

PROGRAMIRANJE 1

**Milena Vujošević Jančić, 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	29
2.1	Naredbe grananja	29
2.2	Rešenja	41
2.3	Petlje	69
2.3.1	Ispis podataka	69
2.3.2	Obrada celih brojeva, rad sa ciframa broja	71
2.3.3	Unos i obrada veće količine podatka (unos i obrada niza brojeva?, nije sjajno zbog nizova)	74
2.3.4	Rad sa karakterima	79
2.3.5	Računanje sume i proizvoda	80
2.3.6	Dvostruka petlja i ispisivanje slike	86
2.4	Rešenja	93

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, 500, 200, 100, 50, 20, 10 i 1 dinar. NAPOMENA: *Pretpostaviti da je cena proizvoda pozitivan ceo broj.*

Primer 1

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

Primer 2

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

[Rešenje 1.11]

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 decimalne.

Primer 1

```
INTERAKCIJA SA PROGRAMOM:
Unesite dimenzije sobe: 4 4 3
Unesite cenu po m2: 500
Moler treba da okreči 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 okreči 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]

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

- a i b su različiti brojevi
- a i b su parni brojevi

1 Uvodni zadaci

c) a i b su pozitivni brojevi, ne veći od 100

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

Primer 1

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

Primer 2

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

[Rešenje 1.29]

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 1.30]

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 2.36]

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
```


1.1 Rešenja

Rešenje 1.1

```

2  /*
3  Navedeni program definise funkciju koja se zove main.
4  Program moze da definise vise funkcija,
5  ali obavezno mora da definise funkciju koja se zove main i
6  izvršavanje programa uvek pocinje od te funkcije. Pored naziva,
7  zapis svake funkcije cine i povratna vrednost funkcije (u ovom
8  slucaju int), lista argumenata koje funkcija koristi (u ovom
9  slucaju funkcija nema argumenata pa se navode samo prazne
10 zagrade, ()) i telo funkcije koje je ograniceno
11 viticastim zagradama ({ i }). O ovim pojmovima bice vise reci
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 (obicno
17 ekran). Deklaracija ove funkcije data je u zaglavlju stdio.h
18 koje je potrebno ukljuciti direktivom #include na pocetku
19 samog programa.
20
21 Da bismo pokrenuli program, prvo ga moramo prevesti u izvrsnu
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, bice generisana
29 izvrsna datoteka pod nazivom a.out koja se pokrece navodjenjem
30 sledece naredbe:
31 ./a.out
32
33 Ukoliko je bilo gresaka prilikom prevodjenja, one se moraju
34 otkloniti a postupak prevodjenja se mora ponoviti.
35 */
36 #include<stdio.h>
37
38 int main()
39 {
40     /* printf: funkcija pomocu koje se vrši ispis */
41     /* Specijalni karakter \n : prelazak u novi red */
42     /* Svaka naredba završava se karakterom ; */
43     printf("Zdravo svima!\n");
44
45     /* Povratna vrednost 0 se obicno koristi da oznaci
46        da je prilikom izvršavanja programa sve proslo

```

1 Uvodni zadaci

```
        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).

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

28

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

38

40    return 0;
}
```

Rešenje 1.3

```
2 #include<stdio.h>
4 int main()
5 {
6     /* Promenljive istog tipa mogu se deklarirati jedna za drugom. */
7     int x, y, rezultat;
8
9
10    printf("Unesi vrednost celobrojne promenljive x: ");
11    scanf("%d", &x);
12
13    printf("Unesi vrednost celobrojne promenljive y: ");
14    scanf("%d", &y);
15
16    /* Dodeljujemo vrednost promenljivoj rezultat. */
17    rezultat = x+y;
18    printf("%d + %d = %d\n", x,y,rezultat);
19
20    /*
21     * Mozemo ispisivati direktno vrednost izraza x-y i bez
22     * njegovog dodeljivanja posebnoj promenljivoj
23     */
24    printf("%d - %d = %d\n",x,y,x-y);
25    printf("%d * %d = %d\n",x,y,x*y);
26
27    /*
28     * Kada bilo koju aritmeticku operaciju primenimo na dve
29     * promenljive istog tipa (u ovom slucaju dva celobrojne
30     * promenljive), rezultat ce biti tog istog tipa. Specijalno,
31     * za operaciju deljenja: kada operator / primenimo na dva
32     * celobrojna argumenta x i y, kao rezultat dobijemo ceo deo
33     * pri deljenju broja x brojem y, a ne kolicnik.
34     * Na primer, rezultat primene operatora / na 7 i 2
35     * je 3, a ne 3.5.
36     */
37    printf("%d / %d = %d\n",x,y,x/y);
38
39    /*
40     * Operator % izracunava ostatak pri celobrojnom deljenju
41     * dve celobrojne promenljive. Na primer, 7%2 ima vrednost 1
42     * (jer je 7=3*2+1).
43     * Da bismo odstampali karakter %, u naredbi printf pisemo %%
44     */
45    printf("%d %% %d = %d\n",x,y,x%y);
46
47    return 0;
48 }
```

Rešenje 1.4

1 Uvodni zadaci

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
5  int main()
6  {
7      /*
8       * S obzirom da su sve promenljive pozitivni celi brojevi,
9       * koristimo tip unsigned int (skraceno unsigned)
10     */
11     unsigned cena;
12     unsigned kolicina;
13     unsigned iznos;
14     unsigned kusur;
15
16     /*
17      * Ucitavamo potrebne podatke. Unutar jednog scanf-a mozemo
18      * ucitati vise podataka odjednom. Za svaki treba navesti
19      * odgovarajuci format za tip podataka koji se unosi
20      * (%u za unsigned).
21     */
22     printf("Unesite cenu, kolicinu i iznos: ");
23     scanf("%u%u%u", &cena, &kolicina, &iznos);
24
25     /* Izracunavamo kusur: */
26     kusur=iznos - kolicina*cena;
27
28     /* I ispisujemo trazenu vrednost: */
29     printf("Kusur je %u dinara.\n", kusur);
30
31     return 0;
32 }
```

Rešenje 1.7

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

```

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

```

Rešenje 1.8

```

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;

```

1 Uvodni zadaci

```
18     y=p;
    printf("posle zamene: x=%d, y=%d\n",x,y);
    return 0;
20 }
```

Rešenje 1.10

```
1  #include <stdio.h>
3  int main()
4  {
5      /* S obzirom da broj treba da bude pozitivan,
6         koristimo tip unsigned.
7      */
8      unsigned x;
9
10     /*
11      Promenljive koje cuvaju cifre treba da budu
12      najmanjeg celobrojnog tipa jer nece sadrzati
13      druge vrednosti osim jednocifrenih celih
14      brojeva. Zbog toga za njih biramo tip char.
15
16     */
17     char cifra_jedinice;
18     char cifra_desetice;
19     char cifra_stotine;
20
21     printf("Unesi trocifreni broj:");
22     scanf("%u", &x);
23
24     /*
25      Na primer, neka je uneti broj 374. Potrebno je da koriscenjem
26      racunskih operacija za rad sa celim brojevima pristupimo
27      njegovoj cifri jedinice, cifri desetice i cifri stotine.
28
29      Primetimo najpre sledece:
30      374/10 = 37
31      374%10 = 4
32      Dakle, operacijama celobrojnog deljenja i ostatka pri deljenju
33      mozemo iz svakog broja izdvojiti njegovu poslednju cifru (u
34      ovom slucaju 4) i broj sastavljen od svih cifara osim poslednje
35      (u ovom slucaju 37).
36
37      Cifri jedinice sada lako pristupamo koriscenjem ostatka pri
38      deljenju sa 10. Ona iznosi upravo 4.
39
40      Pri trazanju cifre desetice mozemo ponovo primeniti princip
41      izdvajanja poslednje cifre kao ostatka pri deljenju sa 10.
42      Razlika je sto ne mozemo deseticu izdvojiti ako primenimo %10
43      na 374 (tada 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.

