

## PROGRAMIRANJE 1



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

**PROGRAMIRANJE 1**  
**Zbirka zadataka sa rešenjima**

**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 sa rešenjima

# Sadržaj

<b>1</b>	<b>Uvodni zadaci</b>	<b>1</b>
1.1	Rešenja . . . . .	5
<b>2</b>	<b>Kontrola toka</b>	<b>13</b>
2.1	Naredbe grananja . . . . .	13
2.2	Petlje . . . . .	19
2.3	Funkcije . . . . .	29
2.4	Rešenja . . . . .	37
<b>3</b>	<b>Predstavljanje podataka</b>	<b>83</b>
3.1	Nizovi . . . . .	83
3.2	Pokazivači . . . . .	92
3.3	Niske . . . . .	96
3.4	Višedimenzioni nizovi . . . . .	101
3.5	Strukture . . . . .	101
3.6	Rešenja . . . . .	105
<b>4</b>	<b>Ulaz i izlaz programa</b>	<b>175</b>
4.1	Standardni tokovi . . . . .	175
4.2	Argumenti komandne linije . . . . .	175
4.3	Datoteke . . . . .	175
4.4	Rešenja . . . . .	179
<b>5</b>	<b>Razni zadaci</b>	<b>197</b>
5.1	Rešenja . . . . .	197
<b>A</b>	<b>Ispitni zadaci</b>	<b>199</b>
A.1	Kvalifikacioni zadaci . . . . .	199
A.2	Ispitni rokovi . . . . .	199
A.2.1	Praktični deo ispita, jun ... . . . .	199

A.3 Rešenja . . . . .	199
-----------------------	-----

# Predgovor

U okviru kursa *Programiranje 1* na Matematičkom fakultetu vežbaju se zadaci koji imaju za cilj da studente nauče ...

*Autori*





# 1

## Uvodni zadaci

**Zadatak 1.1**   Tekst

[Rešenje [1.1](#)]

**Zadatak 1.2**   Tekst

[Rešenje [1.2](#)]

**Zadatak 1.3**   Tekst

[Rešenje [1.3](#)]

**Zadatak 1.4**   Tekst

[Rešenje [1.4](#)]

**Zadatak 1.5**   Tekst

[Rešenje [1.5](#)]

**Zadatak 1.6**   Tekst

[Rešenje [1.6](#)]

**Zadatak 1.7**   Tekst

[Rešenje [1.7](#)]

**Zadatak 1.8**   Tekst

[Rešenje [1.8](#)]

## 1 Uvodni zadaci

---

### Zadatak 1.9 Tekst

[Rešenje 1.9]

### Zadatak 1.10 Tekst

[Rešenje 1.10]

**Zadatak 1.11** Napisati program koji omogućava korisniku da unese ceo broj, a zatim ispisuje njegov kvadrat i kub.

#### Primer 1

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

[Rešenje 1.11]

**Zadatak 1.12** Napisati program koji za unete stranice pravougaonika ispisuje njegov obim i površinu.

#### Primer 1

```
INTERAKCIJA SA PROGRAMOM:  
Unesite duzine stranica pravougaonika: 2 8  
Obim: 20  
Povrsina: 16
```

[Rešenje 1.12]

**Zadatak 1.13** Napisati program koji za unete stranice trougla ispisuje njegov obim i površinu.

#### Primer 1

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

[Rešenje 1.13]

**Zadatak 1.14** Napisati program koji za unete dimenzije sobe u metrima (dužinu, širinu i visinu) ispisuje koju površinu treba da okreći moler. Uračunati da na vrata i prozore otpada oko 20%. Omogućiti i unos cene usluge po kvadratnom

metru i izračunati zaradu koju ostvaruje moler.

*Primer 1*

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesite dimenzije sobe: 4 4 3  
|| Unesite cenu po kvadratnom metru: 500  
|| Moler treba da okreci 51.2 kvadratna metra  
|| Cena krecenja je 25600
```

[Rešenje 1.14]

**Zadatak 1.15** Napisati program koji za unetu količinu jabuka u kilogramima i unetu cenu po kilogramu ispisuje ukupan iznos koji treba platiti.

*Primer 1*

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

[Rešenje 1.15]

**Zadatak 1.16** Napisati program koji pomaže kasirki da obračuna kusur tako što od nje traži da unese cenu artikla, količinu artikla i iznos koji je dobila od kupca.

*Primer 1*

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

[Rešenje 1.16]

**Zadatak 1.17** Napisati program koji prirodnom četvorocifrenom broju koji se unosi sa standardnog ulaza:

- izračunava proizvod cifara
- izračunava razliku sume krajnjih i srednjih cifara
- izračunava sumu kvadrata cifara
- određuje broj koji se dobija ispisom cifara u obrnutom poretku
- određuje broj koji se dobija zamenom cifre jedinice i cifre stotine

## 1 Uvodni zadaci

---

### 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
```

[Rešenje 1.17]

**Zadatak 1.18** Napisati program koji izbacuje cifru desetica datom prirodnom broju.

### Primer 1

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

[Rešenje 1.18]

**Zadatak 1.19** Napisati program koji u datom prirodnom broju  $x$  ubacuje cifru  $c$  na poziciju  $p$  i rezultat ispisuje na standardni izlaz. Brojevi  $x$ ,  $c$  i  $p$  se unose sa standardnog ulaza. Podrazumeva se da je broj  $p$  manji od ukupnog broja cifara broja  $i$  da numeracija cifara počinje od 1. Uputstvo: koristiti funkciju *pow* iz *math.h* biblioteke.

### Primer 1

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesite redom x, c i p: 140 2 2  
|| Rezultat je: 1420
```

[Rešenje 1.19]

**Zadatak 1.20** Napisati program koji:

- unetu dužinu u miljama konvertuje u kilometre ( $1 \text{ mi} = 1.609344 \text{ km}$ )
- unetu težinu u funtama konvertuje u kilograme ( $1 \text{ lb} = 0.45359237 \text{ kg}$ )
- unetu temperaturu u celzijusima konvertuje u farenhajte ( $F = \frac{9 \cdot C}{5} + 32$ )

*Primer 1*

```
INTERAKCIJA SA PROGRAMOM:
Unesite duzinu u miljama: 1.8
Vrednost duzine u kilometrima je: 2.896819
Unesite tezinu u funtama: 10
Vrednost tezine u kilogramima je: 4.535923
Unesite temperaturu u celzijusima: 37.2
Vrednost temperature u farenhajtima je: 98.960007
```

[Rešenje 1.20]

**Zadatak 1.21** Napisati program koji učitava sa standardnog ulaza vreme poletanja i vreme sletanja aviona, a potom ispisuje dužinu trajanja leta. Možemo pretpostaviti da su poletanje i sletanje u istom danu.

*Primer 1*

```
INTERAKCIJA SA PROGRAMOM:
Unesite vreme poletanja: 8 5 0
Unesite vreme sletanja: 12 41 30
Duzina trajanja leta: 4 h 36 min 30 sec
```

[Rešenje 1.21]

## 1.1 Rešenja

### Rešenje 1.1

```
1  /*
   Napisati program koji na standardni izlaz ispisuje tekst "Zdravo
     svete!"
3  */
   #include<stdio.h>
5
   int main()
7  {
   /* printf - funkcija pomocu koje se vrsi ispis */
9  /* oznaka \n : prelazak u novi red */
   printf("Zdravo svete!\n");
11
   /* naredne dve naredbe ispisace reci Zdravo i svete u istom redu*/
13  printf("Zdravo ");
   printf("svete \n");
15
   /* naredne dve naredbe ispisace reci Zdravo i svete u posebnim
     redovima */
```

## 1 Uvodni zadaci

---

```
17  /* jer se u prvoj printf naredbi na kraju oznakom \n prelazi u novi
    red */
19  printf("Zdravo \n");
    printf("svete \n");
21
23  return 0;
}
```

### Rešenje 1.2

```
1  /*
    Napisati program koji poziva korisnika da unese dva cela broja sa
    standardnog ulaza,
3  a zatim ispisuje:
    1) unete vrednosti
5  2) njihov zbir
    3) njihovu razliku
7  4) njihov proizvod
    5) ceo deo pri deljenju jednog broja drugim brojem
9  6) ostatak pri deljenju jednog broja drugim brojem
11 */
13 #include<stdio.h>
15 int main()
    {
17     int x;
        int y;
19     int rezultat;

21     printf("Unesi vrednost celobrojne promenljive x:");
        scanf("%d", &x); /* "%d" - specifikator tipa koji treba uneti (%d
            za int)
23                                &x    - adresa promenljive x
                                */
25
27     printf("Unesi vrednost celobrojne promenljive y:");
        scanf("%d", &y);

29     /* 1) ispis unetih vrednosti */
        printf("x=%d, y=%d\n", x,y); /* umesto prvog %d bice ispisana
            vrednost promenljive x */
31                                /* umesto drugog %d bice ispisana
            vrednost promenljive y */

33     /* 2) ispis zbira */
        rezultat = x+y; /* dodelimo vrednost promenljivoj rezultat */
35     printf("Zbir je %d\n", rezultat);
```

```

37  /* 3) ispis razlike */
printf("Razlika je %d\n",x-y); /* mozemo ispisivati direktno
    vrednost izraza x-y i bez */
39                                /* njegovog dodeljivanja posebno
    promenljivoj */

41
/* 4) ispis proizvoda */
43 printf("%d*d=%d\n",x,y,x*y);

