogućnosti prekoračenja.

Primer 1

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite x: 3
|| Unesite tacnost eps: 0.000001
|| S=-1.131112
```

Primer 2

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

[Rešenje 2.79]

Zadatak 2.80 Napisati program koji učitava realan broj x i prirodan broj n izračunava sumu $S = (1 + \cos(x)) \cdot (1 + \cos(x^2)) \cdot \dots \cdot (1 + \cos(x^n))$. NAPOMENA: Voditi računa o efikasnosti rešenja.

Primer 1

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

Primer 2

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

[Rešenje 2.80]

* **Zadatak 2.81** Napisati program koji učitava ceo prirodan broj n , a na standardni izlaz ispisuje vrednost razlomka

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

Primer 1

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

Primer 2

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

Primer 3

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

[Rešenje 2.81]

* **Zadatak 2.82** 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.
```

[Rešenje 2.82]

* **Zadatak 2.83** Sa standardnog ulaza unosi se ceo pozitivan broj n veći od 0. Napisati program koji računa proizvod

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

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

Primer 1

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

Primer 2

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

Primer 3

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

Primer 4

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

[Rešenje 2.83]

* **Zadatak 2.84** Sa standardnog ulaza unosi se ceo pozitivan neparan broj n . Napisati program koji za uneto n izračunava:

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

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

Primer 1

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

Primer 2

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

Primer 3

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite broj n: 20
|| Neispravan unos
```

Primer 4

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

[Rešenje 2.84]

Zadatak 2.85 Sa standardnog ulaza unose se realni brojevi x i a i ceo pozitivan broj n veći od 0. Napisati program koji izračunava:

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

U slučaju greške pri unosu podataka ispisati odgovarajuću poruku.

Primer 1

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

Primer 2

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

Primer 3

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

Primer 4

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

[Rešenje 2.85]

2.3.6 Dvostruka petlja i ispisivanje slike

Zadatak 2.86 Sa standardnog ulaza unosi se neoznačen broj n . Napisati program koji za uneto n zvezdicama iscrtava

- a) kvadrat stranice n sastavljen od zvezdica.

Primer 1

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

- b) rub kvadrata dimenzije n .

Primer 1

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

Primer 2

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

- c) rub kvadrata dimenzije n koji i na glavnoj dijagonali ima zvezdice.

Primer 1

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

[Rešenje 2.98]

* **Zadatak 2.87** Napisati program koji za uneti ceo 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
* *
*
* *
```

[Rešenje 2.88]

* **Zadatak 2.88** Napisati program koji za uneti ceo 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
+
+++
+
```

[Rešenje 2.88]

Zadatak 2.89 Napisati program koji učitava ceo neoznačen broj n , a potom iscrtava

- a) 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
***
**
*
```

- b) 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
*
**
***
```


- c) 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
*
**
***
**
*
```

- d) rub jednakokrakog pravouglog trougla čije su katete dužine n . Program učitava karakter c i taj karakter koristi za iscrtavanje ruba trougla.

Primer 1

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

Primer 2

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

[Rešenje 2.98]

Zadatak 2.90 Napisati program koji učitava ceo broj n , a potom iscrtava

- a) jednakostranični trougao stranice n koji je sastavljen od zvezdica.

Primer 1

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

- b) trougao koji se dobija spajanjem dva jednakostranični trougla stranice n koji su sastavljeni od zvezdica.

Primer 1

```

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

```

c) rub jednakostraničnog trougla čija stranica je dužine n .

Primer 1

```

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

```

d) sliku koja se dobija spajanjem dva jednakostranična trougla čija stranica je dužine n . Iscrtavati samo rub trouglova.

Primer 1

```

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

```

[Rešenje 2.98]

* **Zadatak 2.91** Napisati program koji za uneti ceo broj n iscrtava strelice dimenzije n .

Primer 1

```

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

```

Primer 2

```

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

```

[Rešenje 2.91]

*** Zadatak 2.92** 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
*
* *
***
* * *
****
* * * *
*****
```

[Rešenje 2.98]

**** Zadatak 2.93** Sa standardnog ulaza unose se neoznačeni celi brojevi m i n . Napisati program koji iscrtava jedan do drugog stranice n kvadrata čija je svaka strana sastavljena od m zvezdica razdvojenih prazninom.

Tekst nije u skladu sa slikom jer nije jasno da se crtaju samo rubovi a ne popunjeni kvadrati.

Danijela: Da li je sada jasnije?

Primer 1

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

Primer 2

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

[Rešenje 2.98]

*** Zadatak 2.94** Sa standardnog ulaza unosi se ceo neoznačen 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
****
*--*
****
```

[Rešenje 2.98]

Zadatak 2.95 Napisati program koji učitava ceo broj n ($n \geq 2$) i koji na standardni izlaz iscrta sliku kuće sa krovom: kuća je kocka stranice n , a krov jednakostranični trougao stranice n .

Primer 1

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

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

[Rešenje 2.98]

Zadatak 2.96 Sa standardnog ulaza učitava se ceo neoznačen broj n . Napisati program koji za uneto n iscrtaava pravougli „trougao” sačinjen od „koordinata” svojih tačaka. „Koordinata” tačke je oblika (i, j) pri čemu $i, j = 0, \dots, n$. Pravougao 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 $(0, 1)$.

Ovo treba preformulisati jer je bez test primera skroz nejasno. U test prime-
rima negde ima blanko posle zareza, negde nema, i to treba ujednaciti.

Mene ovaj zadatak zbunjuje i ne sviđa mi se. Problem su mi koordinate koje se broje nekako cudno i to od broja 1 a ne od nule. Nije mi jasno zasto u temenu pravog ugla ne bi bila koordinata (0,0)?

Danijela: Izmenila sam test primere i tekst, ali se slazem da zadatak nije nesto, mozemo ga izbrisati.

Primer 1

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

Primer 2

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

Primer 3

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

Primer 4

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

[Rešenje 2.98]

* **Zadatak 2.97** Sa standardnog ulaza unosi se 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
```

[Rešenje 2.98]

* **Zadatak 2.98** 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: 1
1
```

Primer 3

```
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 4

```
INTERAKCIJA SA PROGRAMOM:
Unesite broj n: -23
```

[Rešenje 2.98]

2.4 Rešenja

Rešenje 2.37

```
#include <stdio.h>

2
int main()
4 {
    /* Promenljiva i kontrolise koliko puta ce se petlja izvesti i
       naziva se brojac petlje.
       Njenu pocetnu vrednost postavljamo na 0 jer se u pocetku petlja
       nije ni jednom izvela. */
6     int i = 0;

8     /* Pre ulaska u telo petlje proverava se da li je
       ispunjen uslov ulaska u petlju.
10    */
12    while(i < 5)
    {
14        /* Ukoliko uslov ulaska u petlju jeste ispunjen, izvela se telo
           petlje. */
           /* Ispisujemo trazeni tekst. */
16        printf("Mi volimo da programiramo.\n");

18        /* Uvecava se brojac za jedan jer je jednom izvelo telo petlje.
           Ako bi ova vrednost ostala nepromenjena, petlja bi se
           izvela beskonacno. */
```

```
20     i++;  
  
22     /* Nakon poslednje naredbe tela petlje ponovo se ispituje uslov  
    petlje.*/  
  
24 }  
  
26 return 0;  
}
```

Rešenje 2.38

```
#include<stdio.h>  
  
2 int main()  
4 {  
    /* Brojac u petlji. */  
    6 int i=0;  
    /* Promenljiva koja oznacava koliko puta cemo ispisati trazeni  
    tekst. */  
    8 int n;  
  
    10 printf("Unesite ceo broj: ");  
    scanf("%d", &n);  
  
    12 /* 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 2.39

```
1 #include <stdio.h>  
  
3 int main()  
4 {  
    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 određuje koliko brojeva ispisujemo.  
    */  
    9 int n;
```