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

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

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

69     /*
    2. nacin, bez uvođenja dodatnih promenljivih cifra_jedinice,
71     cifra_desetice i cifra_stotine:

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

```

Rešenje 1.11

```

1     #include <stdio.h>

3
4     int main()
5     {
6         unsigned x;
7         printf("Unesi cenu:");
8         scanf("%u", &x);
9
10        /*
11        Na primer, neka je uneta cena 8347 dinara.
        Vrednost x/5000 predstavlja broj novcanica
13        od 5000 dinara pomocu kojih mozemo sakupiti
        celokupnu sumu. 8347 celobrojno deljeno sa
15        5000 (operacija / nad celim brojevima) iznosi 1.
        */
    }

```

1 Uvodni zadaci

```
17  */
    printf("%u=%u*5000+ ", x,x/5000);
    /*
19     Potrebna nam je 1 novcanica od
        5000 dinara, a koliko nam je potrebno ostalih
21     novcanica? Za to moramo pristupiti preostaloj
        sumi. Jedan nacin je da nadjemo ostatak pri deljenju
23     unete vrednosti x (u primeru 8347) sa 5000 (operacija %).
        On iznosi 3347. Ovu vrednost dodeljujemo promeljivoj x.
25  */
    x=x%5000;
27
    /*
29     Nastavljamo postupak trazenjem broja novcanica
        od 2000 dinara i redom za ostale monete.
31  */
    printf("%u*2000 +", x/2000);
33    x=x%2000;
    printf("%u*1000 +", x/1000);
35    x=x%1000;
    printf("%u*500 +", x/500);
37    x=x%500;
    printf("%u*200 +", x/200);
39    x=x%200;
    printf("%u*100 +", x/100);
41    x=x%100;
    printf("%u*50 +",x/50);
43    x=x%50;
    printf("%u*20 +", x/20);
45    x=x%20;
    printf("%u*10 +", x/10);
47    x=x%10;
    printf("%u*1\n", x);
49    return 0;
}
```

Rešenje 1.12

```
1  #include <stdio.h>
3  int main()
4  {
5      unsigned x;
        unsigned obrnuto_x;
7
8      char cifra_jedinice;
9      char cifra_desetice;
        char cifra_stotine;
11
12     printf("Unesi trocifreni broj:");
13     scanf("%u", &x);
```



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

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

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

25  return 0;
}
```

Rešenje 1.13

```
1  #include <stdio.h>

3  int main(){

5      unsigned n, broj_obrnuto, broj_zamena;
6      char j, d, s, h;
7      int proizvod_cifara, razlika_cifara, suma_kvadrata;

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

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;
17     s=(n/100)%10;
18     h=n/1000;

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

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

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

31     /* Odredjujemo broj zapisan istim ciframa ali u obrnutom redosledu
32        */
33     broj_obrnuto= j*1000+d*100+s*10+h;
34     printf("Broj u obrnutom poretku: %u\n", broj_obrnuto);
}
```

1 Uvodni zadaci

```
35 |
37 | /* Odredjujemo broj u kojem su cifra jedinica i
   |     cifra stotina zamenile mesta
39 | */
   | broj_zamena=h*1000+j*100+d*10+s;
41 | printf("Broj sa zamenjenom cifrom jedinica i stotina: %u\n",
   |     broj_zamena);
   |
43 |
   | return 0;
45 | }
```

Rešenje 1.15

```
#include <stdio.h>
2
int main()
4 {
   |
   | unsigned broj, novibroj;
   | unsigned levi, desni, m;
   |
   | printf("Unesite pozitivan prirodan broj: ");
10 | scanf("%u", &broj);
   | printf("Unesite pozitivan dvocifreni broj:");
12 | scanf("%u", &m);
   |
14 | /*
   |     Na primer, za unete broj 12345 i 67, potrebno je ubaciti
16 |     67 izmedju cifre hiljade (2) i cifre stotine (3). Rezultat
   |     je 12|67|345. Potrebno je da razdvojimo uneti broj na levi
18 |     i desni deo: 12 i 345 i izmedju njih umetnemo broj m
   | */
20 |
   | levi = broj/1000;
22 | desni = broj%1000;
   |
24 | /*
   |     Kada levi deo pomnozimo sa 100 000, dobijamo 1 200 000
26 |     Kada m pomnozimo sa 1000, dobijamo           67 000
   |     Dobijene vrednosti saberemo sa desnim delom           345
28 |                                     -----
   |     Konacan rezultat:                                     1 267 345
30 |
   | */
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 */
6     float in;
7     float cm;
9     printf("Unesi broj inca: ");
11    /* "%f" - format za unos/ispis float promenljivih */
12    scanf("%f", &in);
13
14    /* 1 inch = 2.54 cm */
15    cm = in*2.54;
17    /* "%.2f" - ispis realne promenljive na 4 decimale */
18    printf("%.2f in = %.2f cm\n", in, cm);
19
20    return 0;
21 }
```

Rešenje 1.17 Zadatak se rešava analogno zadatku 1.16.

Rešenje 1.18 Zadatak se rešava analogno zadatku 1.16.

Rešenje 1.19 Zadatak se rešava 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     /* Ucitavamo potrebne podatke */
10    printf("Unesite duzine stranica pravougaonika: ");
11    scanf("%f%f", &a, &b);
12 }
```

1 Uvodni zadaci

```
14  /* Obim */
    obim=2*(a+b);

16  /* Povrsina */
    povrsina=a*b;

18

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

22

24  /* Završavamo sa programom */
    return 0;
}
```

Rešenje 1.22

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

3

   /* Zaglavlje math.h sadrzi deklaracije velikog broja
5     matematičkih funkcija i konstanti. U ovom zadatku se
       koristi zbog konstante pi (M_PI)

7

   Ukoliko se koristi i neka funkcija matematičke
9     biblioteke, za prevodjenje je neophodno uključiti
       opciju -lm
11    npr. gcc primer.c -lm
   */

13  int main()
   {
15     float r;
       float O;
17     float P;
       printf("Unesite dužinu poluprečnika kruga:");
19     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;
6    float P,0;

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

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

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

```

Rešenje 1.24

```

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

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

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

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

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

18    /* 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 {

```

1 Uvodni zadaci

```
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
17     koliko iznosi a+b+c celobrojno podeljeno sa 3. To znaci da
19     ce za unete vrednosti 11, 5 i 4 aritmeticka sredina biti
21     6.00. Zaista, zbir 11+5+4 iznosi 20, a kada 20 celobrojno
23     podelimo sa 3 dobijamo 6. Ovu celobrojnu vrednost dodeljujemo
25     realnoj promenljivoj as, cime se ona konvertuje u 6.000000 i
27     ispisujemo je zaokruzenu na dve decimalne. Izlaz iz programa bi
    bio pogresan: 6.00.

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

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

31
33 /*
35     Trazeni kolicnik mozemo dobiti na razne nacine:
37     as=1.0*(a+b+c)/3;
39     ili
41     as=(0.0+a+b+c)/3;
43     ili
45     as=((float)(a+b+c))/3;
47     itd.
49 */

51 printf("Aritmeticka sredina unetih brojeva je %.2f\n", as);
53 return 0;
55 }
```

Rešenje 1.27

```
1 #include <stdio.h>
3
5 int main(){
7     unsigned duzina, sirina, visina;
9     unsigned cena;
11    float površina_za_krecenje;
13    float ukupna_cena;
```

```

11  /* Ucitavamo duzinu, sirinu i visinu sobe */
    printf("Unesite dimenzije sobe: ");
    scanf("%u%u%u", &duzina, &sirina, &visina);
13
    /* Ucitavamo cenu krecenja */
15  printf("Unesite cenu po m2: ");
    scanf("%u", &cena);
17
    /* Povrsina za krecenje odgovara površini kvadra -
19     bez poda jer se on ne kreci */
    povrsina_za_krecenje=0.8*(duzina*sirina+
21                             2*duzina*visina+
                             2*sirina*visina);
23  ukupna_cena=povrsina_za_krecenje*cena;

25  /* Ispisujemo trazene podatke */
    printf("Moler treba da okreći %.2f m2\n",
27         povrsina_za_krecenje);

29  printf("Cena krecenja je %.2f\n", ukupna_cena);

31  /* Završavamo sa programom */
    return 0;
33 }