45 /* 5) ispis kolicnika */
rezultat = x/y;
47 printf("celobrojno deljenje: %d/%d=%d\n",x,y,rezultat); /*
    promenljiva rezultat je celobrojna (int) */
                                                /* ona ne
    moze sadrzati realan broj */
49                                                /* ukoliko
    je x=7, a y=2, tada ce nakon naredbe */
                                                /*
    rezultat=x/y; promenljiva rezultat imati vrednost 2 */
51                                                /* a ne
    2.5 */

53 printf("ostatak pri celobrojnomo deljenju: %d %% %d=%d\n",x,y,x%y);
                                                /*
    operator % izracunava ostatak pri celobrojnomo deljenju */
55                                                /* 7%2 ima
    vrednost 1 (jer je 7=3*2+1) */
                                                /* oznaku
    % u naredbi printf pisemo %% */
57 return 0;
}

```

### Rešenje 1.3

```

/*
2  Napisati program koji sa standardnog ulaza ucitava realnu vrednost
    izrazenu
    u incima, konvertuje tu vrednost u centimetre i ispisuje je na
    standardni izlaz
4  zaokruzenu na dve decimale.
*/
6  #include <stdio.h>

8  int main()
{
10     float in;
    float cm;
12
    printf("Unesi broj inca: ");

```

## 1 Uvodni zadaci

---

```
14  scanf("%f", &in);                                /* "%f" specifikator za unos
    /ispis float promenljivih */
16  cm = in*2.54; /* 1 inch = 2.54 cm */
18  printf("%.2f in = %.2f cm\n", in, cm); /* "%.4f" - ispis realne
    promenljive na 4 decimale */
20  return 0;
}
```

### Rešenje 1.4

```
1  /*
    Napisati program koji sa standardnog ulaza učitava duzinu
    poluprecnika kruga
3   i na standardni izlaz ispisuje njegov obim i površinu
    */
5
7  #include <stdio.h>
    #include <math.h> /* biblioteka matematičkih funkcija; za prevodjenje
        je neophodno uključiti opciju -lm
        npr. gcc primer.c -lm */
9  int main()
    {
11     int r;
        float O;
13     float P;
        printf("Unesi poluprecnik kruga:");
15     scanf("%d", &r);

17     O=2*r*M_PI; /* M_PI - konstanta pi koja se nalazi u biblioteci math
        .h */
        P=r*r*M_PI;
19
        printf("Obim: %f, površina: %f\n",O,P);
21
        return 0;
23 }
```

### Rešenje 1.5

```
1  /*
    Napisati program koji učitava trocifreni broj koji se
3   unosi sa standardnog ulaza i ispisuje njegove cifre na
    standardni izlaz.
5   */
7  #include <stdio.h>
    int main()
```



```

{
9   int x;
   int cifra_jedinice;
11  int cifra_desetice;
   int cifra_stotine;

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

17
   cifra_jedinice = x%10;
   cifra_desetice = (x/10)%10;
19  cifra_stotine = x/100;

21
   printf("Cifre unetog broja su %d,%d,%d\n", cifra_jedinice,
        cifra_desetice, cifra_stotine);

23
   /*
    2. nacin, bez uvođenja dodatnih promenljivih cifra_jedinice,
    cifra_desetice i cifra_stotine:

25
   printf("Cifre unetog broja su %d,%d,%d\n", x%10, (x/10)%10, x/100)
   ;
27  */
   return 0;
29 }

```

## Rešenje 1.6

```

1  /*
   Napisati program koji ucitava trocifreni broj koji se
3  unosi sa standardnog ulaza i ispisuje broj dobijen obrtanjem
   njegovih cifara.
5  */
   #include <stdio.h>
7  int main()
   {
9     int x;
     int obrnuto_x;

11
     int cifra_jedinice;
13     int cifra_desetice;
     int cifra_stotine;

15
     printf("Unesi trocifreni broj:");
17     scanf("%d", &x);

19
     cifra_jedinice = x%10;
     cifra_desetice = (x/10)%10;
21     cifra_stotine = x/100;

```

## 1 Uvodni zadaci

---

```
23     obrnuto_x = cifra_jedinice*100 + cifra_desetice*10 + cifra_stotine
        ;
25     printf("Obrnuto x: %d\n", obrnuto_x);
27     return 0;
}
```

### Rešenje 1.7

```
/*
2   Napisati program koji za unetu duzinu stranice jednakostranicnog
   trougla
   ispisuje njegovu povrstinu.
4 */
6 #include <stdio.h>
   #include <math.h>
8 int main()
   {
10     unsigned int a;
       float P;
12
       printf("Unesi duzinu stranice jednakostranicnog trougla:");
14     scanf("%d",&a);
16
       P = (a*a*sqrt(3))/4;
18
       printf("Povrsina jednakostranicnog trougla stranice %d je %f\n",a
       ,P);
       return 0;
20 }
```

### Rešenje 1.8

```
/*
2   Napisati program koji za unetu cenu proizvoda ispisuje najmanji
   broj
   novcanica koje je potrebno izdvojiti da bi se proizvod platio. Na
   raspolaganju su novcanice od 1000,100,50,10 i 1 dinar. Na primer,
4   za unetu cenu 5178, program na standardni izlaz treba da ispise:
   5178=5*1000+ 1*100 +1*50 +2*10 +8*1
6 */
8
10 #include <stdio.h>
12 int main()
   {
       int x;
```

```
14     printf("Unesi cenu:");
15     scanf("%d", &x);
16
17     printf("%d=%d*1000+ ", x,x/1000);
18     x=x%1000;
19     printf("%d*100 +", x/100);
20     x=x%100;
21     printf("%d*50 +",x/50);
22     x=x%50;
23     printf("%d*10 +", x/10);
24     x=x%10;
25     printf("%d*1\n", x);
26     return 0;
27 }
```

### Rešenje 1.9

```
1  /*
2  Napisati program koji za tri cela broja koja se unose sa standardnog
3  ulaza
4  ispisuje njihovu aritmeticku sredinu na standardni izlaz.
5  */
6
7  #include<stdio.h>
8
9  int main()
10 {
11     int a, b, c;
12     float as;
13
14     printf("Unesi tri cela broja:");
15     scanf("%d%d%d",&a,&b,&c);
16
17     as=(a+b+c)/3.0; /* da bismo dobili kolicnik, jedan argument mora da
18                     bude realan broj */
19
20     /*
21     moguće je i:
22     as=1.0*(a+b+c)/3;
23     ili
24     as=((float)(a+b+c))/3;
25     */
26
27     printf("Aritmeticka sredina unetih brojeva je %f\n", as);
28     return 0;
29 }
```

### Rešenje 1.10

```
1  /*
   Napisati program koji poziva korisnika da unese dve celobrojne
   vrednosti,
3  smesta ih u promenljive x i y, zamenjuje vrednosti tih
   promenljivih i
   stampa ih na standardni izlaz.
5  */
   #include<stdio.h>
7  int main()
   {
9     int x,y;
     int t;
11    printf("Unesi dve celobrojne vrednosti:");
     scanf("%d%d",&x,&y);
13    printf("x=%d, y=%d\n",x,y);
     t=x; /* promenljiva t dobija vrednost promenljive x */
15    x=y; /* promenljiva x dobija vrednost promenljive y */
     y=t; /* promenljiva y dobija vrednost promenljive t */
17    printf("nakon zamene, x=%d, y=%d\n",x,y);
     return 0;
19 }
```

Rešenje [1.11](#)

Rešenje [1.12](#)

Rešenje [1.13](#)

Rešenje [1.14](#)

Rešenje [1.15](#)

Rešenje [1.16](#)

Rešenje [1.17](#)

Rešenje [1.18](#)

Rešenje [1.19](#)

Rešenje [1.20](#)

Rešenje [1.21](#)

## 2

# Kontrola toka

## 2.1 Naredbe grananja

**Zadatak 2.1** Tekst

[Rešenje [2.1](#)]

**Zadatak 2.2** Tekst

[Rešenje [2.2](#)]

**Zadatak 2.3** Tekst

[Rešenje [2.3](#)]

**Zadatak 2.4** Tekst

[Rešenje [2.4](#)]

**Zadatak 2.5** Tekst

[Rešenje [2.5](#)]

**Zadatak 2.6** Tekst

[Rešenje [2.6](#)]

**Zadatak 2.7** Tekst

[Rešenje [2.7](#)]

**Zadatak 2.8** Tekst

[Rešenje [2.8](#)]

**Zadatak 2.9** Tekst

[Rešenje [2.9](#)]

**Zadatak 2.10** Tekst

[Rešenje [2.10](#)]

**Zadatak 2.11** Tekst

[Rešenje [2.11](#)]

**Zadatak 2.12** Tekst

[Rešenje [2.12](#)]

**Zadatak 2.13** Tekst

[Rešenje [2.13](#)]

**Zadatak 2.14** Tekst

[Rešenje [2.14](#)]

**Zadatak 2.15** Sa standardnog ulaza se unosi ceo četvorocifren broj. Napisati program koji ispisuje njegovu najveću cifru na standardni izlaz.

*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!
```

[Rešenje [2.31](#)]

**Zadatak 2.16** Napisati program koji za dati trocifren broj proverava da li je Armstrongov. Broj je Armstrongov ako je jednak zbiru kubova svojih cifara.