2 Kontrola toka

```
11     printf("Unesi pozitivan ceo broj: ");
12     scanf("%d", &n);
13
14     /* U slucaju neispravnih podataka ispisujemo odgovarajucu poruku
15        i izlazimo iz programa. */
16     if (n < 0)
17     {
18         printf("Neispravan unos. Promenljiva mora biti pozitivna!\n");
19         return -1;
20     }
21
22     /* Ispis pocinjemo od 0, zato promenljivu x postavljamo na 0. */
23     x=0;
24     while (x<= n)
25     {
26         /* Ispisujemo broj. */
27         printf("%d\n", x);
28         /* Uvecavamo promenljivu za jedan jer smo broj ispisali i sada
29            zelimo da ispisemo sledeci broj. */
30         x++;
31     }
32     return 0;
33 }
```

Rešenje 2.40

```
1  /* Resenje pod a). */
2
3  #include <stdio.h>
4
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;
11
12     printf("Unesi dva cela broja: ");
13     scanf("%d%d",&n,&m);
14
15     if (m < n)
16     {
17         printf("Neispravan unos. Nisu dobro zadate granice intervala!\n");
18     };
19     return -1;
20 }
21
22 /* Na pocetku ispisujemo prvi broj intervala, a to je n. */
23 i=n;
24 /* uslov petlje se proverava pre ulaska u telo petlje */
```



```
25     while (i<=m)
26     {
27         printf("%d ", i);
28         i++;
29     }
30
31     printf("\n");
32
33     return 0;
34 }
```

```
2  /* Resenje pod b). */
3
4  #include <stdio.h>
5
6  int main()
7  {
8
9      /* Promenljive koje oznacavaju granice intervala. */
10     int n,m;
11     /* Promenljiva koja oznacava trenutno ispisani broj intervala. */
12     int i;
13
14     printf("Unesi dva cela broja: ");
15     scanf("%d%d",&n,&m);
16
17     if (m < n)
18     {
19         printf("Neispravan unos. Nisu dobro zadate granice intervala!\n");
20     }
21     return -1;
22 }
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000
1001
1002
1003
1004
1005
1006
1007
1008
1009
1010
1011
1012
1013
1014
1015
1016
1017
1018
1019
1020
1021
1022
1023
1024
1025
1026
1027
1028
1029
1030
1031
1032
1033
1034
1035
1036
1037
1038
1039
1040
1041
1042
1043
1044
1045
1046
1047
1048
1049
1050
1051
1052
1053
1054
1055
1056
1057
1058
1059
1060
1061
1062
1063
1064
1065
1066
1067
1068
1069
1070
1071
1072
1073
1074
1075
1076
1077
1078
1079
1080
1081
1082
1083
1084
1085
1086
1087
1088
1089
1090
1091
1092
1093
1094
1095
1096
1097
1098
1099
1100
1101
1102
1103
1104
1105
1106
1107
1108
1109
1110
1111
1112
1113
1114
1115
1116
1117
1118
1119
1120
1121
1122
1123
1124
1125
1126
1127
1128
1129
1130
1131
1132
1133
1134
1135
1136
1137
1138
1139
1140
1141
1142
1143
1144
1145
1146
1147
1148
1149
1150
1151
1152
1153
1154
1155
1156
1157
1158
1159
1160
1161
1162
1163
1164
1165
1166
1167
1168
1169
1170
1171
1172
1173
1174
1175
1176
1177
1178
1179
1180
1181
1182
1183
1184
1185
1186
1187
1188
1189
1190
1191
1192
1193
1194
1195
1196
1197
1198
1199
1200
1201
1202
1203
1204
1205
1206
1207
1208
1209
1210
1211
1212
1213
1214
1215
1216
1217
1218
1219
1220
1221
1222
1223
1224
1225
1226
1227
1228
1229
1230
1231
1232
1233
1234
1235
1236
1237
1238
1239
1240
1241
1242
1243
1244
1245
1246
1247
1248
1249
1250
1251
1252
1253
1254
1255
1256
1257
1258
1259
1260
1261
1262
1263
1264
1265
1266
1267
1268
1269
1270
1271
1272
1273
1274
1275
1276
1277
1278
1279
1280
1281
1282
1283
1284
1285
1286
1287
1288
1289
1290
1291
1292
1293
1294
1295
1296
1297
1298
1299
1300
1301
1302
1303
1304
1305
1306
1307
1308
1309
1310
1311
1312
1313
1314
1315
1316
1317
1318
1319
1320
1321
1322
1323
1324
1325
1326
1327
1328
1329
1330
1331
1332
1333
1334
1335
1336
1337
1338
1339
1340
1341
1342
1343
1344
1345
1346
1347
1348
1349
1350
1351
1352
1353
1354
1355
1356
1357
1358
1359
1360
1361
1362
1363
1364
1365
1366
1367
1368
1369
1370
1371
1372
1373
1374
1375
1376
1377
1378
1379
1380
1381
1382
1383
1384
1385
1386
1387
1388
1389
1390
1391
1392
1393
1394
1395
1396
1397
1398
1399
1400
1401
1402
1403
1404
1405
1406
1407
1408
1409
1410
1411
1412
1413
1414
1415
1416
1417
1418
1419
1420
1421
1422
1423
1424
1425
1426
1427
1428
1429
1430
1431
1432
1433
1434
1435
1436
1437
1438
1439
1440
1441
1442
1443
1444
1445
1446
1447
1448
1449
1450
1451
1452
1453
1454
1455
1456
1457
1458
1459
1460
1461
1462
1463
1464
1465
1466
1467
1468
1469
1470
1471
1472
1473
1474
1475
1476
1477
1478
1479
1480
1481
1482
1483
1484
1485
1486
1487
1488
1489
1490
1491
1492
1493
1494
1495
1496
1497
1498
1499
1500
1501
1502
1503
1504
1505
1506
1507
1508
1509
1510
1511
1512
1513
1514
1515
1516
1517
1518
1519
1520
1521
1522
1523
1524
1525
1526
1527
1528
1529
1530
1531
1532
1533
1534
1535
1536
1537
1538
1539
1540
1541
1542
1543
1544
1545
1546
1547
1548
1549
1550
1551
1552
1553
1554
1555
1556
1557
1558
1559
1560
1561
1562
1563
1564
1565
1566
1567
1568
1569
1570
1571
1572
1573
1574
1575
1576
1577
1578
1579
1580
1581
1582
1583
1584
1585
1586
1587
1588
1589
1590
1591
1592
1593
1594
1595
1596
1597
1598
1599
1600
1601
1602
1603
1604
1605
1606
1607
1608
1609
1610
1611
1612
1613
1614
1615
1616
1617
1618
1619
1620
1621
1622
1623
1624
1625
1626
1627
1628
1629
1630
1631
1632
1633
1634
1635
1636
1637
1638
1639
1640
1641
1642
1643
1644
1645
1646
1647
1648
1649
1650
1651
1652
1653
1654
1655
1656
1657
1658
1659
1660
1661
1662
1663
1664
1665
1666
1667
1668
1669
1670
1671
1672
1673
1674
1675
1676
1677
1678
1679
1680
1681
1682
1683
1684
1685
1686
1687
1688
1689
1690
1691
1692
1693
1694
1695
1696
1697
1698
1699
1700
1701
1702
1703
1704
1705
1706
1707
1708
1709
1710
1711
1712
1713
1714
1715
1716
1717
1718
1719
1720
1721
1722
1723
1724
1725
1726
1727
1728
1729
1730
1731
1732
1733
1734
1735
1736
1737
1738