```

Rešenje 1.28

```

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

5  int main()
6  {
7      unsigned x, p;
8      char c;
9      unsigned levo, desno;
10     unsigned novo_x;
11
12     /*
13      Ucitavamo potrebne vrednosti. Sa unosom podataka tipa
14      char moramo biti pazljivi i o tome ce vise biti reci
15      u narednim poglavljima kod zadatka za rad sa funkcijama
16      getchar i putchar. Zbog toga cemo ovde za ucitavanje
17      podataka zatraziti da podatke razdvajamo blanko znakovima
18      (a ne znakom za novi red, zarezom ili nekim drugim separatorom).
19      Ovaj zahtev navodimo u format stringu funkcije scanf tako sto
20      specifikatore promenljivih razdvajamo blanko znakovima.
21
22      Ukoliko specifikatore promenljivih u format stringu pisemo
23      spojeno, tada ih prilikom unosa mozemo razdvojiti bilo kojim
24      karakterom. Zbog toga blanko znake u format stringu funkcije

```

1 Uvodni zadaci

```
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
49     /* Odredjujemo deo broja koji se nalazi levo od pozicije p */
    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
55     /* Ispisujemo dobijenu vrednost */
    printf("Rezultat je: %u\\n", novo_x);

57     /* Završavamo sa programom */
    return 0;
59 }
}
```

Rešenje 1.29

```
1  #include <stdio.h>

3  int main()
    {
5      int a,b,resultata,resultatb,resultatc;

7      printf("Unesite dva cela broja:");
      scanf("%d%d", &a,&b);
9
11     /*
        Izraz a!=b ima vrednost 1 ako je ova
        relacija tacna, a 0 ako je netacna
    */
}
```



```

13  */
    rezultata = 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  rezultatb = (a%2==0 && b%2==0);
24
25  /*
26     Izraz a>0 && a<=100 && b>0 && b<=100 konjunkcija
27     koja se sastoji od cetiri konjunkata. Svaki od
28     konjunkata je izraz koji sadrzi relacioni operator
29     i ima vrednost 1 ako relacija vazi a 0 ako ne vazi
30  */
31
32  rezultatc = (a>0 && a<=100 && b>0 && b<=100);
33
34  printf("a) rezultat=%d\n", rezultata);
35  printf("b) rezultat=%d\n", rezultatb);
36  printf("c) rezultat=%d\n", rezultatc);
37
38  return 0;
39 }

```

Rešenje 1.30

```

1  #include <stdio.h>
2
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 vrednsot 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     */

```

1 Uvodni zadaci

```
23      */
24      max = (a>b) ? a : b;
25
26      printf("Maksimum je %d\n", max);
27  }
```

Rešenje 2.36 Zadatak se rešava analogno zadatku 2.36

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     min = (a<b) ? a : b;
15     max = (a>b) ? a : b;
16
17     /*
18      Racunamo vrednost promenljive rez
19     */
20     rez = (min+0.5)/(1+max*max);
21
22     printf("Rezultat je %.2f\n", rez);
23
24     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 ispisi 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 $[a_1, b_1]$ i $[a_2, b_2]$, program treba da odredi:

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

Primer 1

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

Primer 2

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

Zadatak 2.20 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]

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 uređjene neopadajuće.
```

Primer 2

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

Primer 3

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

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) ugrožava polje (m, n)
- c) da li konj sa (k, l) ugrožava 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);
9
10     /* Promenljiva min dobija vrednost promenljive a. */
11     min=a;
12
13     /*
14      Ako je b<a, promenljiva min ce promeniti vrednost
15      tj. bice joj dodeljena vrednost promenljive b. U
16      suprotnom, vrednost ostaje ista.
17     */
18
19     if (b<a)
20         min = b;
21
22     printf("Minimum je %d\n",min);
23
24     return 0;
25 }
```

2 Kontrola toka

Rešenje 2.2 Rešenje je analogno rešenju broj 2.1.

Rešenje 2.3

```
1 #include<stdio.h>
3 int main()
4 {
5     float x;
6     float apsolutno_x;
7
8     printf("Unesite jedan realan broj:");
9     scanf("%f",&x);
10
11     apsolutno_x=x;
12     if (x<0)
13         apsolutno_x=-x;
14
15     printf("Njegova apsolutna vrednost je %.2f\n",apsolutno_x);
16
17     /*
18      2. nacin, pomocu funkcije fabs za koju je neophodno
19      ukljuciti zaglavlje math.h:
20      apsolutno_x=fabs(x);
21     */
22     return 0;
23 }
```

Rešenje 2.4

```
1 #include <stdio.h>
3 int main()
4 {
5     int x;
6     float rx;
7
8     printf("Unesite jedan ceo broj:");
9     scanf("%d",&x);
10
11     /*
12      Obratiti paznju:
13      x=0 - relacija jednakosti
14           (da li je promenljiva x jednaka nuli)
15      x=0 - naredba dodele
16           (promenljiva x dobija vrednost nula)
17     */
18 }
```



```

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

```

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     */
20
21     if (a>0)

```

2 Kontrola toka

```

    s=s+a;
23
    /* s+=b je skraceni zapis za s=s+b */
25
    if (b>0)
27        s+=b;

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

31    printf("Suma unetih pozitivnih brojeva: %d\n",s);
33    return 0;
}
```

Rešenje 2.6

```

#include <stdio.h>
2
int main()
4
{
    unsigned a,b,c;
    unsigned min;
    unsigned cena_bez_popusta, cena_sa_popustom;

8
    printf("Unesite cene tri artikla:");
10    scanf("%u%u%u",&a,&b,&c);

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

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

26    /*
        Ako je treci broj manji od minimuma, to znaci da
28    promenljiva min ne sadrzi najmanji broj. Dodeljujemo
        joj vrednost treceg broja.
30    */
    if(min>c)
32        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",
37           cena_sa_popustom, cena_bez_popusta-cena_sa_popustom);
40     return 0;
}
```

Rešenje 2.7

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

2 Kontrola toka

```
43     max=s;
45     if(h>max)
46         max=h;
47
48     /* II nacin:
49     * if(j>d && j>s && j>h)
50     *   max=j;
51     * if(d>j && d>s && d>h)
52     *   max=d;
53     * if(s>j && s>d && s>h)
54     *   max=s;
55     * if(h>j && h>d && h>s)
56     *   max=h;
57     */
58
59     /* Ispisujemo rezultat */
60     printf("Najveca cifra je: %d\n", max);
61
62     return 0;
63 }
```

Rešenje 2.8

```
#include<stdio.h>
2
3 int main()
4 {
5     int sati;
6     int minuti;
7     int preostali_sati;
8     int preostali_minuti;
9
10    /*
11     * Ukoliko naredbu printf zelimo da napisemo u dva
12     * reda, i tom prilikom prekidamo deo pod navodnicima,
13     * to mozemo uraditi navodjenjem navodnika na kraju
14     * prvog i na pocetku narednog reda:
15     */
16
17    printf("Unesite vreme (broj sati u intervalu [0,24),\n"
18           "broj minuta u intervalu [0,60)):");
19    scanf("%d%d",&sati,&minuti);
20
21    /*
22     * U slucaju da je unos neispravan, ispisujemo
23     * poruku o gresci i prekidamo dalje izvršavanje
24     * programa.
25
26     * Uslov u if naredbi je disjunkcija (operator ||)
27     * sastavljena od 4 disjunkata. Svaki od njih je
```