*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!
```

[Rešenje 2.31]

**Zadatak 2.17** Za ceo broj  $k$  između 1 i 189 koji se unosi sa standardnog ulaza, odrediti cifru koja se nalazi na  $k$ -toj poziciji niza 12345678910111213....9899 u kom su redom ispisani brojevi od 1 do 99.

### 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 2.31]

**Zadatak 2.18** Sa standardnog ulaza se unosi četvorocifreni pozitivan broj. Napisati program koji računa i ispisuje proizvod parnih cifara datog broja. Ukoliko uneti broj nije pozitivna četvorocifrena vrednost ispisati poruku *Greska!*.

### Primer 1

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

### Primer 2

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesite broj: 3579  
|| Proizvod parnih cifara: 0
```

### Primer 3

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesite broj: 288  
|| Greska!
```

[Rešenje 2.31]

**Zadatak 2.19** Sa standardnog ulaza unosi se 5 karaktera. Proveriti da li je prvi karakter veliko ili malo slovo  $a$ . Ako jeste, ispisati karaktere obrnutim redosledom, a ako nije, ništa ne ispisivati.

### Primer 1

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesite karaktere: A u E f h  
|| h f E u A
```

### Primer 2

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesite karaktere: k L M 9 o
```

[Rešenje 2.31]

**Zadatak 2.20** Sa standardnog ulaza unosi se jedan karakter. Ako je u pitanju malo slovo, zameniti ga odgovarajućim velikim slovom i ispisati na standardni izlaz. Ako je u pitanju veliko slovo, zameniti ga odgovarajućim malim slovom i ispisati ga na standardni izlaz. Ako je u pitanju cifra ispisati poruku *cifra*. Ako je u pitanju bilo koji drugi karakter, onda ga ispisati na standardni izlaz između dveju zvezdica.

*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: >  
|| *>*
```

[Rešenje 2.31]

**Zadatak 2.21** Sa standardnog ulaza se unosi 5 karaktera. Ispisati na izlazu broj unetih 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 2.31]

**Zadatak 2.22** Sa standardnog ulaza se unosi četvorocifren ceo broj. Napisati program koji datom broju razmenjuje najmanju i najveću cifru. Dobijeni broj ispisati na standardni izlaz. Ako uneti broj nije četvorocifren ispisati poruku *Greska!*.

*Primer 1*

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

*Primer 2*

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesite broj: 247  
|| Greska!
```

[Rešenje 2.31]



**Zadatak 2.23** Sa standardnog ulaza se unose tri neoznačena trocifrena broja. Spojiti dva najveća u šestocifren broj. Spajanje izvršiti tako da najveći od trocifrenih brojeva bude na početku šestocifrenog broja. Dobijeni šestocifreni broj ispisati na izlazu. Ako neki od unetih brojeva nije trocifren, ispisati poruku *Greska!*.

*Primer 1*

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesite brojeve: 185 247 311  
|| Trazeni broj je: 311247
```

*Primer 2*

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesite brojeve: 865 11 298  
|| Greska!
```

[Rešenje 2.31]

**Zadatak 2.24** Sa standardnog ulaza se učitavaju realni koeficijenti  $A$  i  $B$  linearne jednačine  $Ax + B = 0$ . Napisati program koji ispisuje rešenja ove jednačine - ukoliko jednačina nema rešenja ili ukoliko ima više od jednog rešenja ispisati odgovarajuće poruke.

*Primer 1*

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesite koeficijente A i B: 2 -5  
|| x=2.5
```

*Primer 2*

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesite koeficijente A i B: 0 18.5  
|| Jednacina nema resenja.
```

[Rešenje 2.31]

**Zadatak 2.25** Napisati program koji za dva data intervala realne prave ( $a_1$ ,  $b_1$ ) i ( $a_2$ ,  $b_2$ ) određuje:

- 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]
```

[Rešenje 2.31]

**Zadatak 2.26** Data je funkcija  $f(x) = 2 \cdot \cos(x) - x^3$ . Sa standardnog ulaza se unosi realan broj  $x$  i broj  $k$  koje može biti 1, 2 ili 3. Napisati program koji izračunava  $F(k, x) = f(f(f(\dots f(x))))$  gde je funkcija  $f$  primenjena  $k$ -puta.

*Primer 1*

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesite redom x i k: 2.31 2  
|| F(2.31, 2)=2557.516602
```

*Primer 2*

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesite redom x i k: 12 1  
|| F(12, 1)=-1726.312256
```

[Rešenje 2.31]

**Zadatak 2.27** Napisati program koji za uneti broj  $n$  ( $1 \leq n \leq 7$ ) koji predstavlja redni broj dana u nedelji ispisuje ime 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!
```

[Rešenje 2.31]

**Zadatak 2.28** Sa standardnog ulaza se učitavaju dva cela broja i jedan od karaktera +, -, \*, / ili % koji predstavlja operaciju koju treba izvršiti nad unetim brojevima. Napisati program koji korišćenjem *switch* naredbe analizira o kom karakteru je reč i na standardni izlaz ispisuje rezultat. 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!
```

[Rešenje 2.31]

**Zadatak 2.29** 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 2.30** 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 2.31]

**Zadatak 2.31** 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.
```

[Rešenje 2.31]

## 2.2 Petlje

**Zadatak 2.32** Tekst

[Rešenje 2.32]

**Zadatak 2.33** Tekst

[Rešenje 2.33]

**Zadatak 2.34** Tekst

[Rešenje [2.34](#)]

**Zadatak 2.35** Tekst

[Rešenje [2.35](#)]

**Zadatak 2.36** Tekst

[Rešenje [2.36](#)]

**Zadatak 2.37** Tekst

[Rešenje [2.37](#)]

**Zadatak 2.38** Tekst

[Rešenje [2.38](#)]

**Zadatak 2.39** Tekst

[Rešenje [2.39](#)]

**Zadatak 2.40** Tekst

[Rešenje [2.40](#)]

**Zadatak 2.41** Tekst

[Rešenje [2.41](#)]

**Zadatak 2.42** Tekst

[Rešenje [2.42](#)]

**Zadatak 2.43** Tekst

[Rešenje [2.43](#)]

**Zadatak 2.44** Tekst

[Rešenje [2.44](#)]

**Zadatak 2.45** Tekst

[Rešenje [2.45](#)]

**Zadatak 2.46** Sa standardnog ulaza unosi se 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: -1 1 0 3
|| -1
```

*Primer 3*

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

[Rešenje 2.71]

**Zadatak 2.47** Sa standardnog ulaza unosi se realan broj  $m$ , ceo pozitivan broj  $n$  i  $n$  realnih brojeva. Izračunati i ispisati koliko je brojeva među unetima manje od zadatog broja  $m$ .

*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
```

[Rešenje 2.71]

**Zadatak 2.48** 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. Prilikom implementacije koristiti *switch* naredbu. Ne praviti razliku između malih i velikih slova.

*Primer 1*

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite broj n: 5
|| Unesite n karaktera: u A b a o
|| Samoglasnik a: 2
|| Samoglasnik e: 0
|| Samoglasnik i: 0
|| Samoglasnik o: 1
|| Samoglasnik u: 0
```

*Primer 2*

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite broj n: 7
|| Unesite n karaktera: j k + E E a e
|| Samoglasnik a: 1
|| Samoglasnik e: 3
|| Samoglasnik i: 0
|| Samoglasnik o: 0
|| Samoglasnik u: 0
```

[Rešenje 2.71]

## 2 Kontrola toka

---

**Zadatak 2.49** Sa standardnog ulaza unosi se ceo neoznačen broj. Napisati program koji proverava i ispisuje da li se cifra 5 nalazi u njegovom zapisu ili ne.

### 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 2.71]

**Zadatak 2.50** Napisati program koji unetom broju uklanja nule sa 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 2.71]

**Zadatak 2.51** Napisati program koji uneti neoznačeni ceo broj transformiše tako što svaku parnu cifru u zapisu broja uveća za 1.

### 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 2.71]

**Zadatak 2.52** Sa standardnog ulaza unosi se neoznačen ceo broj. Napisati program koji formira i ispisuje broj koji se dobija izbacivanjem svake druge cifre polaznog broja. Cifre se posmatraju sa desna na levo.

### 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 2.71]

**Zadatak 2.53** Sa standardnog ulaza unose se realan broj  $x$  i ceo neoznačen broj  $n$ . Napisati program koji izračunava  $x^n$ .

### 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 2.71]

**Zadatak 2.54** Sa standardnog ulaza unose se realan broj  $x$  i ceo broj  $n$ . Napisati program koji izračunava  $x^n$ .

### Primer 1

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

### Primer 2

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

[Rešenje 2.71]

**Zadatak 2.55** Sa standardnog ulaza unose se realan broj  $x$  i ceo neoznačen broj  $n$ . Napisati program koji izračunava sumu  $S = x + 2 \cdot x^2 + 3 \cdot x^3 + \dots + n \cdot x^n$ .

## 2 Kontrola toka

---

### 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 2.71]

**Zadatak 2.56** Sa standardnog ulaza unose se realan broj  $x$  i ceo neoznačen broj  $n$ . Napisati program koji izračunava 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 2.71]

**Zadatak 2.57** Napisati program koji sa zadatom tačnošću izračunava sumu  $S = 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \dots$ .

### 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 2.71]

**Zadatak 2.58** Napisati program koji sa zadatom tačnošću izračunava sumu  $S = 1 - x + \frac{x^2}{2!} - \frac{x^3}{3!} + \frac{x^4}{4!} - \dots$ .

### Primer 1

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesite x: 3  
|| Unesite tacnost eps: 0.001  
|| S=0.049997
```

### Primer 2

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

[Rešenje 2.71]

**Zadatak 2.59** Sa standardnog ulaza unosi se neoznačen ceo broj. Napisati program koji formira i ispisuje broj koji se dobija izbacivanjem cifara koje su jednake zbiru svojih suseda. Cifre se posmatraju sa desna na levo.