1739
1740
1741
1742
1743
1744
1745
1746
1747
1748
1749
1750
1751
1752
1753
1754
1755
1756
1757
1758
1759
1760
1761
1762
1763
1764
1765
1766
1767
1768
1769
1770
1771
1772
1773
1774
1775
1776
1777
1778
1779
1780
1781
1782
1783
1784
1785
1786
1787
1788
1789
1790
1791
1792
1793
1794
1795
1796
1797
1798
1799
1800
1801
1802
1803
1804
1805
1806
1807
1808
1809
1810
1811
1812
1813
1814
1815
1816
1817
1818
1819
1820
1821
1822
1823
1824
1825
1826
1827
1828
1829
1830
1831
1832
1833
1834
1835
1836
1837
1838
1839
1840
1841
1842
1843
1844
1845
1846
1847
1848
1849
1850
1851
1852
1853
1854
1855
1856
1857
1858
1859
1860
1861
1862
1863
1864
1865
1866
1867
1868
1869
1870
1871
1872
1873
1874
1875
1876
1877
1878
1879
1880
1881
1882
1883
1884
1885
1886
1887
1888
1889
1890
1891
1892
1893
1894
1895
1896
1897
1898
1899
1900
1901
1902
1903
1904
1905
1906
1907
1908
1909
1910
1911
1912
1913
1914
1915
1916
1917
1918
1919
1920
1921
1922
1923
1924
1925
1926
1927
1928
1929
1930
1931
1932
1933
1934
1935
1936
1937
1938
1939
1940
1941
1942
1943
1944
1945
1946
1947
1948
1949
1950
1951
1952
1953
1954
1955
1956
1957
1958
1959
1960
1961
1962
1963
1964
1965
1966
1967
1968
1969
1970
1971
1972
1973
1974
1975
1976
1977
1978
1979
1980
1981
1982
1983
1984
1985
1986
1987
1988
1989
1990
1991
1992
1993
1994
1995
1996
1997
1998
1999
2000
2001
2002
2003
2004
2005
2006
2007
2008
2009
2010
2011
2012
2013
2014
2015
2016
2017
2018
2019
2020
2021
2022
2023
2024
2025
2026
2027
2028
2029
2030
2031
2032
2033
2034
2035
2036
2037
2038
2039
2040
2041
2042
2043
2044
2045
2046
2047
2048
2049
2050
2051
2052
2053
2054
2055
2056
2057
2058
2059
2060
2061
2062
2063
2064
2065
2066
2067
2068
2069
2070
2071
2072
2073
2074
2075
2076
2077
2078
2079
2080
2081
2082
2083
2084
2085
2086
2087
2088
2089
2090
2091
2092
2093
2094
2095
2096
2097
2098
2099
2100
2101
2102
2103
2104
2105
2106
2107
2108
2109
2110
2111
2112
2113
2114
2115
2116
2117
2118
2119
2120
2121
2122
2123
2124
2125
2126
2127
2128
2129
2130
2131
2132
2133
2134
2135
2136
2137
2138
2139
2140
2141
2142
2143
2144
2145
2146
2147
2148
2149
2150
2151
2152
2153
2154
2155
2156
2157
2158
2159
2160
2161
2162
2163
2164
2165
2166
2167
2168
2169
2170
2171
2172
2173
2174
2175
2176
2177
2178
2179
2180
2181
2182
2183
2184
2185
2186
2187
2188
2189
2190
2191
2192
2193
2194
2195
2196
2197
2198
2199
2200
2201
2202
2203
2204
2205
2206
2207
2208
2209
2210
2211
2212
2213
2214
2215
2216
2217
2218
2219
2220
2221
2222
2223
2224
2225
2226
2227
2228
2229
2230
2231
2232
2233
2234
2235
2236
2237
2238
2239
2240
2241
2242
2243
2244
2245
2246
2247
2248
2249
2250
2251
2252
2253
2254
2255
2256
2257
2258
2259
2260
2261
2262
2263
2264
2265
2266
2267
2268
2269
2270
2271
2272
2273
2274
2275
2276
2277
2278
2279
2280
2281
2282
2283
2284
2285
2286
2287
2288
2289
2290
2291
2292
2293
2294
2295
2296
2297
2298
2299
2300
2301
2302
2303
2304
2305
2306
2307
2308
2309
2310
2311
2312
2313
2314
2315
2316
2317
2318
2319
2320
2321
2322
2323
2324
2325
2326
2327
2328
2329
2330
2331
2332
2333
2334
2335
2336
2337
2338
2339
2340
2341
2342
2343
2344
2345
2346
2347
2348
2349
2350
2351
2352
2353
2354
2355
2356
2357
2358
2359
2360
2361
2362
2363
2364
2365
2366
2367
2368
2369
2370
2371
2372
2373
2374
2375
2376
2377
2378
2379
2380
2381
2382
2383
2384
2385
2386
2387
2388
2389
2390
2391
2392
2393
2394
2395
2396
2397
2398
2399
2400
2401
2402
2403
2404
2405
2406
2407
2408
2409
2410
2411
2412
2413
2414
2415
2416
2417
2418
2419
2420
2421
2422
2423
2424
2425
2426
2427
2428
2429
2430
2431
2432
2433
2434
2435
2436
2437
2438
2439
2440
2441
2442
2443
2444
2445
2446
2447
2448
2449
2450
2451
2452
2453
2454
2455
2456
2457
2458
2459
2460
2461
2462
2463
2464
2465
2466
2467
2468
2469
2470
2471
2472
2473
2474
2475
2476
2477
2478
2479
2480
2481
2482
2483
2484
2485
2486
2487
2488
2489
2490
2491
2492
2493
2494
2495
2496
2497
2498
2499
2500
2501
2502
2503
2504
2505
2506
2507
2508
2509
2510
2511
2512
2513
2514
2515
2516
2517
2518
2519
2520
2521
2522
2523
2524
2525
2526
2527
2528
2529
2530
2531
2532
2533
2534
2535
2536
2537
2538
2539
2540
2541
2542
2543
2544
2545
2546
2547
2548
2549
2550
2551
2552
2553
2554
2555
2556
2557
2558
2559
2560
2561
2562
2563
2564
2565
2566
2567
2568
2569
2570
2571
2572
2573
2574
2575
2576
2577
2578
2579
2580
2581
2582
2583
2584
2585
2586
2587
2588
2589
2590
2591
2592
2593
2594
2595
2596
2597
2598
2599
2600
2601
2602
2603
2604
2605
2606
2607
2608
2609
2610
2611
2612
2613
2614
2615
2616
2617
2618
2619
2620
2621
2622
2623
2624
2625
2626
2627
2628
2629
2630
2631
2632
2633
2634
2635
2636
2637
2638
2639
2640
264
```

```
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;
11
12    printf("Unesi dva cela broja: ");
13    scanf("%d%d",&n,&m);
14
15    if (m < n)
16    {
17        printf("Neispravan unos. Nisu dobro zadate granice intervala!\n");
18    };
19    return -1;
20 }
21
22
23 /* Uslov petlje se proverava na kraju svake iteracije. */
24 /* Zbog toga se do while petlja izvršava bar jednom, čak i u
25    slučaju
26    da uslov petlje nikada nije ispunjen. */
27 i=n;
28 do                /* Petlja se zapocinje bez provere uslova. */
29 {
30     printf("%d ",i); /* Stampa se vrednost promenljive i. */
31     i++;             /* Uvecava se vrednost promenljive i. */
32 }
33 while(i<=m);        /* Proverava se uslov i ukoliko je ispunjen,
34                      nastavlja se sa sledecom iteracijom. */
35                      /* U suprotnom, petlja se završava i program
36                      se nastavlja od prve naredbe koja sledi za petljom. */
37 printf("\n");
38
39 return 0;
40 }
```