```

28     izraz sa relacionim operatorom i ima vrednost
30     1 ako je izraz tacan i 0 u suprotnom. Da bi
32     disjunkcija bila tacna, bar jedan od disjunkata
34     mora da bude tacan. Zbog lenjog izracunavanja,
36     vrednost disjunkata ce biti racunata do vrednosti
38     prvog disjunkta koji je tacan. To je znak da
40     je uslov u if naredbi ispunjen i nema potrebe
42     racunati vrednosti drugih disjunkata.
44     */
46     if (sati>24 || sati<0 || minuti>59 || minuti<0)
48     {
50         printf("Neispravan unos.\n");
52         return -1;
54     }
56     preostali_sati = 24-sati-1;
58     preostali_minuti = 60-minuti;
60
62     if (preostali_minuti==60)
64     {
66         preostali_sati++;
68         preostali_minuti=0;
70     }
72
74     /*
76     Ukoliko naredbu printf zelimo da napisemo u dva
78     reda i pritom ne prekidamo deo pod navodnicima,
80     to mozemo uraditi bez navodjenja dodatnih karaktera:
82     */
84     printf("Do ponoci je ostalo %d sati i %d minuta\n",
86           preostali_sati, preostali_minuti);
88
90     return 0;
92 }

```

Rešenje 2.9

```

1  #include <stdio.h>
2
3  int main()
4  {
5      char c;
6      printf("Unesite jedan karakter:");
7      scanf("%c", &c);
8
9      printf("Uneti karakter: %c, njegov ASCII kod: %d\n", c, c);
10
11     /*
12     Razlika izmedju ASCII koda svakog malog i odgovarajuceg
13     velikog slova je konstanta koja se moze sracunati

```

2 Kontrola toka

```
14     izrazom 'a'-'A' (i iznosi 32)
15     */
16
17     if (c>='A' && c<='Z')
18         printf("odgovarajuće malo slovo: %c, njegov ASCII kod: ",
19                "%d\n", c+('a'-'A'), c+('a'-'A'));
20
21     if (c>='a' && c<='z')
22         printf("odgovarajuće veliko slovo: %c, njegov ASCII kod: ",
23                "%d\n", c-('a'-'A'), c-('a'-'A'));
24
25     return 0;
26 }
```

Rešenje 2.10

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

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

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

Rešenje 2.11

```
1  #include <stdio.h>
    #include <ctype.h>
3
4  int main()
5  {
6      /*
7       Broj pojavljivanja slova a i A se
        inicijalizuje na 0
9      */
        int br_a = 0;
11
12     /*
13      Funkcija getchar ucitava jedan karakter.
        Njena povratna vrednost je ASCII kod
15      ucitanog karaktera.

17      Funkcija tolower za dati karakter vraca:
        - odgovarajuce 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 ++
        */
27     if (tolower(getchar()) == 'a')
        br_a++;
29     if (tolower(getchar()) == 'a')
        br_a++;
31     if (tolower(getchar()) == 'a')
        br_a++;
33     if (tolower(getchar()) == 'a')
        br_a++;
35     if (tolower(getchar()) == 'a')
        br_a++;
}
```

2 Kontrola toka

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

Rešenje 2.98

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



```
    printf("Godina nije prestupna\n");
13     return 0;
15 }
```

Rešenje 2.14

```
1  #include <stdio.h>
   #include <stdlib.h> /*abs*/
3
   int main()
5  {
   int n, j, d, s;
7
   /* Ucitavamo broj */
9   printf("Unesite broj: ");
   scanf("%d", &n);
11
   /*
13    Uzimamo apsolutnu vrednost broja za slucaj da je
    uneti broj negativan
15   */
   n = abs(n);
17
   /*
19    Ako broj nije trocifren, izdajemo poruku o gresci i
    prekidamo dalje izvršavanje programa
21   */
   if(n<100 || n>999)
23   {
       printf("Greska: Niste uneli trocifren broj!\n");
25       return -1;
   }
27
   /*
29    Izdvajamo cifre broja:
    j -jedinice, d - desetice, s - stotine
31   */
   j=n%10;
33   d=(n/10)%10;
   s=n/100;
35
   /* Proveravamo da li je broj Armstrongov */
37   if(n==j*j*j+d*d*d+s*s*s)
       printf("Broj je Armstrongov.\n");
39   else
       printf("Broj nije Armstrongov.\n");
41
   return 0;
43 }
```

Rešenje 2.15

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

```

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

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

65  return 0;
}

```

Rešenje 2.16

```

1  #include <stdio.h>

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

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

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

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

25     if (b > max)
27     {
          max = b;
29     }

```

```
31     stepenMax = 10;
32 }
33 if (b < min)
34 {
35     min = b;
36     stepenMin = 10;
37 }
38
39 if (c > max)
40 {
41     max = c;
42     stepenMax = 100;
43 }
44 if (c < min)
45 {
46     min = c;
47     stepenMin = 100;
48 }
49
50 if (d > max)
51 {
52     max = d;
53     stepenMax = 1000;
54 }
55 if (d < min)
56 {
57     min = d;
58     stepenMin = 1000;
59 }
60
61 int rez;
62 /* Ideja:
63    4179, mesta menjamo tako sto oduzmemo 9 i dodamo 1,
64    ^ ^  odnosno oduzemo 100 i dodamo 900 */
65
66 if (broj > 0)
67     rez = broj - max*stepenMax + min*stepenMax
68             - min*stepenMin + max*stepenMin;
69 else
70     rez = broj + max*stepenMax - min*stepenMax
71             + min*stepenMin - max*stepenMin;
72
73 printf("%d\n",rez);
74
75 return 0;
76 }
77 }
```

Rešenje 2.20

```

1 #include <stdio.h>
3 int main()
4 {
5     int x;
6     printf("Unesite jedan ceo broj:");
7     scanf("%d",&x);
9
10    if (x==0)
11        printf("Broj je jednak nuli\n");
12    else if (x<0)
13        printf("Broj je manji od nule\n");
14    else
15        printf("Broj je veci od nule\n");
16
17    return 0;
18 }

```

Rešenje 2.21

```

1 #include <stdio.h>
2 #include <math.h>
3
4 int main()
5 {
6     float a,b,c;
7     float D;
8     float x1,x2;
9     printf("Unesite koeficijente A, B i C:");
10    scanf("%f%f%f",&a,&b,&c);
11
12    /*
13     Proveravamo da li je kvadratna jednacina
14     korektno zadata.
15    */
16    if (a==0)
17        if (b==0)
18            /* slucaj a==0 && b==0 && c==0 */
19            if(c==0)
20                printf("Jednacina ima beskonacno mnogo resenja\n");
21            /* slucaj a==0 && b==0 && c!=0 */
22            else
23                printf("Jednacina nema resenja\n");
24            /* slucaj a==0 && b!=0 */
25            else
26            {
27                x1=-c/b;
28                printf("Jednacina ima jedinstveno realno resenje %.2f\n",
29                        x1);
30            }
31    /* slucaj a!=0 */

```

```
33     else
34     {
35         D=b*b-4*a*c;
36         if (D<0)
37             printf("Jednacina nema realnih resenja\n");
38         else if (D>0)
39         {
40             /* funkcija sqrt nalazi se u biblioteci math.h
41              (prevodjenje sa -lm opcijom) */
42             x1 = (-b+sqrt(D))/(2*a);
43             x2 = (-b-sqrt(D))/(2*a);
44             printf("Jednacina ima dva razlicita realna resenja %.2f ",
45                    "i %.2f\n",x1,x2);
46         }
47         else
48         {
49             x1 = (-b)/(2*a);
50             printf("Jednacina ima jedinstveno realno resenje %.2f\n",x1);
51         }
52     }
53     return 0;
54 }
```

Rešenje 2.22

```
1 #include <stdio.h>
2 #include <stdlib.h>
3
4 int main()
5 {
6     int x;
7     char c1;
8     char c10;
9     char c100;
10    char c1000;
11
12    printf("Unesi jedan cetvorocifreni broj:");
13    scanf("%d", &x);
14
15    /*
16     Uzimamo apsolutnu vrednost unetog broja kako u slucaju
17     da je negativan ne bismo za cifre dobili negativne brojeve.
18     Funkcija abs nalazi se u zaglavlju stdlib.h
19    */
20    x=abs(x);
21
22    if (x<1000 || x>9999)
23    {
24        printf("Uneti broj nije cetvorocifren\n");
25        return -1;
26    }
27 }
```