*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 2.71]

**Zadatak 2.60** Napisati program koji proverava da li je dati prirodan broj palindrom. Broj je palindrom ako se isto čita i sa leve i sa desne strane.

*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 2.71]

**Zadatak 2.61** 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 n brojeva: 18 365 25 1 78
|| 78
```

[Rešenje 2.71]

**Zadatak 2.62** 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
|| 365
```

### Primer 2

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite broj n: 7
|| Unesite n brojeva: 3 892 18 21 639 742 85
|| 892
```

[Rešenje 2.71]

**Zadatak 2.63** 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 prva cifra iz zapisa broja. Ukoliko ima više takvih, ispisati prvi.

### Primer 1

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite broj n: 5
|| Unesite n brojeva: 8 964 32 511 27
|| 964
```

### Primer 1

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite broj n: 3
|| Unesite n brojeva: 41 669 8
|| 8
```

[Rešenje 2.71]

**Zadatak 2.64** 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 2.71]

**Zadatak 2.65** 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 2.71]

**Zadatak 2.66** Sa standardnog ulaza se unose celi brojevi sve do unosa broja 0. Napisati program koji izračunava i ispisuje razliku najvećeg i najmanjeg unetog broja.

*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 2.71]

**Zadatak 2.67** Sa standardnog ulaza se unose realni brojevi sve do unosa broja 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

```

[Rešenje 2.71]

**Zadatak 2.68** Napisati program koji za uneti ceo broj  $n$  iscertava rub kvadrata dimenzije  $n$ .

### Primer 1

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

### Primer 2

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

[Rešenje 2.71]

**Zadatak 2.69** Napisati program koji za uneti ceo broj  $n$  i karakter  $c$  iscrtava rub jednakokrako pravouglog trougla čije su katete dužine  $n$ .

### 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 2.71]

**Zadatak 2.70** Napisati program koji za uneti ceo broj  $n$  iscrtava *krstiće* dimenzije  $n$ .

### Primer 1

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

### Primer 2

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

[Rešenje 2.71]

**Zadatak 2.71** 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 2.71]

## 2.3 Funkcije

**Zadatak 2.72** Tekst

[Rešenje 2.72]

**Zadatak 2.73** Tekst

[Rešenje 2.73]

**Zadatak 2.74** Tekst

[Rešenje 2.74]

**Zadatak 2.75** Tekst

[Rešenje 2.75]

**Zadatak 2.76** Tekst

[Rešenje 2.76]

**Zadatak 2.77** Tekst

[Rešenje 2.77]

**Zadatak 2.78** Tekst

[Rešenje 2.78]

**Zadatak 2.79** Tekst

[Rešenje 2.79]

**Zadatak 2.80** Tekst

[Rešenje 2.80]

**Zadatak 2.81** Tekst

[Rešenje 2.81]

**Zadatak 2.82** Tekst

[Rešenje 2.82]

**Zadatak 2.83** Tekst

[Rešenje 2.83]

**Zadatak 2.84** Tekst

[Rešenje 2.84]

**Zadatak 2.85** Tekst

[Rešenje 2.85]

**Zadatak 2.86** Napisati funkciju *int min(int x, int y, int z)* koja izračunava minimum tri broja. Napisati program koji sa standardnog ulaza učitava tri cela broja i ispisuje rezultat poziva funkcije.

*Primer 1*

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

*Primer 2*

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

[Rešenje 2.104]

**Zadatak 2.87** Napisati funkciju *unsigned int apsolutna\_vrednost(int x)* koja izračunava apsolutnu vrednost broja *x*. Napisati program koji sa standardnog ulaza učitava jedan ceo broj i ispisuje rezultat poziva funkcije.

*Primer 1*

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

*Primer 2*

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

[Rešenje 2.104]

**Zadatak 2.88** Napisati funkciju *float razlomljeni\_deo(float x)* koja izračunava razlomljeni deo broja  $x$ . Napisati program koji sa standardnog ulaza učitava jedan realan broj i ispisuje rezultat poziva funkcije.

*Primer 1*

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

*Primer 2*

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

[Rešenje 2.104]

**Zadatak 2.89** Napisati funkciju *void romb(int n)* koja iscrtava romb čija je stranica dužine  $n$ . Napisati program koji učitava ceo pozitivan broj i ispisuje rezultat poziva funkcije. U slučaju pogrešnog unosa, ispisati poruku o grešci.

*Primer 1*

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

*Primer 2*

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

*Primer 3*

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

[Rešenje 2.104]

**Zadatak 2.90** Napisati funkciju *void grafikon\_h(int a, int b, int c, int d)* koja vrši horizontalno prikazivanje zadatah vrednosti. Napisati program koji učitava četiri pozitivna cela broja i prikazuje rezultat poziva funkcije. U slučaju pogrešnog unosa, ispisati poruku o grešci.

*Primer 1*

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

*Primer 2*

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

### Primer 3

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

[Rešenje 2.104]

**Zadatak 2.91** Napisati funkciju *void grafikon\_v(int a, int b, int c, int d)* koja vrši vertikalno prikazivanje zadatih vrednosti. Napisati program koji učitava četiri pozitivna cela broja i ispisuje rezultat poziva funkcije. U slučaju pogrešnog unosa, ispisati poruku o grešci.

### Primer 1

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

### Primer 2

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

### Primer 3

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

[Rešenje 2.104]

**Zadatak 2.92** Napisati funkciju *int prestupna(int godina)* koja za zadatu godinu proverava da li je prestupna. Funkcija treba da vrati 1 ako je godina prestupna ili 0 ako nije. Napisati program koji učitava dva cela broja  $g1$  i  $g2$  i ispisuje sve godine iz intervala  $[g1, g2]$  koje su prestupne.

### Primer 1

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

### Primer 2

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



*Primer 3*

```

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

```

*Primer 4*

```

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

```

[Rešenje 2.104]

**Zadatak 2.93** Napisati funkciju *int zbir\_delilaca(int n)* koja izračunava zbir delilaca broja *n*. Napisati program koji sa standardnog ulaza učitava ceo broj *k* i ispisuje zbir delilaca svakog broja od 1 do *k*.

*Primer 1*

```

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

```

*Primer 2*

```

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

```

[Rešenje 2.104]

**Zadatak 2.94** Napisati funkciju *int ukloni\_stotine(int n)* koja modifikuje zadati broj tako što iz njegovog zapisa uklanja cifru stotina (ako postoji). Napisati program koji za brojeve koji se unose sa standardnog ulaza sve do pojave broja 0 ispisuje rezultat primene funkcije.

*Primer 1*

```

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

```

*Primer 2*

```

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

```

[Rešenje 2.104]

**Zadatak 2.95** Napisati funkciju *int rotacija(int n)* koja rotira cifre zadatog broja za jednu poziciju u levo. Napisati program koji za brojeve koji se unose sa

## 2 Kontrola toka

---

standardnog ulaza sve do pojave broja 0 ispisuje rezultat primene funkcije.

### Primer 1

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

### Primer 2

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

[Rešenje 2.104]

**Zadatak 2.96** Napisati funkciju *float aritmeticka\_sredina(int n)* koja računa aritmetičku sredinu cifara datog broja. Napisati i program koji testira rad napisane funkcije. Rezultat ispisivati na tri decimale.

### Primer 1

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

### Primer 2

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

### Primer 3

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

[Rešenje 2.104]

**Zadatak 2.97** Napisati funkciju *int zapis(int x, int y)* koja proverava da li se brojevi *x* i *y* zapisuju pomoću istih cifara. Funkcija treba da vrati vrednost 1 ako je uslov ispunjen, odnosno 0 ako nije. Napisati i program koji učitava dva cela broja i ispisuje rezultat primene funkcije.

### Primer 1

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

### Primer 2

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

### Primer 3

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

[Rešenje 2.104]

**Zadatak 2.98** Napisati funkciju *int faktorijel(int n)* koja računa faktorijel broja  $n$ . Napisati i program koji učitava dva cela broja  $x$  i  $y$  ( $0 \leq x, y \leq 12$ ) i ispisuje vrednost zbira  $x! + y!$ .

*Primer 1*

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

*Primer 2*

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

*Primer 3*

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

[Rešenje 2.104]

**Zadatak 2.99** Napisati funkciju *int rastuce(int n)* koja ispituje da li su cifre datog celog broja u rastućem poretku. Funkcija treba da vrati vrednost 1 ako cifre ispunjavaju uslov, odnosno 0 ako ne ispunjavaju uslov. Napisati i program koji učitava ceo broj i ispisuje rezultat primene funkcije.

*Primer 1*

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

*Primer 2*

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

*Primer 3*

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

[Rešenje 2.104]

**Zadatak 2.100** Broj je Armstrongov ako je jednak sumi nekog stepena svojih cifara.

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

## 2 Kontrola toka

---

- c) Napisati program koji za ceo broj koji se unosi sa standardnog ulaza proverava da li je Armstrongov (koristeci funkciju *armstrong*).

### Primer 1

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

### Primer 2

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

### Primer 3

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

[Rešenje 2.104]

**Zadatak 2.101** Napisati funkciju *int par\_nepar(int n)* koja ispituje da li su cifre datog celog broja naizmenično parne i neparne. Funkcija treba da vrati vrednost 1 ako cifre ispunjavaju uslov, odnosno 0 ako ne ispunjavaju uslov. Napisati i program koji učitava ceo broj i testira rad funkcije.