Rešenje 2.41

```
#include <stdio.h>
2
3 int main()
4 {
5     /* Pamtimozastopna dva Fibonacijeva broja i na osnovu njih
6        racunamo sledeci. */
7     /* Promenljive prvi i drugi su brojevi koje pamtimoz i na osnovu
8        njih racunamo treci. */
9 }
```

```

/* Na osnovu teksta zadatka, promenljive prvi i drugi postavljamo
na 1. */
8  int prvi = 1;
  int drugi = 1;
10 int treci;
/* Promenljiva pozicija je podatak koji ucitavamo i odnosi se na
   poziciju u Fibonacijevom nizu
12   za koju treba izracunati vrednost. */
  int pozicija;
14 /* Promenljiva i oznacava do koje pozicije smo izracunali vrednosti
   . Kako imamo prve dve
      vrednosti, ovu promenljivo postavljamo na 2. */
16 int i = 2;

18 printf("Unesite poziciju u Fibonacijevom nizu: ");
  scanf("%d", &pozicija);

20
/* Pozicija ne moze biti 0 i ne moze biti negativan broj. */
22 if (pozicija < 1)
{
24     printf("Neispravan unos. Pozicija u Fibonacijevom nizu mora biti
        pozitivan broj koji nije 0!\n");
        return -1;
26 }

28 while(i < pozicija)
{
30     /* Na osnovu dva uzastopna racunamo treci. */
        treci = prvi + drugi;

32
        /* Potom razmenjujemo vrednosti. Uzastopna dva koja pamtimo
           postaju
34           sledeca uzastopna dva broja Fibonacijevog niza. */
        prvi = drugi;
36        drugi = treci;

38        /* Prelazimo na racunanje sledeceg broja na sledecoj poziciji. */
        i++;
40    }

42    printf("Trazeni broj je: %d\n", drugi);

44    return 0;
}

```

Rešenje 2.42

```

1  #include<stdio.h>
  int main()
3  {
    int a0;

```

2 Kontrola toka

```
5  int an,an1;

7  printf("Unesi pocetni clan niza brojeva:");
   scanf("%d",&a0);

9

11 if (a0 <= 0)
   {
13     printf("Nekorektan unos. Broj mora biti pozitivan.\n");
       return -1;
   }

15 printf("%d\n", a0);

17 an=a0;
19 while(an!=1)
   {
21     if (an%2) /* Ukoliko je vrednost izraza an%2 razlicita od nule,*/
       {
           /* izraz se tumaci kao tacan i izvrsavaju se naredbe iz
           if grane. */
23         an1=(3*an+1)/2;
       }
25     else /* U suprotnom, ukoliko je vrednost izraza an%2 jednaka nuli
           , izraz */
       {
           /* se tumaci kao netacan i izvrsavaju se naredbe iz else
           grane. */
27         an1=an/2;
       }
29     printf("%d\n",an1);
       an=an1;
31 }

33
35 return 0;
}
```

Rešenje 2.43

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

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

```

15 printf("Uneti format papira: ");
   scanf("%u", &format);
17
   /* duzina/sirina = 1 : sqrt(2) duzina*sirina = 1000x1000mm^2
19    * Na osnovu ovih odnosa dobijamo pocetnu vrednost za sirinu
   * i duzinu, odnosno vrednosti za papir A0.
21    */
   duzina = sqrt(1000*1000/sqrt(2));
23   sirina = sqrt(2)*duzina;

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

35   /* Duzina i sirina celi brojevi. */
37   konacna_sirina = (unsigned int)sirina;
   konacna_duzina = (unsigned int)duzina;
39
   printf("%u %u\n", konacna_duzina, konacna_sirina);
41
   return 0;
43 }

```

Rešenje 2.44

```

1  #include<stdio.h>
   #include<stdlib.h>
3
   int main()
5  {
       int x;
7       char cifra;
       printf("Unesi ceo broj:");
9       scanf("%d", &x);

11      /* Pretvaranje u apsolutnu vrednost se vrši za slucaj kada je unet
       * negativan broj kako bismo osigurali da ce nam izdvojene cifre
13      * biti pozitivne.
       */
15      x = abs(x);

17      /* Kako uklanjamo cifre broja (pogledati telo petlje)
       * u nekom trenutku broj ce postati 0 jer smo uklonili
19      * sve njegove cifre. Tada prekidamo rad petlje.

```

```
21     */
22     while(x>0)
23     {
24         /* Izdvajamo poslednju cifru broja x. */
25         cifra=x%10;
26         printf("%d\n", cifra);
27         /* Uklanjammo poslednju cifru broja x. */
28         x/=10;
29     }
30
31     return 0;
32 }
```

Rešenje 2.45

```
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
25          * ako je to slucaj ispusje se i.
26          */
27         if (x%i==0)
28             printf("%d \n",i);
29     }
30
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. */
    if (x%i==0)
39  /* U slucaju kada je delilac koren broja, npr. 4 za 16,
    ispisujemo ga jednom. */
        if (i==x/i)
41            printf("%d \n",i);
        /* U suprotnom, npr. 2 za 16, ispisujemo i 2 i 8. */
43        else
            printf("%d %d \n",i,x/i);
45    }
    return 0;
47 }

```

Rešenje 2.46

```

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

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

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

13       /* Izdvajamo poslednju cifru broja. */
        cifra=n%10;

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

25        /* Ako izdvojena cifra nije jednaka broju 5, broj delimo sa 10
        * kako bi mogli da izdvojimo i preostale cifre broja na isti
27        * nacin.
        */
        n=n/10;
29    }

31    /* Ispisujemo rezultat */
33    if(indikator==0){
        printf("Cifra 5 se ne nalazi u zapisu!\n");
35    }

```

2 Kontrola toka

```
    else{
37         printf("Cifra 5 se nalazi u zapisu!\n");
    }
39
    return 0;
41 }
```

Rešenje 2.47

```
1  #include <stdio.h>
3  int main()
4  {
5      /* Prirodni broj koji se unosi. */
6      int n;
7      /* Promenljiva u koju se smesta suma cifara broja. */
8      int suma = 0;
9      /* Pomocna promenljiva u koju se smesta unesen broj. */
10     int pom_n;
11
12     printf("Unesi broj ");
13     scanf("%d", &n);
14
15     /* U zadatku pise da se unosi prirodan broj,
16      * sto znaci da treba proveriti da li je veci
17      * od 0
18      */
19
20     if (n <= 0)
21     {
22         printf("Neispravan unos.\n");
23         return -1;
24     }
25
26     /* Potrebno je koristiti pomocnu promenljivu jer u telu petlje
27      * se odstranjuju cifre broja i na taj nacin uneseni broj
28      * se menja. Nakon rada petlje potrebno je ponovo koristiti
29      * uneseni broj, a to znaci da treba sacuvati neizmenjen
30      * broj.
31      */
32     pom_n = n;
33
34     while(pom_n != 0)
35     {
36         /* Na sumu dodajemo poslednju cifru. */
37         suma += pom_n % 10;
38         /* Sa broja skidamo poslednju cifru. */
39         pom_n /= 10;
40     }
41
42     if (n % suma == 0)
```



```
43     printf("Deljiv je sumom svojih cifara.\n");
    else
45     printf("Nije deljiv sumom svojih cifara.\n");
47     return 0;
}
```

Rešenje 2.48

```
1  #include <stdio.h>
3  int main(){
    int n;
5
    /* Ucitavamo broj */
7    printf("Unesite broj: ");
    scanf("%d", &n);
9
    if(n==0){
11     printf("0\n");
    }
13    else{
        /* Sve dok je poslednja cifra u zapisu broja n nula */
15        while(n%10==0){
            /* Broj delimo sa 10 tj. uklanjamo mu nulu sa kraja */
17            n=n/10;
        }
19
        /* Ispisujemo rezultat */
21        printf("%d\n", n);
    }
23
25    return 0;
}
```