```

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

```

Rešenje 2.23

```

1  #include <stdio.h>

3  int main(){

5     char c;

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

9

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

23   return 0;
}

```

Rešenje 2.24

```

1  #include <stdio.h>

```

```
3 int main(){
4     int k, n, broj;
5
6     printf("Unesite k: ");
7     scanf("%d", &k);
8
9     if(k<10){
10        /* Trazi se jednocifren broj */
11        printf("Na %d-toj poziciji je broj %d.\n", k, k);
12    }
13    else
14        /* Trazi se dvocifreni broj */
15        if(k>=10 && k<=189){
16
17            /* Odredjujemo broj dvocifrenih brojeva koji se mogu
18               zapisati pomocu k cifara */
19
20            if(k%2!=0){
21                /*
22                 Ako je k neparan broj, zapisan je ceo broj
23                 dvocifrenih brojeva
24
25                 9 oduzimamo jer je 9 broj cifara potrebnih za
26                 zapis jednocifrenih brojeva
27                 */
28                n=(k-9)/2;
29
30                /* Broj o kojem se radi je */
31                broj=9+n;
32
33                /* Ujedno, za neparno k se trazi cifra jedinica
34                   izdvojenog broja */
35                printf("Na %d-toj poziciji je broj %d.\n", k, broj%10);
36            }
37            else{
38                /* Ako je k paran broj, zapisan je ceo broj
39                   dvocifrenih brojeva i zapoceto je sa zapisom sledeceg
40
41                   9 oduzimamo jer je 9 broj cifara potrebnih za zapis
42                   jednocifrenih brojeva
43                 */
44                n=(k-9)/2 +1;
45
46                /* Broj o kojem se radi je */
47                broj= 9 + n;
48
49                /* Ujedno, za parno k se trazi cifra desetica
50                   izdvojenog broja */
51                printf("Na %d-toj poziciji je broj %d.\n", k, broj/10);
52            }
53        }
54    }
```



```

55     }
    else{
57         printf("Greska: Nedozvoljena vrednost broja k!\n");
    }

59     return 0;
61 }

```

Rešenje 2.25

```

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

3
   int main(){
5       float x;
       int k;
7       float F;

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

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

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

35     /* Ispisujemo rezultat */
       printf("%f\n", F);

37     return 0;
39 }

```

Rešenje 2.26

```
1  #include <stdio.h>
3
5  int main(){
7      int dan;
9      printf("Unesite broj: ");
10     scanf("%d", &dan);
12
13     switch(dan)
14     {
15         case 1:
16             printf("ponedeljak\n");
17             break;
18         case 2:
19             printf("utorak\n");
20             break;
21         case 3:
22             printf("sreda\n");
23             break;
24         case 4:
25             printf("cetvrtak\n");
26             break;
27         case 5:
28             printf("petak\n");
29             break;
30         case 6:
31             printf("subota\n");
32             break;
33         case 7:
34             printf("nedelja\n");
35             break;
36         default:
37             printf("Greska: nedozvoljeni unos!\n");
38     }
39     return 0;
40 }
```

Rešenje 2.27

```
1  #include <stdio.h>
3
5  int main()
6  {
7      char c;
8      printf("Unesite jedan karakter:");
```

```
7  scanf("%c", &c);
9  /*
   Da bismo utvrditi da li je karakter samoglasnik,
11  neophodno je proveriti da li odgovara nekom od
   sledecih karaktera: A,E,I,O,U,a,e,i,o,u
13  */
   switch(c)
15  {
       case 'A' :
17       case 'E' :
       case 'I' :
19       case 'O' :
       case 'U' :
21       case 'a' :
       case 'e' :
23       case 'i' :
       case 'o' :
25       case 'u' : printf("Uneti karakter je samoglasnik\n");
                   break;
27       default : printf("Uneti karakter nije samoglasnik\n");
                   break;
29  }
31  return 0;
}
```

Rešenje 2.28

```
1  #include <stdio.h>
3  int main()
   {
5
       char op;
7       int x, y;
9
       printf("Unesite operator i dva cela broja: ");
       scanf("%c %d %d", &op, &x, &y);
11
       switch(op){
13         case '+':
           printf("Rezultat je: %d\n", x+y);
15         break;
           case '-':
17         printf("Rezultat je: %d\n", x-y);
           break;
19         case '*':
           printf("Rezultat je: %d\n", x*y);
21         break;
           case '/':
```

```
23     if(y==0)
24         printf("Greska: deljenje nulom nije dozvoljeno!\n");
25     else
26         printf("Rezultat je: %f\n", x*1.0/y);
27     break;
28     case '%':
29         printf("Rezultat je: %d\n", x%y);
30         break;
31     default:
32         printf("Greska: nepoznat operator!\n");
33 }
34
35 return 0;
36 }
```

Rešenje 2.29

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

```
35     printf("leto\n");
36     break;
37 case 7:
38 case 8:
39     printf("leto\n");
40     break;
41 case 9:
42     if (d<23)
43         printf("leto\n");
44     else
45         printf("jesen\n");
46     break;
47 case 10:
48 case 11:
49     printf("jesen\n");
50     break;
51 case 12:
52     if (d<22)
53         printf("jesen\n");
54     else
55         printf("zima\n");
56 }
57 return 0;
58 }
```

Rešenje 2.30

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

```
25     else
26         prestupna = 0;
27
28     printf("Unesite redni broj meseca: ");
29     scanf("%d",&mesec);
30
31     switch(mesec)
32     {
33         case 1: printf("Januar, 31 dan\n");
34                 break;
35         case 2:
36             if(prestupna)
37                 printf("Februar, 29 dana\n");
38             else
39                 printf("Februar, 28 dana\n");
40             break;
41
42         case 3: printf("Mart, 31 dan\n");
43                 break;
44         case 4: printf("April, 30 dana\n");
45                 break;
46         case 5: printf("Maj, 31 dan\n");
47                 break;
48         case 6: printf("Jun, 30 dana\n");
49                 break;
50         case 7: printf("Jul, 31 dan\n");
51                 break;
52         case 8: printf("Avgust, 31 dan\n");
53                 break;
54         case 9: printf("Septembar, 30 dana\n");
55                 break;
56         case 10: printf("Oktobar, 31 dan\n");
57                 break;
58         case 11: printf("Novembar, 30 dana\n");
59                 break;
60         case 12: printf("Decembar, 31 dan\n");
61                 break;
62         default: printf("Lose unet redni broj meseca!\n");
63     }
64
65     return 0;
66 }
```

Rešenje 2.31

```
1  #include <stdio.h>
2
3  int main(){
4      int dan, mesec, godina, dozvoljen_broj_dana;
5
6      /* Citamo datum */
```

```
7      printf("Unesite datum: ");
      scanf("%d.%d.%d", &dan, &mesec, &godina);

9

11     /* Proveravamo godinu */
    if(godina<0){
13         printf("Datum nije korektan (neispravna godina)!\n");
        return 0;
    }

15

17     /* Proveravamo mesec */
    if(mesec<1 || mesec>12){
19         printf("Datum nije korektan (neispravan mesec)!\n");
        return 0;
    }

21

23     /* Ako je mesec korektan, proveravamo broj dana */
    switch(mesec){
25         case 1:
27         case 3:
29         case 5:
31         case 7:
33         case 8:
35         case 10:
37         case 12:
            /* Dozvoljeni broj dana za januar, mart, maj, jul, avgust,
               * oktobar i decembar je 31 */
            dozvoljen_broj_dana=31;
            break;
39         case 2:
            /* Proveravamo da li je godina prestupna */
            if(godina%4==0 && godina%100!=0 || godina%400==0)
            /* Ako jeste, dozvoljeni broj dana za februar je 29 */
            dozvoljen_broj_dana=29;
            else
            /* Ako nije, dozvoljeni broj dana za februar je 28 */
            dozvoljen_broj_dana=28;
            break;
43         case 4:
45         case 6:
47         case 9:
49         case 11:
            /* Dozvoljeni broj dana za april, jun, septembar i
               novembar je 30 */
            dozvoljen_broj_dana=30;
            break;
51     }