### Primer 1

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

### Primer 2

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

### Primer 3

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

[Rešenje 2.104]

**Zadatak 2.102** Napisati funkciju *int prebrojavanje(float x)* koja prebrojava koliko puta se broj *x* pojavljuje u nizu brojeva koji se unose sa standardnog ulaza sve do pojave nule. Napisati program koji učitava vrednost broja *x* i testira rad napisane funkcije.

### Primer 1

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

### Primer 2

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

[Rešenje 2.104]

**Zadatak 2.103** Napisati funkciju *long int fibonaci(int n)* koja računa  $n$ -ti element Fibonačijevog niza. Fibonačijev niz je niz za koji važi:  $F_0 = 1$ ,  $F_1 = 1$ ,  $F_{n+2} = F_{n+1} + F_n$  za  $n \geq 0$ . Napisati i program koji učitava ceo broj  $n$  ( $0 \leq n \leq 50$ ) i ispisuje traženi Fibonačijev broj.

*Primer 1*

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite broj n: 7
|| 21
```

*Primer 2*

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

[Rešenje 2.104]

**Zadatak 2.104** Napisati funkciju *char sifra(char c, int k)* koja za dati karakter  $c$  određuje šifru na sledeći način: ukoliko je  $c$  slovo, šifra je karakter koji se nalazi  $k$  pozicija iza njega u abecedi. U suprotnom karakter ostaje nepromenjen. Šifrovanje treba da bude kružno, što znači da je, na primer, šifra za  $c='b'$  i  $k=2$  karakter  $'z'$ . Napisati program koji učitava karakter po karakter do kraja ulaza (do pojave EOF koji se generiše kombinacijom CTRL+D) i ispisuje šifrovani tekst.

*Primer 1*

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

[Rešenje 2.104]

## 2.4 Rešenja

### Rešenje 2.1

```
2  /*
   * Napisati program koji za uneto vreme ispisuje koliko je sati i
   * minuta ostalo
```

## 2 Kontrola toka

---

```
do ponoci.
4  */
#include<stdio.h>
6
int main()
8 {
    int sati;
10    int minuti;
    int preostali_sati;
12    int preostali_minuti;

14    printf("Unesi vreme (broj sati u intervalu [0,24), broj minuta u
        intervalu [0,60]):");
    scanf("%d%d",&sati,&minuti);

16    preostali_sati = 24-sati-1;
18    preostali_minuti = 60-minuti;
    if (preostali_minuti==60)
20    {
        preostali_sati++;
        preostali_minuti=0;
22    }

24    printf("Do ponoci je ostalo %d sati i %d minuta\n", 24-sati-1, 60-
        minuti);
26    return 0;
}
```

### Rešenje 2.2

```
1  /*
    Napisati program koji za uneti ceo broj ispisuje njegovu reciprocnu
    vrednost.
3  Ukoliko je uneti broj jednak nuli, ispisati poruku "Nedozvoljeno
    deljenje nulom".
    */
5
#include <stdio.h>
7
int main()
9 {
    int x;
11    float rx;

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

15
    /*
17    obratiti paznju:
        x==0 - relacija jednakosti (da li je vrednost promenljive x
        jednaka nuli)
    */
}
```

```

19     x=0 - naredba dodele (promenljiva x dobija vrednost nula)
    */
21
22     if (x==0)
23         printf("Nedozvoljeno deljenje nulom\n");
24     else
25     {
26         rx = 1.0/x;
27         printf("Reciprocna vrednost unetog broja:%f\n",rx);
28     }
29
30     return 0;
31 }

```

### Rešenje 2.3

```

1  #include <stdio.h>
    /*
3  Napisati program koji za uneti ceo broj x ispisuje da li je jednak
    nuli,
    manji od nule ili veci od nule.
    */
5  int main()
6  {
7
8      int x;
9      printf("Unesi ceo broj:");
10     scanf("%d",&x);
11
12     /*
13         obratiti paznju:
14         x==0 - relacija jednakosti (da li je vrednost promenljive x
15         jednaka nuli)
16         x=0 - naredba dodele (promenljiva x dobija vrednost nula)
17     */
18     if (x==0)
19         printf("Broj je jednak nuli\n");
20     else if (x<0)
21         printf("Broj je manji od nule\n");
22     else
23         printf("Broj je veci od nule\n");
24
25     return 0;
26 }

```

### Rešenje 2.4

```

1  /*
    Napisati program koji za godinu koja se unosi sa standardnog ulaza
    na standardni izlaz

```

## 2 Kontrola toka

---

```
3  ispisuje da li je prestupna.
4  */
5
6  #include <stdio.h>
7
8  int main()
9  {
10     int x;
11     printf("Unesi godinu:");
12     scanf("%d",&x);
13
14     if ((x%4==0 && x%100!=0) || x%400==0)
15         printf("Godina je prestupna\n");
16     else
17         printf("Godina nije prestupna\n");
18     return 0;
19 }
```

### Rešenje 2.5

```
1  /*
2   Napisati program koji za 2 cela broja uneta sa standardnog ulaza
3   ispisuje njihov minimum na standardni izlaz.
4   */
5
6  #include <stdio.h>
7  int main()
8  {
9     int a,b;
10     int min1;
11     int min2;
12     int min3;
13
14     scanf("%d%d",&a,&b);
15
16
17     /* 1. nacin */
18     if (a<b)
19         min1=a;
20     else
21         min1=b;
22
23     printf("Minimum unetih brojeva (1.nacin) je %d\n",min1);
24
25     /* 2. nacin */
26     min2 = (a<b) ? a : b;
27     printf("Minimum unetih brojeva (2.nacin) je %d\n",min2);
28
29     /* 3. nacin */
30     min3=a;
31     if (b<a)
```



```

    min3 = b;
33  printf("Minimum unetih brojeva (3.nacin) je %d\n",min3);
35  return 0;
}

```

## Rešenje 2.6

```

/*
2  a) Napisati program koji za 3 cela broja uneta sa standardnog ulaza
   ispisuje njihov minimum na standardni izlaz.
4  b) Neka uneti brojevi predstavljaju cene artikla. Ukoliko se
   najjeftiniji
   artikal dobija za 1 dinar, napisati kolika je ukupna cena, kao i
   koliko
6  dinara se uštedi zahvaljujuci popustu.
*/
8
#include <stdio.h>
10 int main()
{
12     int a,b,c;
    int min;
14     int min1;
    int min2;
16     int cena_bez_popusta, cena_sa_popustom;

18     scanf("%d%d%d",&a,&b,&c);

20
    if (a<b)
22         if (a<c) /* poredak: a<b,a<c => a,b,c ili a,c,b */
            min=a;
24         else /* poredak: a<b, a>=c => a<b, c<=a => c,a,b */
            min=c;
26     else /* b<=a */
        if (b<c) /* poredak: b<=a,b<c => b,a,c ili b,c,a */
28            min=b;
        else /* poredak: b<=a, c<=b => c,b,a */
30            min=c;

32     printf("Minimum unetih brojeva (1.nacin) je %d\n",min);

34     /* 2. nacin */
    /* najpre odredimo minimum brojeva a,b*/
36     if (a<b)
        min1=a;
38     else
        min1=b;
40
    if (c<min1)

```

## 2 Kontrola toka

```
42     min1=c;
    printf("Minimum unetih brojeva (2.nacin) je %d\n",min1);
44
    /* 3. nacin */
46     min2=a;
    if(min2>b)
48         min2=b;
    if(min2>c)
50         min2=c;

52     printf("Minimum unetih brojeva (3.nacin) je %d\n",min2);

54     cena_bez_popusta=a+b+c;
    cena_sa_popustom = cena_bez_popusta - min2 + 1;

56     printf("Cena sa popustom: %.2f\n Cena bez popusta: %d\n Usteda:
    %.2f\n", cena_sa_popustom, cena_bez_popusta, cena_bez_popusta-
    cena_sa_popustom);
58
    return 0;
60 }
```

### Rešenje 2.7

```
/*
2  Napisati program koji za koeficijente kvadratne jednacine
   koji se unose sa standardnog ulaza na standardni izlaz
4  ispisuje koliko realnih resenja jednacina ima i ako ih ima, ispisuje
   resenja jednacine
   zaokruzena na dve decimale.
6  */
   #include <stdio.h>
8   #include <math.h>
   int main()
10  {
    float a,b,c;
12     float D;
    float x1,x2;
14     printf("Unesi koeficijente kvadratne jednacine:");
    scanf("%f%f%f",&a,&b,&c);
16

    /* proveravamo da li je kvadratna jednacina korektno zadata */
18     if (a==0)
        if (b==0)
20         if(c==0) /* slucaj a==0 && b==0 && c==0 */
            printf("Jednacina ima beskonacno mnogo resenja\n");
        else /* slucaj a==0 && b==0 && c!=0 */
22             printf("Jednacina nema resenja\n");
        else /* slucaj a==0 && b!=0 */
24         {
            x1=-c/b;
26
```

```

    printf("Jednacina ima jedinstveno realno resenje %.2f\n",x1)
;
28 }
else /* slucaj a!=0 */
30 {
    D=b*b-4*a*c; /* funkcija sqrt nalazi se u biblioteci math.h (
prevodjenje sa -lm opcijom) */
    if (D<0)
32     printf("Jednacina nema realnih resenja\n");
    else if (D>0)
34     {
        x1 = (-b+sqrt(D))/(2*a);
        x2 = (-b-sqrt(D))/(2*a);
36     printf("Jednacina ima dva razlicita realna resenja %.2f i %.2
f\n",x1,x2);
    }
    else
40     {
        x1 = (-b)/(2*a);
42     printf("Jednacina ima jedinstveno realno resenje %.2f\n",x1);
    }
44 }
46 }
48 return 0;
}

```

## Rešenje 2.8