Rešenje 2.49

```
1  #include <stdio.h>
3  int main()
{
5    unsigned int x;
    /* Tezina trenutne pozicije u broju. Moze biti 1, 10, 100,
7     * 1000 itd.
    */
9    int pozicija;
    /* Trenutna izdvojena cifra iz broja x. */
11    int cifra;
```

```
13  /* Broj dobijen nakon transformacije. */
14  unsigned int y;
15
16  printf("Unesite broj: ");
17  scanf("%d", &x);
18
19  if(x <= 0)
20  {
21      printf("Nekorektan unos.\n");
22      return -1;
23  }
24
25  /* Posto pocinjemo sa izdvajanjem cifara od cifre jedinica,
26   * postavlja se tezinu (stepen) pozicije na 1.
27   */
28  pozicija = 1;
29  y = 0;
30
31  /* Provera da li ima cifara u zapisu broja. */
32  while(x > 0) {
33
34      /* Izdvaja se poslednja cifra iz zapisa. */
35      cifra = x % 10;
36
37      /* Provera da li je cifra parna. */
38      if(cifra % 2 == 0){
39          /* I ako jeste, uvecava se. */
40          cifra++;
41      }
42
43      /* Novi broj se formira tako sto se izdvojena cifra pomnozi
44       * odgovarajucom tezinom (stepenom) pozicije.
45       */
46      y += cifra*pozicija;
47
48      /* Priprema se broj za izdvajanje naredne cifre, uklanja se
49       * poslednja cifra broja.
50       */
51      x /= 10;
52
53      /* Uvecava se tezinu (stepen) pozicije. */
54      pozicija *= 10;
55  }
56
57  /* Ispisuje se izracunatu vrednost. */
58  printf("%d\n", y);
59
60  return 0;
61 }
62 }
```

Rešenje 2.50

```
1  #include <stdio.h>
   #include <math.h>
3
   int main() {
5
       unsigned int x;
7       /* Tezina trenutne pozicije u broju. Moze biti 1, 10, 100,
          * 1000 itd.
9          */
       int stepen_deset;
11      /* Trenutna izdvojena cifra iz broja x. */
       int cifra;
13      /* Redni broj cifre koja se trenutno obradjuje, gledano s desna
          * na levo.
15          */
       int rbr;
17      /* Broj dobijen nakon transformacije. */
       unsigned int y;
19
       /* Ucitavanje broja. */
21      printf("Unesite broj: ");
       scanf("%d", &x);
23
       if(x <= 0)
25       {
           printf("Nekorektan unos.\n");
27           return -1;
       }
29      /* Postavlja se vrednost stepena na 0 - to znaci da se prvo mnozi
          sa
          * 10^0=1.
31          */
       stepen_deset = 0;
33
       /* Postavlja se vrednost broja koji se formira na 0. */
35       y = 0;
       /* Postavlja se redni broj pozicije na 0. */
37       rbr = 0;
39
       /* Provera da li ima cifara u zapisu broja. */
       while(x > 0) {
41
           /* Izdvajanje cifre. */
43           cifra = x%10;
45
           /* Proverava se da li je pozicija izdvojene cifre parna -
              * cifre na parnim pozicijama se zadrzavaju.
47              */
           if(rbr % 2 == 0) {
49               /* Ako jeste parna
```

```
51     * izdvojena cifra se dodaje novom broju.
    * Neophodno je izvršiti promenu tipova, jer je double
    povratni
    * tip funkcije pow.
53     */
    y += cifra * ((int) pow(10, stepen_deset));
55
    /* Uvecava se stepen zbog naredne cifre. */
57     stepen_deset++;
    }
59
    /* Azurira se redni broj cifre. */
61     rbr++;
    /* I priprema se broj za naredno izdvajanje. */
63     x /= 10;
    }
65
    /* Ispisuje se rezultat. */
67     printf("%d\n", y);
69
    return 0;
71 }
```

Rešenje 2.51

```
1  #include <stdio.h>
3  int main(){
    unsigned n, novo_n;
5     int stepen;
    int cifra_levo, cifra_sredina, cifra_desno;
7
    /* Ucitavanje broja. */
9     printf("Unesite broj: ");
    scanf("%u", &n);
11
13     /* Stepen broja 10 sa kojim se mnoze cifre izdvojenog broja. */
    stepen=1;
15
    /* Nova vrednost broja. */
17     novo_n=0;
19
    /* Provera da li u zapisu broja postoje barem tri cifre. */
    while(n>99){
21         /* Izdvaja se srednja cifra, cifra desno od nje i cifra
            * levo od nje: npr. za trojku 583 8 je srednja cifra,
23             * 3 je cifra desno, a 5 cifra levo.
            */
25         cifra_desno=n%10;
```

```

27     cifra_sredina=(n/10)%10;
    cifra_levo=(n/100)%10;

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

31     /* Azurira se vrednost stepena. */
    stepen=stepen*10;

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

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

37         /* Inace, zadrzava se srednja cifra i odbacuje
    * se samo poslednja.
    */
        n=n/10;

39     }
}

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

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

45     return 0;

47 }

```

Rešenje 2.52

```

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

3  int main() {

5     int x;
   int broj_cifara;
   int min_stepen, max_stepen;

```

2 Kontrola toka

```
9  int pom;
10 int leva_cifra, desna_cifra;
11 int indikator;

12
13 printf("Unesite broj: ");
14 scanf("%d", &x);

15
16 /* Ako je korisnik uneo negativan broj, analizira se njegova
17  * apsolutna vrednost.
18  */
19 if(x < 0)
20     x=-x;

21
22
23 /* Odredjuje se broj cifara u zapisu broja x da bi moglo da
24  * se izdvajaju istovremeno cifre i sa leve i sa desne strane.
25  */
26 broj_cifara = 0;
27 pom = x;
28 while(pom > 0) {
29     pom /= 10;
30     broj_cifara++;
31 }

32
33 /* Odredjuje se stepen koji stoji uz krajnju levu cifru broja. */
34 max_stepen = (int) pow (10, broj_cifara-1);

35
36 /* Indikator je promenljiva koja ukazuje da li je broj
37  * palindrom ili ne.
38  */
39 indikator=1;
40 while(x!=0 && indikator==1){
41     /* Izdvaja se leva cifra. */
42     leva_cifra=x/max_stepen;
43     /* Izdvaja se desna cifra. */
44     desna_cifra=x%10;
45     /* Ako su cifre razlicite, odmah se moze zakljuciti da
46      * broj nije palindrom i prekida se izvršavanje petlje.
47     */
48     if(leva_cifra!=desna_cifra){
49         indikator=0;
50         break;
51     }
52     /* Formira se nova vrednost broja x tako sto se odbacuje
53      * krajnja leva i krajnja desna cifra.
54     */
55     x=(x/max_stepen-x%10)/10;
56     /* Koriguje se maksimalan stepen tako dobijenog broja -
57      * deli se sa 100 jer su odbacene 2 cifre.
58     */
59     max_stepen=max_stepen/100;
60 }
```

```
61     /* Ispisuje se rezultat. */
63     if(indikator==1)
        printf("Broj je palindrom!\n");
65     else
        printf("Broj nije palindrom!\n");
67
69     return 0;
}
```

Rešenje 2.53

```
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;
12
13     printf("Unesi pozitivan ceo broj:");
14     scanf("%d",&n);
15
16     if (n < 0)
17     {
18         printf("Neispravan unos.\n");
19         return -1;
20     }
21
22     /* Promenljivama koje ce sadrzati sume se pre ulaska u petlju
23      * dodeljuje 0 (neutral za sabiranje).
24      */
25     suma_poz=0;
26     suma_neg=0;
27     i=0;
28
29     printf("Unesite brojeve: ");
30     while(i<n)
31     {
32         scanf("%d", &x);
33
34         if (x<0)
35             suma_neg+=x;
36         else
37             suma_poz+=x;
```

2 Kontrola toka

```
39     i++;
40 }
41
42 printf(" Suma pozitivnih: %d\n Suma negativnih: %d\n", suma_poz,
43        suma_neg);
44 return 0;
45 }
```