53     /* Proveravamo dan */
    if(dan<0 || dan>dozvoljen_broj_dana){
55         printf("Datum nije korektan (neispravan dan)!\n");
        return 0;
    }

57 }
```

2 Kontrola toka

```
59      /* Sve provere su ispunjene pa zakljucujemo
        da je datum korektan */
61      printf("Ispravan datum!\n");

63      return 0;
    }
```

Rešenje 2.32

```
1  #include <stdio.h>

3  int main(){
    int dan, mesec, godina;
5     int prethodni_dan, prethodni_mesec, prethodni_godina;

7     /* Citamo datum */
    printf("Unesite datum: ");
9     scanf("%d.%d.%d", &dan, &mesec, &godina);

11    /* Racunamo dan, mesec i godinu prethodnog dana */
    prethodni_dan=dan-1;
13    prethodni_mesec=mesec;
    prethodni_godina=godina;

15    /* I po potrebi vrsimo korekcije */

17    /* Ako je u pitanju prvi u mesecu */
19    if(prethodni_dan==0){
        /* Treba korigovati mesec */
21        prethodni_mesec=mesec-1;
        /* Ako je u pitanju januar */
23        if(prethodni_mesec==0){
            /* Treba korigovati i godinu */
25            prethodni_mesec=12;
            prethodni_godina=godina-1;
27        }

29        /* Analiziramo redni broj meseca kako bi odredili
           tacan dan*/
31        switch(prethodni_mesec){
            case 1:
33            case 3:
            case 5:
35            case 7:
            case 8:
37            case 10:
            case 12:
39                prethodni_dan=31;
                break;
41            case 2:
                if((prethodni_godina%4==0 && prethodni_godina%100!=0)
```



```

43         || prethodni_godina%400==0)
           prethodni_dan=29;
45     else
           prethodni_dan=28;
47     break;
           case 4:
49     case 6:
           case 9:
51     case 11:
           prethodni_dan=30;
53     }
    }
55
    /* Ispisujemo datum koji smo izracunali */
57     printf("Prethodni datum: %d.%d.%d\n",
           prethodni_dan, prethodni_mesec, prethodni_godina);
59
    return 0;
61 }

```

Rešenje 2.33 Rešenje je analogno rešenju zadatka 2.32.

Rešenje 2.35

```

#include<stdio.h>
2  #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

      printf("Unesi realne koordinate 4 tacke:");
18     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;
printf("Obim pravougaonika je %f\n",0);
}
else
    printf("Tacke ne cine pravougaonik sa stranicama ",
           "koje su paralelne koordinatnim osama\n");
break;
case 'h':
    /*
        Ukoliko se tacke A(x1,y1) i B(x2,y2) ne nalaze
        na pravoj koja je paralelna x osi, izracunamo k,n
        za pravu odredjenu tackama A(x1,y1) i B(x2,y2)
    */
    if ((x1-x2)!=0)
    {
        k = (y1-y2)/(x1-x2);
        n = y1-k*x1;
        /*
            Proverimo da li tacke C(x3,y3) i D(x4,y4) nalaze
            na toj pravoj
        */
        if (y3==x3*k+n && y4==x4*k+n)
            printf("Tacke su kolinearne, pripadaju pravoj ",
                   "y=%f*x+%f\n",k,n);
        else
            printf("Tacke nisu kolinearne\n");
    }
    /*
        Ukoliko se A i B nalaze na pravoj koja je paralelna
        x osi, proverimo da li tacke C(x3,y3) i D(x4,y4)
        nalaze na toj pravoj
    */
    else
        if (x3==x1 && x4==x1)
            printf ("Tacke su kolinearne, pripadaju pravoj ",
                    "x=%f\n",x1);
        else
            printf("Tacke nisu kolinearne\n");
        break;
case 'j':
    delta = x1*y2-x2*y1;
    deltap = x2*(y4-y3)-y2*(x4-x3);
    deltaq = x1*(y4-y3)-y1*(x4-x3);
    if (delta!=0)
        printf("Sistem ima jedinstveno resenje %.2f, %.2f\n",
               deltap/delta, deltaq/delta);
    else if (deltap==0 && deltaq==0)
        printf("Sistem je neodredjen ili nema resenja.\n");
    else
        printf("Sistem nema resenja\n");
    break;
default:

```

```

80     }
82     return 0;
}
printf("Nekorektan unos\n");

```

2.3 Petlje

2.3.1 Ispis podataka

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.
|| 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 deluje 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. Takođe, treba izbegavati formulacije koje sugerisu redosled programiranja i naredbi u programu. Dakle više idemo na formulacije šta 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 podataka (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
Aritmeticka sredina: 5.5775

```

Primer 2

```

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

```

[Rešenje 2.57]

2 Kontrola toka

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 artikala. Program učitava realan pozitivan broj m , ceo pozitivan broj n i n realnih pozitivnih brojeva različitih od 0. Ispisati koliko artikala ima manju ili jednaku cenu od m . U slučaju greške ispisati odgovarajuću poruku.

Primer 1

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

Primer 2

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

Primer 3

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

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 moze se napisati rec Zima.
```

Primer 2

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

[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 standradnog ulaza unose se realan broj x i ceo pozitivan broj n . Napisati program koji izračunava n -ti stepen broja x , tj. x^n .

Primer 1

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

Primer 2

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

Primer 3

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

[Rešenje 2.72]

Zadatak 2.73 Sa standradnog 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*

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:

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

U slučaju greške pri unosu podataka ispisati odgovarajuću poruku.

Primer 1

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

Primer 2

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

Primer 3

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

Primer 4

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

[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 iscertava 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 standarnog ulaza unose se neoznačeni celi brojevi m i n . Napisati program koji iscertava 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 standarnog 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 iscrtava 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 iscrtava 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 primerima negde ima blanko posle zareza, negde nema, i to treba ujednačiti.

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 izvorsiti i
       naziva se brojac petlje.
       Njenu pocetnu vrednost postavljamo na 0 jer se u pocetku petlja
       nije ni jednom izvrsila. */
6     int i = 0;

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

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

```

2 Kontrola toka

```
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. */
    int i=0;
    /* Promenljiva koja oznacava koliko puta cemo ispisati trazeni
    tekst. */
    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()
{
5     /* Promenljivu x koristimo u dve svrhe. Prvo, ova promenljiva
    kontrolise koliko puta se petlja izvorsila.
    Drugo, ovu promenljivu koristimo za ispis potrebnih vrednosti.
    */
7     int x;
    /* Promenljiva n se unosi i odredjuje koliko brojeva ispisujemo.
    */
9     int n;
```

```
11 printf("Unesi pozitivan ceo broj: ");
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 */
```

2 Kontrola toka

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

```
/* Resenje pod b). */
2
#include <stdio.h>
4
int main()
6 {
8     /* Promenljive koje oznacavaju granice intervala. */
9     int n,m;
10    /* Promenljiva koja oznacava trenutno ispisani broj intervala. */
11    int i;
12
13    printf("Unesi dva cela broja: ");
14    scanf("%d%d",&n,&m);
15
16    if (m < n)
17    {
18        printf("Neispravan unos. Nisu dobro zadate granice intervala!\n");
19        return -1;
20    }
21
22    /* naredba i=n se izvrsava jednom, pre prve
23       iteracije */
24    for(i=n;i<=m;i++) /* uslov petlje i<=m se proverava pre svake
25       iteracije */
26        printf("%d ", i); /* naredba i++ se izvrsava nakon svake
27       iteracije */
28
29    printf("\n");
30
31    return 0;
32 }
```

```
1 /* Resenje pod c). */
2
3 #include <stdio.h>
```

```

5 int main()
6 {
7     /* Promenljive koje oznacavaju granice intervala. */
8     int n,m;
9     /* Promenljiva koja oznacava trenutno ispisani broj intervala. */
10    int i;
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    /* Uslov petlje se proverava na kraju svake iteracije. */
23    /* Zbog toga se do while petlja izvrsava bar jednom, cak i u
24    slucaju
25        da uslov petlje nikada nije ispunjen. */
26    i=n;
27    do                                /* Petlja se zapocinje bez provere uslova. */
28    {
29        printf("%d ",i); /* Stampa se vrednost promenljive i. */
30        i++;             /* Uvecava se vrednost promenljive i. */
31    }
32    while(i<=m);          /* Proverava se uslov i ukoliko je ispunjen,
33    nastavlja se sa sledecom iteracijom. */
34    /* U suprotnom, petlja se zavrшава i program
35    se nastavlja od prve naredbe koja sledi za petljom. */
36    printf("\n");
37    return 0;
38 }