```

1  /*
3  Napisati program koji za karakter koji ucitava jedan karakter i :
- u slucaju da je uneta cifra, ispisuje nju i njen ascii kod
5 - u slucaju da je uneto malo slovo, ispisuje njega, njegov ascii kod,
odgovarajuce veliko slovo i njegov ascii kod
- u slucaju da je uneto veliko slovo, ispisuje njega, njegov ascii
7 kod, odgovarajuce malo slovo i njegov ascii kod
- u ostalim slucajevima, ispisuje uneti karakter i njegov ascii kod
*/
9 #include <stdio.h>
int main()
11 {
    char c;
13     printf("Unesi jedan karakter:");
    scanf("%c", &c);
15
    if (c>='0' && c<='9')
17     printf("cifra:%c ascii:%d\n",c,c);
    else if (c>='A' && c<='Z')
19     printf("veliko slovo:%c ascii:%d odgovarajuce malo:%c, ascii:%d
\n",c,c,c-'A'+'a',c-'A'+'a'); /* Razlika izmedju ascii koda

```

## 2 Kontrola toka

```
        svakog malog i odgovarajućeg velikog slova

                                je konstanta koja se može
        sracunati izrazom 'a'-'A' (i iznosi 32) */
21 else if (c>='a' && c<='z')
        printf("malo slovo:%c ascii:%d odgovarajuće veliko:%c, ascii:%d
        \n",c,c,c-'a'+'A',c-'a'+'A');
23 else
        printf("karakter:%c ascii:%d\n",c,c);
25
        return 0;
27 }
```

### Rešenje 2.9

```
1  /*
3  Napisati program koji učitava tri cela broja i ispisuje zbir onih
    unetih brojeva
    koji su pozitivni.
5
6  */
7  #include<stdio.h>
8  int main()
9  {
10     int a,b,c;
11     int s;
12     printf("Unesi prvi ceo broj:");
13     scanf("%d",&a);
14     printf("Unesi drugi ceo broj:");
15     scanf("%d",&b);
16     printf("Unesi treci ceo broj:");
17     scanf("%d",&c);
18
19     s=0; /* inicijalizujemo promenljivu s na nulu */
20
21     if (a>0)
22         s=s+a; /* naredba dodele: vrednost izraza a desne strane znaka
                jednakosti
23                 dodeljujemo promenljivoj sa leve strane znaka
                jednakosti.
                Staru vrednost promenljive s sabereemo sa vrednoscu
                promenljive a
25                 i dobijenu vrednost upisemo u promenljivu s */
26
27     if (b>0)
28         s+=b; /* operator +=
                s+=b je skraceni zapis za s=s+b
29                 */
30
31     if (c>0)
```

```
33     s+=c;
35     printf("Suma unetih pozitivnih brojeva: %d\n",s);
36     return 0;
37 }
```

### Rešenje 2.10

```
1  /*
3  Napisati program koji za realan broj unet sa standardnog ulaza
   ispisuje njegovu apsolutnu vrednost.
5
6  */
7
8  #include<stdio.h>
9  #include<math.h>
10 #include<stdlib.h>
11 int main()
12 {
13     float x;
14     float y;
15
16     printf("Unesi jedan realan broj:");
17     scanf("%f",&x);
18
19     /* 1. nacin */
20     if (x>0)
21         y=x;
22     else
23         y=-x;
24     printf("Apsolutna vrednost broja %f je %f\n",x,y);
25
26     /* 2. nacin */
27     y=x;
28     if (y<0)
29         y=-y;
30
31     printf("Apsolutna vrednost broja %f je %f\n",x,y);
32
33     /* 3. nacin - pogresan! */
34     y=abs(x); /* funkcija abs vraca ceo broj! za racunanje apsolutne
35                vrednosti realnog broja treba koristiti funkciju fabs */
36     /* funkcija abs se nalazi u zaglavlju stdlib.h */
37     printf("Apsolutna vrednost broja %f je %f\n",x,y);
38
39     /* 4. nacin */
40     y=fabs(x); /* funkcija fabs se nalazi u zaglavlju math.h */
41     printf("Apsolutna vrednost broja %f je %f\n",x,y);
42     return 0;
43 }
```

### Rešenje 2.11

```
2  /*
   Napisati program koji poziva korisnika da unese jedan karakter i
       ispisuje
4  da li je uneti karakter samoglasnik.
   */
6
   #include <stdio.h>
8
   int main()
10 {
       char c;
12     printf("Unesi jedan karakter:");
       scanf("%c", &c);
14     switch(c)
       {
16         case 'A' :
17         case 'E' :
18         case 'I' :
19         case 'O' :
20         case 'U' :
21         case 'a' :
22         case 'e' :
23         case 'i' :
24         case 'o' :
25         case 'u' : printf("Uneli ste samoglasnik\n");
26                     break;
27         default : printf("Niste uneli samoglasnik\n");
28                     break;
29     }
30
       return 0;
32 }
```

### Rešenje 2.12

```
1  /*
   Napisati program koji za uneti dan i mesec ispisuje godisnje doba kom
       pripadaju. Mozemo podrazumevati da je unos korektan.
3  */
5
   #include <stdio.h>
7
   int main()
9 {
```

```
11  int d,m;
    printf("Unesi dan i mesec");
    scanf("%d%d",&d,&m);
13
    switch(m) /* argument u naredbi switch mora biti celobrojna
               promenljiva */
15  {
        case 1: /* argument u naredbi case mora biti celobrojna
                  konstanta */
17        case 2: /* ispitujemo da li je m==2 */
            printf("zima\n");
            break;
19        case 3:
            if (d<21)
21            printf("zima\n");
            else
23            printf("prolece\n");
            break;
25        case 4:
        case 5:
            printf("prolece\n");
            break;
27        case 6:
            if (d<21)
31            printf("prolece");
            else
33            printf("leto");
            break;
35        case 7:
        case 8:
            printf("leto");
            break;
37        case 9:
            if (d<23)
41            printf("leto\n");
            else
43            printf("jesen\n");
            break;
45        case 10:
        case 11:
            printf("jesen\n");
            break;
47        case 12:
            if (d<22)
51            printf("jesen\n");
            else
53            printf("zima\n");
55    }
    return 0;
57 }
```

### Rešenje 2.13

```
1  /*
2  Napisati program koji od korisnika zahteva da unese
3  cetvorocifreni broj. Program za taj broj proverava
4  da li su cifre uredjene rastuce, opadajuce ili nisu
5  uredjene i stampa odgovarajucu poruku na standardni
6  izlaz. Voditi racuna o nekorektnim unosima. Na primer,
7  pokretanje programa moze da izgleda ovako:
8
9  Unesi jedan cetvorocifreni broj: -1357
10 Cifre su mu uredjene neopadajuce.
11
12 ili ovako
13
14 Unesi jedan cetvorocifreni broj: 9952
15 Cifre su mu uredjene nerastuce.
16
17 ili ovako
18
19 Unesi jedan cetvorocifreni broj: 9572
20 Cifre su mu nisu uredjene.
21
22 Unesi jedan cetvorocifreni broj: 123
23 Uneti broj nije cetvorocifren.
24
25 */
26
27 #include <stdio.h>
28 #include <stdlib.h>
29
30 int main()
31 {
32     int x;
33     char c1;    /* cifre su brojevi {0,1,2,3,4,5,6,7,8,9} */
34     char c10;
35     char c100;
36     char c1000;
37
38     printf("Unesi jedan cetvorocifreni broj:");
39     scanf("%d", &x);
40
41     x=abs(x); /* u slucaju da je broj negativan, uzimamo njegovu
42                apsolutnu vrednost
43                kako ne bismo za cifre dobili negativne brojeve */
44                /* funkcija abs nalazi se u zaglavlju stdlib.h */
45
46     if (x<1000 || x>9999)
47         printf("Uneti broj nije cetvorocifren\n");
48     else
49     {
```



```

51     c1 = x%10;
    c10 = (x/10)%10;
    c100 = (x/100)%10;
53     c1000 = (x/1000)%10;

55     printf("Cifre broja: %d,%d,%d,%d\n",c1000,c100,c10,c1);

57     if (c1000<=c100 && c100<=c10 && c10<=c1)
        printf("Cifre su uredjene neopadajuće \n");
59     else if (c1000>=c100 && c100>=c10 && c10>=c1)
        printf("Cifre su uredjene nerastuće \n");
61     else
        printf("Cifre nisu uredjene\n");
63 }
    return 0;
65 }

```

### Rešenje 2.14