Rešenje 2.54

```
1  #include <stdio.h>
2
3  int main()
4  {
5      /* Promenljiva x oznacava tekuci uneti broj. */
6      int n, x;
7      /* Brojac. */
8      int i;
9      int zbir=0;
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     printf("Unesite n brojeva: ");
21
22     /* Inicijalizuje se brojac sa kojim se kontrolise broj
23      * učitavanja - treba da ih bude tacno n.
24      */
25     i=0;
26     while(i<n){
27         /* Ucitava se broj. */
28         scanf("%d", &x);
29
30         /* Proverava se da li broj negativan i neparan. */
31         if(x<0 && x%2!=0){
32             /* Ako jeste, dodajemo ga na zbir. */
33             zbir=zbir+x;
34         }
35
36         /* Uvecava se brojac iteracija. */
37         i++;
38     }
39
40     /* Ispisuje se rezultat. */
41     printf("%d\n", zbir);
```



```
43     return 0;
}
```

Rešenje 2.55

```
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)
30             /* Naredba break prekida petlju. Izvršavanje
31             * se nastavlja od prve naredbe nakon petlje.
32             */
33             break;
34
35         /* Ako je makar 1 broj razlicit od 0 promenljiva u
36         * ce biti postavljena na 1.
37         */
38         u = 1;
39         /* Ako je unet negativan broj, taj broj se ne množi
40         * sa ukupnim proizvodom p; zato se nastavlja dalje.
41         */
42         if (x<0)
43             /* Naredba continue prekida trenutnu iteraciju petlje
44             * tako sto preskace sve naredbe koje nakon njega slede.
45             * Izvršavanje se nastavlja od provere uslova petlje.
```

2 Kontrola toka

```
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
63 return 0;
}
```

Rešenje 2.56

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

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

15
17    if (n<=0)
    {
19        printf("Neispravan unos\n");
        return -1;
21    }

23    /* Prva cena se unosi iznad petlje
    * kako bi bio njegova vrednost bila dodeljena promenljivoj
    min_cena.
25    * Neophodno je da promenljiva min bude inicijalizovana pre
    ulaska u petlju
    * da bi uslov x<min mogao da bude ispitan u prvoj iteraciji.
27    */
    printf("Unesite cenu artikala:");
```

```

29     scanf("%f", &cena);
    /* Proveravamo da li je cena isprano uneta vrednost. */
31     if (cena <= 0)
    {
33         printf("Cena ne moze biti negativna.\n");
        return -1;
35     }

37     min_cena = cena;

39

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

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

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

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

```

Rešenje 2.57

```

1  #include <stdio.h>
    #include <math.h>
3
    int main(){
5
6     float x;
7     int broj_brojeva;
    float suma;
9
10    broj_brojeva=0;
11    suma=0;

12
13    printf("Unesite brojeve: ");
15
    while(1){

```

2 Kontrola toka

```
17      /* Ucitavanje broja. */
      scanf("%f", &x);

19

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

23

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

27

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

31

    if (broj_brojeva == 0)
33        printf("Nisu uneseni brojevi.\n");
    else
35        printf("Aritmeticka sredina: %.4f\n", suma/broj_brojeva);

37    return 0;
}
```

Rešenje 2.58

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

3

   int main(){

5

       float cena;
       int broj_artikla;
       float suma;

9

       broj_artikla=0;
       suma=0;

11

13       printf("Unesite cene: ");

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

19           if(cena == 0)
               break;

21           if (cena < 0)
23           {
               printf("Cena ne moze biti negativna.\n");
25               return -1;
           }
       }
```

```
27     suma+=cena;
29
31     /* I uvecava se broj ucitanih brojeva. */
    broj_artikla++;
33 }
35
37     if (broj_artikla == 0)
        printf("Nisu unesene cene.\n");
    else
        printf("Aritmeticka sredina: %.4f\n", suma/broj_brojeva);
39
41     return 0;
43 }
```

Rešenje 2.59

```
1  #include <stdio.h>
3
5  int main()
6  {
7      /* Promenljiva cena oznacava trenutno unesenu cenu. */
8      float cena;
9      float m;
10     int n, i;
11     int broj_brojeva=0;
12
13     printf("Unesite koliko novaca ima Nikola: ");
14     scanf("%f", &m);
15
16     if (m < 0)
17     {
18         printf("Nikola ne moze imati negativno novaca.\n");
19         return -1;
20     }
21
22     printf("Unesite broj artikala: ");
23     scanf("%d", &n);
24
25     if (n < 0)
26     {
27         printf("Broj artikala ne moze biti negativan.\n");
28         return -1;
29     }
30
31     printf("Unesite cene artikala: ");
32
33     i=0;
34     while(i<n){
35         /* Ucitava se cena artikla. */
36         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 2.98

```
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     }
18
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 }
```

```
29     return 0;
31 }
```

Rešenje 2.98

```
1  #include <stdio.h>
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     }
30
31     printf("Unesite brojeve: ");
32     /* Pre petlje ucitava se jedan broj, a u petlji
33      * se ucitava njegov sledbenik i proverava se
34      * da li su razlicitog znaka. */
35     scanf("%lf", &broj);
36
37     /* Kako je vec jedan broj unesen, brojac se postavlja
38      * na 1, a ne na 0. */
39     for(i=1; i<n; i++)
40     {
41         /* Ucitava se sledbenik. */
42         scanf("%lf", &sledbenik);
43
44         /* Ako su razlicitog znaka proizvod je manji od 0. */
```

2 Kontrola toka

```
45     if (sledbenik * broj < 0)
46         broj_promena++;
47     /* Problem je ako je proizvod jednak 0. Tada mora
48        * proveriti da li je jedan od brojeva negativan jer tada
49        * postoji promena znaka. */
50     else if (sledbenik*broj==0 && (sledbenik < 0 || broj < 0))
51         broj_promena++;
52
53     /* Tekuci sledbenik postaje tekuci broj, a
54        * u sledecoj iteraciji petlje se ucitava
55        * sledeci sledbenik. */
56     broj = sledbenik;
57 }
58
59 printf("Broj promena je %d.\n", broj_promena);
60
61 return 0;
62 }
```

Rešenje 2.62

```
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
```



```

29      * jer nije uvek zgodno pretpostaviti da je maksimalna
      vrednost 0.
      * Na primer, ako je zadatak naci maksimum celih brojeva, a
      korisnik
31      * unese -32 -7 i -22, maksimalni je broj -7, sto je manje od
      0.
      */
33      max_desetica=0;

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

39          /* Izdvajanje cifre desetice procitanog broja. */
          x_desetica=(abs(x)/10)%10;

41          /* Proverava da li je izdvojena cifra veca od
          * trenutne maksimalne cifre desetice. */
          if(x_desetica>max_desetica)
43              {
45                  /* Ako jeste vece, pamti se nova najveća cifra,
47                     * kao i broj u kom se pojavila. */
                  max_desetica=x_desetica;
49                  broj=x;
          }
51      }

53      printf("Broj sa najvećom cifrom desetice je %d\n", broj);

55      return 0;

57  }

```

Rešenje 2.63

```

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

3
   int main(){
5
       int n;
       int x, x_kopija, br_cifara;
7       int max_br_cifara, broj;
       int i;
9

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

13          if (n < 0)
15          {
               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;
38         while(x_kopija!=0){
39             x_kopija=x_kopija/10;
40             br_cifara++;
41         }
42
43         /* Ako je broj cifara unetog broja veci od maksimalnog */
44         if(br_cifara>max_br_cifara){
45             /* Cuvamo ga */
46             max_br_cifara=br_cifara;
47             /* I zbog ispisa rezultata, cuvamo i originalni broj */
48             /* Zbog ovoga smo morali i da racunamo broj cifara nad kopijom
49                broja x kako ne bismo promenili njegovu vrednost */
50             broj=x;
51         }
52     }
53
54     printf("Najvise cifara ima broj %d\n", broj);
55
56     return 0;
57 }
}
```

Rešenje 2.64

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

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

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

44     /* Ispisujemo rezultat */
    printf("%d\n", broj);

46     return 0;

48
50 }
```