```

Rešenje 2.41

```

1 #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. */

```

2 Kontrola toka

```
/* 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;
```



```

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,*/
       {
23         /* izraz se tumaci kao tacan i izvrsavaju se naredbe iz
           if grane. */
           an1=(3*an+1)/2;
       }
25     else /* U suprotnom, ukoliko je vrednost izraza an%2 jednaka nuli
           , izraz */
       {
27         /* se tumaci kao netacan i izvrsavaju se naredbe iz else
           grane. */
           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;

```

2 Kontrola toka

```
15 printf("Uneti format papira: ");
16 scanf("%u", &format);
17
18 /* duzina/sirina = 1 : sqrt(2) duzina*sirina = 1000x1000mm^2
19  * Na osnovu ovih odnosa dobijamo pocetnu vrednost za sirinu
20  * i duzinu, odnosno vrednosti za papir A0.
21  */
22 duzina = sqrt(1000*1000/sqrt(2));
23 sirina = sqrt(2)*duzina;
24
25 /* Kako vec imamo odredjenu sirinu i duzinu za papir A0,
26  * petlju krecemo od izracunavanja za papir A1, pa
27  * brojac i postavljamo na 1.
28  */
29 for(i=1; i<=format; i++)
30 {
31     nova_duzina = sirina/2;
32     sirina = duzina;
33     duzina = nova_duzina;
34 }
35
36 /* Duzina i sirina celi brojevi. */
37 konacna_sirina = (unsigned int)sirina;
38 konacna_duzina = (unsigned int)duzina;
39
40 printf("%u %u\n", konacna_duzina, konacna_sirina);
41
42 return 0;
43 }
```

Rešenje 2.44

```
1 #include<stdio.h>
2 #include<stdlib.h>
3
4 int main()
5 {
6     int x;
7     char cifra;
8     printf("Unesi ceo broj:");
9     scanf("%d", &x);
10
11     /* Pretvaranje u apsolutnu vrednost se vrsi za slucaj kada je unet
12     * negativan broj kako bismo osigurali da ce nam izdvojene cifre
13     * biti pozitivne.
14     */
15     x = abs(x);
16
17     /* Kako uklanjamo cifre broja (pogledati telo petlje)
18     * u nekom trenutku broj 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         /* Uklanjamo 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 posledjnu cifru broja. */
15        cifra=n%10;

17        /* Proveravamo da li je bas ona jednaka broju 5 */
        if(cifra==5){
19            /* 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
            * nacin.
            */
27        n=n/10;
29    }

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

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

    return 0;
}
```

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)
```

2 Kontrola toka

```
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   * postavljja 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>
2 #include <math.h>
3
4 int main() {
5     unsigned int x;
6     /* Tezina trenutne pozicije u broju. Moze biti 1, 10, 100,
7      * 1000 itd.
8      */
9     int stepen_deset;
10    /* Trenutna izdvojena cifra iz broja x. */
11    int cifra;
12    /* Redni broj cifre koja se trenutno obradjuje, gledano s desna
13     * na levo.
14     */
15    int rbr;
16    /* Broj dobijen nakon transformacije. */
17    unsigned int y;
18
19    /* Ucitavanje broja. */
20    printf("Unesite broj: ");
21    scanf("%d", &x);
22
23    if(x <= 0)
24    {
25        printf("Nekorektan unos.\n");
26        return -1;
27    }
28    /* Postavlja se vrednost stepena na 0 - to znaci da se prvo mnozi
29     * sa
30     * 10^0=1.
31     */
32    stepen_deset = 0;
33
34    /* Postavlja se vrednost broja koji se formira na 0. */
35    y = 0;
36    /* Postavlja se redni broj pozicije na 0. */
37    rbr = 0;
38
39    /* Provera da li ima cifara u zapisu broja. */
40    while(x > 0) {
41
42        /* Izdvajanje cifre. */
43        cifra = x%10;
44
45        /* Proverava se da li je pozicija izdvojene cifre parna -
46         * cifre na parnim pozicijama se zadrzavaju.
47         */
48        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

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

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

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

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

63 return 0;

65 }
```

Rešenje 2.52

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

3
5 int main() {
7     int x;
    int broj_cifara;
    int min_stepen, max_stepen;
```

```
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;
```

```

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.
```

```

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>
   int main()
3  {
    /* Broj artikala. */
    int n;
    /* Brojac. */
    int i;
    /* Cena trenutno unetnog artikla. */
    float cena;
    /* Minimalna cena. */
    float min_cena;

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

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

21

23     /* Prva cena se unosi iznad petlje
    * kako bi bio njegova vrednost bila dodeljena promenljivoj
    min_cena.
    * 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:");

```

2 Kontrola toka

```
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)
            min_cena = cena;
53         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;
8       float suma;
9
10      broj_brojeva=0;
11      suma=0;
12
13      printf("Unesite brojeve: ");
14
15      while(1){
```



```

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
        printf("Unesite cene: ");
13
        while(1){
15
            scanf("%f", &cena);
17
            if(cena == 0)
                break;
19
            if (cena < 0)
21             {
23                 printf("Cena ne moze biti negativna.\n");
                return -1;
25             }
        }
    }

```

```
27     suma+=cena;
29
30     /* I uvecava se broj ucitanih brojeva. */
31     broj_artikla++;
32 }
33
34 if (broj_artikla == 0)
35     printf("Nisu unesene cene.\n");
36 else
37     printf("Aritmeticka sredina: %.4f\n", suma/broj_brojeva);
38
39 return 0;
40 }
```

Rešenje 2.59

```
1 #include <stdio.h>
2
3 int main()
4 {
5     /* Promenljiva cena oznacava trenutno unesenu cenu. */
6     float cena;
7     float m;
8     int n, i;
9     int broj_brojeva=0;
10
11     printf("Unesite koliko novaca ima Nikola: ");
12     scanf("%f", &m);
13
14     if (m < 0)
15     {
16         printf("Nikola ne moze imati negativno novaca.\n");
17         return -1;
18     }
19
20     printf("Unesite broj artikala: ");
21     scanf("%d", &n);
22
23     if (n < 0)
24     {
25         printf("Broj artikala ne moze biti negativan.\n");
26         return -1;
27     }
28
29     printf("Unesite cene artikala: ");
30
31     i=0;
32     while(i<n){
33         /* Ucitava se cena artikla. */
34         scanf("%f", &cena);
```

```
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       * su
8       * razlicitog znaka. */
9      double broj, sledbenik;
10     /* Brojac. */
11     int i;
12     int broj_promena = 0;
13
14     printf("Unesite broj n ");
15     scanf("%d", &n);
16
17     if (n < 0)
18     {
19         printf("Neispravan unos.\n");
20         return -1;
21     }
22
23     /* Prvo se proveara da li uopste ima unosa, i ako
24      * unosa nema, ispisuje se odgovarajuca poruka i
25      * izlazi iz programa. */
26     if (n == 0)
27     {
28         printf("Broj promena je 0.\n");
29         return 0;
30     }
31
32     printf("Unesite brojeve: ");
33     /* Pre petlje ucitava se jedan broj, a u petlji
34      * se ucitava njegov sledbenik i proverava se
35      * da li su razlicitog znaka. */
36     scanf("%lf", &broj);
37
38     /* Kako je vec jedan broj unesen, brojac se postavlja
39      * na 1, a ne na 0. */
40     for(i=1; i<n; i++)
41     {
42         /* Ucitava se sledbenik. */
43         scanf("%lf", &sledbenik);
44
45         /* Ako su razlicitog znaka proizvod je manji od 0. */
46     }
```

```

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     /* Tekuci sledbenik postaje tekuci broj, a
53        * u sledecoj iteraciji petlje se ucitava
54        * sledeci sledbenik. */
55     broj = sledbenik;
56 }

57

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

59

60 return 0;
61 }