```

1  /*
   Sa standardnog ulaza unose se jedan karakter i 8 realnih brojeva koji
   predstavljaju
3  koordinate cetiri tacke: A(x1, y1), B(x2, y2), C(x3, y3), D(x4, y4).
   Na osnovu unetog karaktera
   ispisuje se odgovarajuca poruka na standardni izlaz:
5  k - proverava da li su date tacke temena pravougaonika cije su
   stranice paralelne koordinatnim osama i
   u slucaju da jesu, ispisuje obim datog pravougaonika; mozemo
   podrazumevati da ce korisnik koordinate tacaka
7  unosi redom A,B,C,D, pri cemu ABCD opisuje pravougaonik cije su
   stranice AB,BC,CD i DA, a dijagonale AC i BD
   na primer, tacke (1,1),(2,1),(2,2),(1,2) cine pravougaonik cije
   su stranice paralelne koordinatnim osama i ciji je obim 4
9  a tacke (1,1),(2,2),(3,3),(4,4) ne cine pravougaonik
   h - proverava da li su unete tacke kolinearne i ukoliko jesu,
   ispisati jednacinu prave kojoj pripadaju
11  na primer, tacke (1,2),(2,3),(3,4),(4,5) su kolinearne i
   pripadaju pravoj y=x+1
   tacke (1,1),(1,2),(1,3),(1,4) su kolinearne i pripadaju pravoj x
   =1
13  a tacke (1,1),(2,1),(2,2),(1,2) nisu kolinearne
   j - Kramerovim pravilom proverava da li je dati sistem jednačina
15  x1 * p + x2 * q = x4 - x3
   y1 * p + y2 * q = y4 - y3
17  odredjen, neodredjen ili nema resenja, i u slucaju da je odredjen
   ispisati resenja.
   na primer, za unete koordinate (1,1),(1,1),(1,0),(2,2) sistem
   nema resenja
19  za unete koordinate (1,1),(1,1),(1,1),(1,1) sistem je
   neodredjen ili nema resenja

```

```

                za unete koordinate (6,1),(8,3),(10,-4),(9,1) sistem
                ima jedinstveno resenje 4.30, 3.10
21
22 */
23
24 #include<stdio.h>
25 #include<math.h>
26 int main()
27 {
28     char c;
29     float x1,y1,x2,y2,x3,y3,x4,y4;
30     float kab,kbc,kad;
31     float dab,dad;
32     float delta, deltap, deltaq;
33     float 0;
34     float k,n;
35
36     printf("Unesi jedan karakter:");
37     scanf("%c",&c);
38
39     printf("Unesi realne koordinate 4 tacke:");
40     scanf("%f%f%f%f%f%f%f", &x1,&y1,&x2,&y2,&x3,&y3,&x4,&y4);
41
42     switch (c)
43     {
44         case 'k':
45             if (y1==y2 && y3==y4 && x1==x4 && x2==x3)
46             {
47                 dab = sqrt(pow(x1-x2,2)+pow(y1-y2,2)); // funkcija pow(x
48                 ,y) racuna vrednost stepene funkcije x^y
49                 dad = sqrt(pow(x1-x4,2)+pow(y1-y4,2)); // x i y su
50                 realne vrednosti
51                 0 = 2*dab + 2*dad;
52                 printf("Obim pravougaonika je %f\n",0);
53             }
54             else
55                 printf("Tacke ne cine pravougaonik sa stranicama koje su
56                 paralelne koordinatnim osama\n");
57                 break;
58             case 'h':
59                 if ((x1-x2)!=0) // ukoliko se tacke A(x1,y1) i B(x2,y2) ne
60                 nalaze na pravoj koja je paralelna x osi
61                 {
62                     k = (y1-y2)/(x1-x2); //izracunamo k,n za pravu odredjenu
63                     tackama A(x1,y1) i B(x2,y2)
64                     n = y1-k*x1;
65
66                     if (y3==x3*k+n && y4==x4*k+n) // proverimo da li tacke
67                     C(x3,y3) i D(x4,y4) nalaze na toj pravoj
68                     printf("Tacke su kolinearne, pripadaju pravoj y=%f*x
69                     +%f\n",k,n);
70                 }
71                 else

```

```

        printf("Tacke nisu kolinearne\n");
65     }
    else // ukoliko se A i B nalaze na pravoj koja je paralelna
        x osi
67         if (x3==x1 && x4==x1) // proverimo da li tacke C(x3,y3)
            i D(x4,y4) nalaze na toj pravoj
                printf ("Tacke su kolinearne, pripadaju pravoj x=%f\n
",x1);
69         else
            printf("Tacke nisu kolinearne\n");
71         break;
    case 'j':
73         delta = x1*y2-x2*y1;
        deltap = x2*(y4-y3)-y2*(x4-x3);
75         deltaq = x1*(y4-y3)-y1*(x4-x3);
        if (delta!=0)
77             printf("Sistem ima jedinstveno resenje %.2f, %.2f\n",
deltap/delta, deltaq/delta);
        else if (deltap==0 && deltaq==0)
79             printf("Sistem je neodredjen ili nema resenja.\n");
        else
81             printf("Sistem nema resenja\n");
            break;
83     default:
        printf("Nekorektan unos\n");
85 }
    return 0;
87 }

```

Rešenje 2.31

Rešenje 2.31

Rešenje 2.31

Rešenje 2.31

Rešenje 2.31

Rešenje 2.31

Rešenje 2.31

Rešenje 2.31

Rešenje 2.31

Rešenje 2.31

Rešenje 2.31

Rešenje 2.31

Rešenje 2.31

Rešenje 2.31

Rešenje 2.31

Rešenje 2.31

Rešenje 2.31

Rešenje 2.32

```
1  /*
2  Napisati program koji 10 puta ispisuje tekst "We love C programming".
3  */
4
5  #include<stdio.h>
6
7  int main()
8  {
9      int i=0;          /* promenljiva i kontrolise koliko puta ce se petlja
10                       izvrsiti */
11      while (i<10) /* pre ulaska u telo petlje proverava se da li je */
12      {             /* ispunjen uslov petlje */
13          printf("We love C programming\n");
14          i++; /* operator ++ uvecava i promenljivu za 1
15              i++; ima isti efekat kao i=i+1;
16              ili i+=1;
17
18              ukoliko ne bismo menjali vrednost promenljive i doslo
19              bi
20              do beskonacne petlje!
21          */
22      }
```

```

23     brojanje u while petlji smo mogli realizovati i preko uslova:

25     i=1;
26     while(i<=10)
27     {
28         ...
29     }

31     ili

33     i=2;
34     while(i<=11)
35     {
36         ...
37     }

39     ili

41     i=3;
42     while(i<13)
43     {
44         ...
45     }

47     Brojanje pocev od 0 uz koriscenje stroge nejednakosti
48     je u duhu programskog jezika C i zato cemo ovaj nacin
49     brojanja najcesce koristiti

51     */
52     return 0;
53 }

```

### Rešenje 2.33

```

/*
2  Napisati program koji poziva korisnika da unese pozitivan ceo broj n
3  a potom ispisuje brojeve od 0 do n-1.
4  */

6  #include<stdio.h>

8  int main()
9  {
10     int x;
11     int n;

12     printf("Unesi pozitivan ceo broj:");
13     scanf("%d", &n);
14     x=0;
15     while (x<n)
16     {
17         printf("%d\n", x);
18     }

```

```
        x++;
20    }
    return 0;
22 }
```

### Rešenje 2.34

```
/*
2   Napisati program koji za uneti pozitivan ceo broj
   izracunava njegov faktoriyel. Testirati program
4   za razlicite vrednosti promenljive x. Obratiti paznju
   da pocev od 23! dolazi do prekoračenja.
6 */

8 #include<stdio.h>

10 int main()
{
12     int x;
    unsigned long f;
14     int i;
    int original;

16     printf("Unesi x>=0:");
18     scanf("%d",&x);

20     original=x;
    f=1;
22     if (x<0)
        printf("Nekorektan unos\n");
    else
24     {
        while (x>1)
26         {
            f=f*x; /* vrednost izraza sa desne strane naredbe dodele
28                    dodeljujemo promenljivoj sa leve strane naredbe
                    dodele
30                    */
            x--; /* operator -- umanjuje vrednost promenljive x za 1
32                    naredba x--; ima isti efekat kao x-=1;
                    ili x=x-1;
34                    */
        }
36     printf("%d! = %lu\n",x,f); /* nekorektno: vrednost
    promenljive x je unistena */
    printf("%d! = %lu\n",original,f); /* korektno: promenljiva
38     original sadrzi vrednost promenljive x pre ulaska u petlju */

40 }
```

```
42     return 0;
    }
```

### Rešenje 2.35

```
1  /*
   * Napisati program koji poziva korisnika da unese pozitivan ceo broj
   * n,
   * a zatim za unetih n celih brojeva ispisuje sumu pozitivnih i sumu
   * negativnih brojeva.
   */
7  #include<stdio.h>
9
11 int main()
12 {
13     int n;
14     int x;
15     int suma_poz;
16     int suma_neg;
17     int i;
18
19     printf("Unesi pozitivan ceo broj:");
20     scanf("%d",&n);
21
22     suma_poz=0; /* promenljivim koje ce sadrzati sumu se pre ulaska u
   * petlju */
23     suma_neg=0; /* dodeljuje se 0 (neutral za sabiranje) */
24     i=0;
25
26     while(i<n)
27     {
28         printf("Unesi ceo broj:");
29         scanf("%d", &x);
30
31         if (x<0)
32             suma_neg+=x;
33         else
34             suma_poz+=x;
35
36         i++;
37     }
38
39     printf(" Suma pozitivnih: %d\n Suma negativnih: %d\n",suma_poz,
40           suma_neg);
41     return 0;
42 }
```

### Rešenje 2.36