Rešenje 2.65

```
1 #include <stdio.h>
2 #include <stdlib.h>
3
4 int main(){
5
6     int n;
7     int d;
8     /* Uzastopni brojevi za koje se racuna rastojanje. */
9     int x, y;
10    int broj_parova;
11    int i;
12
13    printf("Unesite brojeve n i d: ");
14    scanf("%d %d", &n, &d);
15
16    if (n < 0 || d < 0)
17    {
18        printf("Neispravan unos.\n");
19        return -1;
20    }
21
22    broj_parova=0;
23
24    printf("Unesite n brojeva: ");
25
26    /* Prvi broj se učitava pre petlje. */
27    scanf("%d", &x);
28
29    for(i=1; i<n; i++){
30        scanf("%d", &y);
31
32        /* Provera da li su brojevi na rastojanju d. */
33        if(abs(y-x)==d)
34            broj_parova++;
35
36        /* Broj iz tekuće iteracije se čuva kako bi
37           * mogao da se upotrebljava u narednoj iteraciji. */
38        x=y;
39    }
40
41    printf("Broj parova: %d\n", broj_parova);
42
43    return 0;
44
45 }
```

Rešenje 2.66

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

```
1 #include <math.h>
3
5 int main(){
7     int x;
9     int min, max;
11
13     printf("Unesite brojeve: ");
15
17     /* Prvi broj se učitava izvan petlje zbog
18        * inicijalizacije maksimuma i minimuma. */
19     scanf("%d", &x);
20     max=x;
21     min=x;
23
25     /* Sve dok se ne unese 0, učitavaju se brojevi u petlji. */
26     while(x!=0){
28         /* Provera da li je procitani broj veci od aktuelnog maksimuma.
29         */
30         if(x>max)
31             max=x;
32         /* Provera da li je procitani broj manji od aktuelnog minimuma.
33         */
34         if(x<min)
35             min=x;
37         /* Učitavanje narednog broja. */
38         scanf("%d", &x);
39     }
41
43     printf("Razlika: %d\n", max-min);
45
47     return 0;
49 }
```

Rešenje 2.67

```
1 #include <stdio.h>
3
5 int main()
6 {
7     char c;
9
11    /* Funkcija getchar učitava jedan karakter.
12     * Naredbom dodele (c=getchar()) promenljivoj c
13     * bice dodeljena vrednost ascii koda unetog karaktera.
14     * Obratiti posebnu paznju na zagrade. */
15
17    while((c=getchar())!='.')
18    {
19    }
```

2 Kontrola toka

```
13 {
14     if (c>='A' && c<='Z')
15         /* Razlika izmedju ascii koda svakog malog i odgovarajuceg
16          * velikog slova je konstanta koja se moze sracunati izrazom
17          * 'a'-'A' (i iznosi 32). */
18         putchar(c+'a'-'A');
19     else if (c>='a' && c<='z')
20         putchar(c-'a'+'A');
21     else
22         putchar(c);
23 }
24
25 return 0;
26 }
```

Rešenje 2.68

```
#include <stdio.h>

2
int main()
3 {
4     char c;

5
6     /* Inicijalizacija brojaca na 0. */
7
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)
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==' ')
```

```

        br_b++;
36
    br_k++;
38 }

40 printf("velika: %d, mala: %d, cifre: %d, beline: %d \n", br_v, br_m
    , br_c, br_b);
    printf("suma cifara: %d\n", suma);
42
    return 0;
44 }

```

Rešenje 2.69

```

#include <stdio.h>
2
int main()
4 {
    /* Promenljiva i je brojac. */
    int n, i;
    /* Brojaci za svaki od samoglasnika. */
    int broj_a = 0, broj_e = 0, broj_i = 0, broj_o = 0, broj_u = 0;
    /* Promenljiva c je tekuci ucitani karakter. */
    char c, belina;
10

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

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

    for(i=0; i<n; i++)
    {
22         /* Prvo se cita belina koja se nalazi nakon prethodnog
            * unosa, pa tek posle procitane beline se cita uneseni
            * karakter. */
        scanf("%c%c", &belina, &c);
24

        /* Provera da li je ucitani karakter samoglasnik. */
        switch(c)
        {
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;
32             }
        }
34
    }
36 }

```

```
38     printf("samoglasnik a: %d\n", broj_a);
39     printf("samoglasnik e: %d\n", broj_e);
40     printf("samoglasnik i: %d\n", broj_i);
41     printf("samoglasnik o: %d\n", broj_o);
42     printf("samoglasnik u: %d\n", broj_u);
43
44     return 0;
45 }
46 }
```

Rešenje 2.70

```
1  /* Sa standardnog ulaza se unosi ceo broj n, a zatim i n karaktera.
2     Napisati program koji proverava da li se od
3     unetih karaktera moze napisati rec Zima. */
4
5  #include <stdio.h>
6  #include <math.h>
7
8  int main(){
9
10     int n;
11     int broj_Z, broj_i, broj_m, broj_a;
12     char novi_red, c;
13     int i;
14
15     broj_Z=0;
16     broj_i=0;
17     broj_m=0;
18     broj_a=0;
19
20     printf("Unesite broj: ");
21     scanf("%d", &n);
22
23     /* Ucitavanje karakter po karakter. */
24     for(i=0; i<n; i++){
25         printf("Unestite %d. karakter: ", i+1);
26         /* Prvo se cita belina koja se nalazi nakon prethodnog
27            * unosa, pa tek posle procitane beline se cita
28            uneseni
29            * karakter. */
30         scanf("%c%c", &novi_red, &c);
31
32         /* Analiziramo karakter */
33         switch(c){
34             case 'Z':
35                 broj_Z++;
36                 break;
37             case 'i':
38                 broj_i++;
39         }
40     }
41 }
```



```

        break;
39     case 'm':
        broj_m++;
41     break;
        case 'a':
43     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 2.71

```

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;

```

2 Kontrola toka

```
27     }
29     original=x;
30     f=1;
31
32     while (x>1)
33     {
34         f=f*x;
35         x--;
36     }
37
38     printf("Faktorijel = %lu\n", f);
39
40
41     return 0;
42 }
43 }
```

Rešenje 2.72

```
1  #include <stdio.h>
2
3  int main(){
4
5      int n;
6      float x;
7      float vrednost;
8      unsigned exp;
9
10     printf("Unesite redom brojeve x i n: ");
11     scanf("%f %d", &x, &n );
12
13     if (n < 0)
14     {
15         printf("Neispravan unos.\n");
16         return -1;
17     }
18
19     /* Pocetna vrednost stepena koji se racuna. */
20     vrednost=1;
21
22     for(exp=1; exp<=n; exp++)
23         vrednost=vrednost*x;
24
25     printf("%f\n",vrednost);
26
27     return 0;
28 }
```

Rešenje 2.73

```
1 #include <stdio.h>
3 int main(void){
5     int n, n_abs;
6     float x;
7     float vrednost;
8     unsigned exp;
9
10    printf("Unesite redom brojeve x i n: ");
11    scanf("%f %d", &x, &n );
12
13    /* Pocetna vrednost stepena koji se racuna. */
14    vrednost=1;
15
16    /* Stepenovanje. */
17    n_abs=abs(n);
18    for(exp=1; exp<=n_abs; exp++)
19        vrednost=vrednost*x;
20
21    /* Ukoliko je stepen bio negativan treba
22     * odrediti 1/x^n, sto je zapravo 1/vrednost. */
23    if(n<0){
24        printf("%.3f\n",1/vrednost);
25    }
26    else{
27        printf("%.3f\n",vrednost);
28    }
29
30    return 0;
31 }
```

Rešenje 2.74

```
1 #include <stdio.h>
3 int main()
4 {
5     int n;
6     /* Brojac. */
7     int i;
8     /* Promenljiva u kojoj se cuva suma kubova. */
9     int s;
10
11
12    printf("Unesite pozitivan ceo broj:");
13    scanf("%d", &n);
```

2 Kontrola toka