```

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
43              * trenutne maksimalne cifre desetice. */
          if(x_desetica>max_desetica)
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;
7       int x, x_kopija, br_cifara;
       int max_br_cifara, broj;
9       int i;

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++){
26
        scanf("%d", &x);

28
        /* 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){
36
            /* Cuvamo je */
            max_vodeca_cifra=vodeca_cifra;
38
            /* I zbog ispisa, cuvamo i broj u kojem se ona pojavljuje */
            /* 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

48
    return 0;

50
}
```

Rešenje 2.65


```
1  #include <stdio.h>
   #include <stdlib.h>
3
   int main(){
5
       int n;
7       int d;
           /* Uzastopni brojevi za koje se racuna rastojanje. */
9       int x, y;
       int broj_parova;
11      int i;

13
       printf("Unesite brojeve n i d: ");
15      scanf("%d %d", &n, &d);

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

23      broj_parova=0;

25      printf("Unesite n brojeva: ");

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

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

37         /* Broj iz tekuće iteracije se čuva kako bi
               * mogao da se upotrebljava u narednoj iteraciji. */
39         x=y;
       }
41
       printf("Broj parova: %d\n", broj_parova);
43
       return 0;
45
   }
```

Rešenje 2.66

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

2 Kontrola toka

```
1  #include <math.h>
3
5  int main(){
7
9      int x;
10     int min, max;
12
13     printf("Unesite brojeve: ");
15
16     /* Prvi broj se učitava izvan petlje zbog
17      * inicijalizacije maksimuma i minimuma. */
18     scanf("%d", &x);
19     max=x;
20     min=x;
22
23     /* Sve dok se ne unese 0, učitavaju se brojevi u petlji. */
24     while(x!=0){
26
27         /* Provera da li je procitani broj veci od aktuelnog maksimuma.
28          */
29         if(x>max)
30             max=x;
31         /* Provera da li je procitani broj manji od aktuelnog minimuma.
32          */
33         if(x<min)
34             min=x;
36
37         /* Učitavanje narednog broja. */
38         scanf("%d", &x);
39     }
41
42     printf("Razlika: %d\n", max-min);
44
45     return 0;
47 }
49 }
```

Rešenje 2.67

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

```

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 return 0;
25 }

```

Rešenje 2.68

```

#include <stdio.h>

2
int main()
3
4 {
5     char c;

6

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

14

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

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

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

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

    return 0;
}
```

Rešenje 2.70

```
1  /* Sa standardnog ulaza se unosi ceo broj n, a zatim i n karaktera.
   Napisati program koji proverava da li se od
   unetih karaktera moze napisati rec Zima. */
3
#include <stdio.h>
5 #include <math.h>

7 int main(){

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

13    broj_Z=0;
15    broj_i=0;
    broj_m=0;
17    broj_a=0;

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

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

31        /* Analiziramo karakter */
        switch(c){
33            case 'Z':
                broj_Z++;
35            break;
            case 'i':
37                broj_i++;
        }
```

2 Kontrola toka

```
        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 faktorijel.
7     * Kako faktorijel je jako veliki broj, za tip podataka
        * se uzima unsigned long, da bi mogla da se upise
9     * sto veca vrednost. */
    unsigned long f;
11    int i;
    int original;

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

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

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

```
27     }
29     original=x;
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
13    printf("Unesite pozitivan ceo broj:");
14    scanf("%d", &n);
```



```
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(){
    int n, i;
    float x, S, stepen;

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

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

    /* Inicijalizacija sume. */
    S=0;

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

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

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

    return 0;
}
```

Rešenje 2.77

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

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

7     S=1;
8     stepen=1;
9     for(i=1; i<=n; i++)
10     {
11         stepen=stepen*x;
12     }
```

```

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

28     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

```

2 Kontrola toka

```
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
18     S=0;
19     clan = 1;
20     i=1;
21     znak=1;
22
23     /* Kako clanovi sume mogu biti negativni, potrebno je
       posmatrati
24     * apsolutnu vrednost clana. */
25     while(fabs(clan)>eps)
26     {
27         S = S + clan;
28
29         /* Promena znaka. */
30         znak = - znak;
31         clan = clan*x / i;
32         clan *= znak;
33
34         i++;
35     }
36
37     printf("S=%f\n", S);
38
39     return 0;
40 }
```

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: ");
```

```

13     scanf("%lf %d", &x, &n);
15     if (n <= 0)
16     {
17         printf("Neispravan unos.\n");
18         return -1;
19     }
21     for(i=0; i<n; i++)
22     {
23         stepen *= x;
24         proizvod *= 1 + cos(stepen);
25     }
27     printf("Proizvod = %lf\n", proizvod);
29     return 0;
30 }

```

Rešenje 2.81

```

1  #include <stdio.h>
3  int main()
4  {
5      int n, i;
6      double Razlomak;
7
8      printf("Unesite prirodan broj: ");
9      scanf("%d", &n);
10
11     if (n <= 0)
12     {
13         printf("Neispravan unos.\n");
14         return -1;
15     }
17     Razlomak = n;
19     /* Razlomak se izracunava "od nazad", odnosno, krece se
20      * od najnizeg razlomka 1/n i od njega se nadalje formira
21      * sledeci, "visi" razlomak itd. Završava se kada se stigne
22      * do koraka 0 + 1/R. */
23     for(i=n-1; i>= 0; i--)
24         Razlomak = i + 1/Razlomak;
26     printf("Razlomak = %lf\n", Razlomak);
28     return 0;
29 }

```

Rešenje 2.82

```
1 #include <stdio.h>
2 #include <math.h>
3 int main(){
4     /* Promenljiva i je brojac, promenljiva S cuva izracunatu
5     sumu,
6     * promenljiva znak moze biti 1 ili -1 i odredjuje znak
7     * trenutnog clana sume, a promenljiva clan je tekuci clan
8     niza. */
9     int i, znak, n;
10    float S;
11    float x, clan;
12
13    printf("Unesite x i n: ");
14    scanf("%f%d", &x, &n);
15
16    if (n <= 0)
17    {
18        printf("Neispravan unos.\n");
19        return -1;
20    }
21
22    S=0;
23    clan = 1;
24    i=1;
25    znak=1;
26
27    /* Kako clanovi sume mogu biti negativni, potrebno je
28    posmatrati
29    * apsolutnu vrednost clana. */
30    while(i <= 2*n)
31    {
32        S = S + clan;
33
34        /* Promena znaka. */
35        znak = - znak;
36        /* Svaki clan suma se od prethodnog clana razlikuje za x
37        ^2/(i*(i+1)). */
38        clan = clan*x*x / (i*(i+1));
39        clan *= znak;
40
41        i+=2;
42    }
43
44    printf("S=%f\n", S);
45
46    return 0;
47 }
```

Rešenje 2.83

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

Rešenje 2.84

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

2 Kontrola toka

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

Rešenje 2.88

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

```
11  /* Krstice koje iscrtavamo mozemo posmatrati kao dijagonale
    13  kvadrata dimenzije n */
    15  /* Prolazimo kroz sve vrste kvadrata */
    17  for(i=1; i<=n; i++){
    19      /* Prolazimo kroz sve kolone kvadrata */
    21      for(j=1; j<=n; j++){
    23          /* Ako se nalazimo na glavnoj ili sporednoj dijagonali */
    25          if(i==j || i+j==n+1)
    27              /* Stampamo zvezdu */
    29              putchar('*');
    31          else
    33              /* Inace, stampamo blanko znak */
    35              putchar(' ');
    37      }
    39      /* Nakon uspesno iscrtane vrste, stampamo znak za novi red */
    41      putchar('\n');
    43  }
    45  return 0;
    47 }
```

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](#)