```
15     if (n < 0)
16     {
17         printf("Neispravan unos.\n");
18         return -1;
19     }
20
21     i=1;
22     s=0;
23
24     for(i=1; i<=n; i++)
25         s+=i*i*i;
26
27     printf("Suma kubova od 1 do %d: %d\n", n, s);
28     return 0;
29 }
```

Rešenje 2.75

```
#include <stdio.h>
2
3 int main()
4 {
5     int n;
6     /* Brojac. */
7     int i;
8     /* Promenljiva u kojoj se cuva suma kubova. */
9     int s;
10
11
12     printf("Unesite pozitivan ceo broj:");
13     scanf("%d", &n);
14
15     if (n < 0)
16     {
17         printf("Neispravan unos.\n");
18         return -1;
19     }
20
21     i=1;
22     s=0;
23
24     for(i=1; i<=n; i++)
25     {
26         s+=i*i*i;
27         printf("i=%d, s=%d\n", i, s);
28     }
29
30     return 0;
31 }
```

Rešenje 2.76

```
1 #include <stdio.h>
3 int main(){
4     int n, i;
5     float x, S, stepen;
7     printf("Unesite redom brojeve x i n: ");
8     scanf("%f %d", &x, &n);
9
10    if (n < 0)
11    {
12        printf("Neispravan unos.\n");
13        return -1;
14    }
15
16    /* Inicijalizacija sume. */
17    S=0;
18
19    /* Stepen promenljiva ce sadrzati vrednosti stepena x^n.
20     * Pocetna vrednost joj je 1 jer je x^0 = 1. */
21    stepen=1;
22
23    for(i=1; i<=n; i++)
24    {
25        stepen=stepen*x;
26        S=S+i*stepen;
27    }
28
29    printf("S=%f\n", S);
30
31    return 0;
32 }
```

Rešenje 2.77

```
1 #include <stdio.h>
2 int main(){
3     unsigned n, i;
4     float x, S, stepen;
5
6     printf("Unesite redom brojeve x i n: ");
7     scanf("%f %u", &x, &n);
8
9     S=1;
10    stepen=1;
11    for(i=1; i<=n; i++)
12    {
13        stepen=stepen*x;
```

2 Kontrola toka

```
    S=S+1/stepen;
15 }
17 printf("S=%f\n", S);
19 return 0;
}
```

Rešenje 2.78

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

Rešenje 2.79

```
#include <stdio.h>
#include <math.h>
2 int main(){
4     /* Promenljiva i je brojac, promenljiva S cuva izracunatu
     sumu,
     * promenljiva znak moze biti 1 ili -1 i odredjuje znak
```

```

6      * trenutnog clana sume, a promenljiva clan je tekuci clan
      niza. */
7      int i, znak;
8      float S;
9      float x, eps, clan;

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

13
14     printf("Unesite tacnost eps: ");
15     scanf("%f", &eps);

16
17     S=0;
18     clan = 1;
19     i=1;
20     znak=1;

21
22     /* Kako clanovi sume mogu biti negativni, potrebno je
       posmatrati
       * apsolutnu vrednost clana. */
23     while(fabs(clan)>eps)
24     {
25         S = S + clan;

26         /* Promena znaka. */
27         znak = - znak;
28         clan = clan*x / i;
29         clan *= znak;

30         i++;
31     }

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

33     return 0;
34 }

```

Rešenje 2.80

```

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

3
4  int main()
5  {
6      int n, i;
7      double x;
8      double stepen = 1;
9      double proizvod = 1;

10
11     printf("Unesite redom brojeve x i n: ");

```

2 Kontrola toka

```
scanf("%lf %d", &x, &n);
13
if (n <= 0)
15 {
    printf("Neispravan unos.\n");
17     return -1;
}
19
for(i=0; i<n; i++)
21 {
    stepen *= x;
23     proizvod *= 1 + cos(stepen);
}
25
printf("Proizvod = %lf\n", proizvod);
27
return 0;
29 }
```

Rešenje 2.81

```
1 #include <stdio.h>
3 int main()
{
5     int n, i;
    double Razlomak;
7
    printf("Unesite prirodan broj: ");
9     scanf("%d", &n);
11
    if (n <= 0)
    {
13         printf("Neispravan unos.\n");
        return -1;
15     }
17
    Razlomak = n;
19
    /* Razlomak se izracunava "od nazad", odnosno, kreće se
     * od najnižeg razlomka 1/n i od njega se nadalje formira
21     * sledeci, "visi" razlomak itd. Završava se kada se stigne
     * do koraka 0 + 1/R. */
23     for(i=n-1; i>= 0; i--)
        Razlomak = i + 1/Razlomak;
25
    printf("Razlomak = %lf\n", Razlomak);
27
    return 0;
29 }
```


Rešenje 2.82

```

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

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

13
15     if (n <= 0)
   {
17         printf("Neispravan unos.\n");
         return -1;
19     }

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

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

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

37         i+=2;

39     }

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

43     return 0;
   }

```

Rešenje 2.83

```
1 #include <stdio.h>
2 #include <math.h>
3
4
5 int main()
6 {
7     int n, i;
8     /* Promenljiva clan je deo proizvoda i predstavlja 1/i!. */
9     double clan;
10    double S = 1;
11
12    printf("Unesite prirodan broj: ");
13    scanf("%d", &n);
14
15    if (n <= 1)
16    {
17        printf("Neispravan unos.\n");
18        return -1;
19    }
20
21    clan = 1;
22    for(i=2; i<=n; i++)
23    {
24        clan = clan/i;
25        S *= 1 + clan;
26    }
27
28    printf("S = %lf\n", S);
29
30    return 0;
31 }
```

Rešenje 2.84

```
1 #include <stdio.h>
2 #include <math.h>
3
4
5 int main()
6 {
7     int n, i, znak = -1;
8     /* Promenljiva clan je deo proizvoda i predstavlja 1*3*5*...*i.
9     */
10    long int clan;
11    long int S = 0;
12
13    printf("Unesite prirodan broj: ");
14    scanf("%d", &n);
15
16    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 2.85

```
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;
29 }
```

Rešenje 2.98

Rešenje 2.88

```
1 #include <stdio.h>
3 int main(){
5     int n, i, j;
7     printf("Unesite broj n: ");
8     scanf("%d", &n);
9
10    /* Krstice koje iscrtavamo mozemo posmatrati kao dijagonale
11       kvadrata dimenzije n */
12
13    /* Prolazimo kroz sve vrste kvadrata */
14    for(i=1; i<=n; i++){
15
16        /* Prolazimo kroz sve kolone kvadrata */
17        for(j=1; j<=n; j++){
18
19            /* Ako se nalazimo na glavnoj ili sporednoj dijagonali */
20            if(i==j || i+j==n+1)
21                /* Stampamo zvezdu */
22                putchar('*');
23            else
24                /* Inace, stampamo blanko znak */
25                putchar(' ');
26        }
27
28        /* Nakon uspesno iscrtane vrste, stampamo znak za novi red */
29        putchar('\n');
30    }
31
32    return 0;
33 }
```

Rešenje 2.88

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

```
11  /* Krstice koje iscrtavamo mozemo posmatrati kao dijagonale
    kvadrata dimenzije n */
13
    /* Prolazimo kroz sve vrste kvadrata */
13  for(i=1; i<=n; i++){
15
    /* Prolazimo kroz sve kolone kvadrata */
    for(j=1; j<=n; j++){
17
        /* Ako se nalazimo na glavnoj ili sporednoj dijagonali */
19        if(i==j || i+j==n+1)
            /* Stampamo zvezdu */
21            putchar('*');
        else
23            /* Inace, stampamo blanko znak */
            putchar(' ');
25    }
27
    /* Nakon uspesno iscrtane vrste, stampamo znak za novi red */
    putchar('\n');
29 }
31
return 0;
}
```

Rešenje 2.98

Rešenje 2.98

Rešenje 2.91

Rešenje 2.98

Rešenje 2.98

Rešenje 2.98

Rešenje 2.98

Rešenje 2.98

Rešenje 2.98

Rešenje 2.98