```
2  /*
   Napisati program koji omogućava korisniku da unosi cele brojeve dok
   ne unese nulu. Nakon toga ispisati proizvod onih unetih brojeva
   koji
4   su pozitivni.
   */
6
   #include <stdio.h>
8  int main()
   {
10     int x;
       int p;
12
       p=1;
14     while (1) /* izraz 1 je konstantan; razlicit je od nule sto znaci
       da ga tumacimo kao tacnog */
       {
16         printf("Unesi jedan ceo broj:");
           scanf("%d", &x);
18         if (x==0) /* ukoliko je uneta nula */
           break; /* break prekidamo petlju; izvorsavanje se nastavlja
           od prve naredbe nakon petlje */
20
           if (x<0) /* ukoliko je unet negativan broj, tu vrednost ne
           zelimo da pomnozimo sa ukupnim proizvodom p; zato moramo
           nastaviti dalje */
22             continue; /* sa izvorsavanjem petlje; continue prekida
           trenutnu iteraciju petlje tako sto preskace sve naredbe
           koje nakon njega slede; izvorsavanje se
           nastavlja od provere uslova petlje */
24         p=p*x;
       }
26
       printf("Proizvod unetih brojeva je %d\n",p);
28
       return 0;
30 }
```

### Rešenje 2.37

```
2  /*
   Napisati program koji za uneti ceo broj ispisuje njegove cifre
   u obrnutom poretku.
4  */
6
   #include<stdio.h>
   #include<stdlib.h>
8  int main()
```



```

{
10     int x;
    char cifra;
12     printf("Unesi ceo broj:");
    scanf("%d", &x);
14
    x = abs(x); /* pretvaranje u apsolutnu vrednost se vrši za slučaj
        kada je unet
16                negativan broj kako bismo osigurali da će nam
        izdvojene cifre
            biti pozitivne
18                */

20     while(x>0)
    {
22         cifra=x%10;          /* izdvajamo poslednju cifru broja x */
        printf("%d\n", cifra);
24         x/=10;              /* ako je npr x=1582, x%10 će biti 2,
                                a x/10 će biti 158;
                                npr x=5, x%10 će biti 5
                                a x/10 će biti 0 */
26
28     }

30     return 0;
}

```

### Rešenje 2.38

```

1  /*
    Napisati program koji omogućava korisniku da unosi karaktere dok ne
        zada tacku i ukoliko je karakter malo slovo,
3  ispisuje odgovarajuće veliko, ukoliko je karakter veliko slovo
        ispisuje odgovarajuće malo, a u suprotnom ispisuje
        isti karakter kao i uneti.
5  */

7  #include <stdio.h>

9  int main()
    {
11     int c;

13     /* funkcija getchar učitava jedan karakter.
        naredbom dodele (c=getchar()) promenljivoj c bice dodeljena
        vrednost
15         ascii koda unetog karaktera
        obratiti paznju na zagrade!
17     */
    while((c=getchar())!='.')
19     {
        if (c>='A' && c<='Z')

```

## 2 Kontrola toka

```
21     putchar(c+'a'-'A'); /* Razlika izmedju ascii koda svakog malog
    i odgovarajuceg velikog slova
                                je konstanta koja se moze sracunati
    izrazom 'a'-'A' (i iznosi 32) */
23     else if (c>='a' && c<='z')
        putchar(c-'a'+'A');
25     else
        putchar(c);
27 }
29 return 0;
}
```

### Rešenje 2.39

```
/*
2   Napisati program koji omogucava korisniku da unosi karaktere dok
   ne zada EOF a potom ispisuje broj velikih slova, broj malih slova
   , broj cifara, broj belina i zbir cifara.
4   */
6   #include <stdio.h>
8
10  int main()
11  {
12      /* promenljivoj c dodelicemo povratnu vrednost funkcije getchar()
        funkcija getchar() ucitava jedan karakter sa standardnog ulaza
        i vraca njegov ascii kod; povratna vrednost funkcije getchar je
14      int, pa i promenljiva c mora biti tipa int
        */
16      int c;
18
        /* brojac moraju biti inicijalizovani na 0 */
19      int br_v=0;
20      int br_m=0;
21      int br_c=0;
22      int br_b=0;
23      int br_k=0;
24      int suma=0;
26
        while((c=getchar())!=EOF) /* petlja se zavrшава kada
            korisnik ne unese karakter, vec zada konstantu EOF */
        { /* ova konstanta se zadaje
            kombinacijom tastera CTRL+D. U tom slucaju, getchar() vraca -1*/
28            if (c>='A' && c<='Z')
                br_v++; /* <=> br_v = br_v+1; */
30            else if (c>='a' && c<='z')
                br_m++;
32            else if (c>='0' && c<='9')
            {
```

```

34     br_c++;
        suma=suma+c-'0';           /* funkcija getchar() vraca ascii
36     kod unetog karaktera; ascii kodovi cifara 0,1,...,9
                                   su redom 48,49,...,57; Na primer,
        za unetu 1
                                   promenljiva c ce imati vrednost
        49. Zbog toga bi bilo pogresno racunati
38     zbir kao zbir=zbir+c. Promenljivu zbir zato
        racunamo kao zbir=zbir+(c-'0')
                                   jer c-'0' ce za unetu 0 proizvesti 48-'0' sto je
        0,
                                   za unetu 1 49-'0' sto je 1, za unetu 2 50-'0' sto
40     je 2, ...*/
    }
42     else if (c=='\t' || c=='\n' || c==' ')
        br_b++;
44
        br_k++;
46    }

48    printf("velika: %d, mala: %d, cifre: %d, beline: %d, svi: %d\n",
        br_v, br_m, br_c, br_b, br_k);
    printf("suma cifara: %d\n", suma);
50
    return 0;
52 }

```

### Rešenje 2.40

```

/* Niz prirodnih brojeva formira se na sledeci nacin:
2  an+1 = an/2 ako je an parno
   an+1 = (3*an+1)/2 ako je an neparno
4  Napisati program koji za uneti pocetni clan niza a0 stampa niz
   brojeva sve do prvog clana jednakog
1.
6  */
#include<stdio.h>
8  int main()
{
10     int a0;
        int an,an1;
12
        printf("Unesi pocetni clan niza brojeva:");
14     scanf("%d",&a0);

16     if (a0>0)
        {
18         printf("%d\n", a0);

20         an=a0;
        while(an!=1)

```

## 2 Kontrola toka

```
22     {
23         if (an%2) /* ukoliko je vrednost izraza an%2 razlicita od nule,
24         */
25         { /* izraz se tumaci kao tacan i izvrsavaju se naredbe
26         iz if grane */
27             an1=(3*an+1)/2;
28         }
29         else /* u suprotnom, ukoliko je vrednost izraza an%2 jednaka
30         nuli, izraz */
31         { /* se tumaci kao netacan i izvrsavaju se naredbe iz else
32         grane */
33             an1=an/2;
34         }
35         printf("%d\n",an1);
36         an=an1;
37     }
38 }
39
40 else
41     printf("Nekorektan unos\n");
42
43 return 0;
44 }
```

### Rešenje 2.41

```
1  /*
2  Napisati program koji za uneti ceo broja n ispisuje n puta tekst
3  "We love C programming" koriscenjem while, for i do while petlje.
4  Obratiti paznju
5  na rezultat kada je n<=0.
6  */
7
8  #include <stdio.h>
9
10 int main()
11 {
12
13     int n,m;
14     int i;
15
16     printf("Unesi ceo broj:");
17     scanf("%d",&n);
18
19     /* 1. nacin - while petlja */
20     printf("while: ");
21
22     i=0;
23     while (i<n) /* uslov petlje se proverava pre ulaska u
24     telo petlje */
25     {
26         printf("We love C programming\n");
27     }
```

```

25     i++;
26 }
27
28 printf("\n");
29
30 /* 2. nacin - for petlja */
31 printf("for: ");
32
33     /* naredba i=0 se izvršava jednom, pre prve
34     iteracije */
35     for(i=0; i<n; i++) /* uslov petlje i<=m se proverava pre svake
36     iteracije */
37         printf("We love C programming\n"); /* naredba i++ se izvršava
38     nakon svake iteracije */
39
40 printf("\n");
41
42 /* 3. nacin - do while petlja */
43 printf("do while: "); /* uslov petlje se proverava na kraju svake
44     iteracije */
45
46     /* zbog toga se do while petlja izvršava
47     bar jednom, cak i u slucaju */
48     /* da uslov petlje nikada nije ispunjen */
49     i=0;
50     do /* petlja se zapocinje bez
51     provere uslova */
52     {
53         printf("We love C programming\n"); /* stampa se dati tekst */
54         i++; /* uvecava se vrednost
55     promenljive i */
56     }
57     while(i<n); /* proverava se uslov i
58     ukoliko je ispunjen, nastavlja se sa sledecom iteracijom */
59     /* u suprotnom, petlja se
60     završava i program se nastavlja od prve naredbe koja sledi za
61     petljom */
62     printf("\n");
63
64     return 0;
65
66 }

```

### Rešenje 2.42

```

2  /*
3  Napisati program koji za uneta dva cela broja n i m ispisuje sve
4  cele brojeve
5  iz intervala [n,m] koriscenjem while, for i do while petlje.
6  Obratiti paznju
7  na rezultat kada je n>m.
8  */

```

```
#include <stdio.h>

8
int main()
10 {

12     int n,m;
    int i;

14

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

18     /* 1. nacin - while petlja */
    printf("while: ");

20

    i=n;
22     while (i<=m)           /* uslov petlje se proverava pre ulaska u
        telo petlje */
    {
24         printf("%d ", i);
        i++;
26     }

28     printf("\n");

30     /* 2. nacin - for petlja */
    printf("for: ");

32                                /* naredba i=n se izvrsava jednom, pre prve
        iteracije */
    for(i=n;i<=m;i++)         /* uslov petlje i<=m se proverava pre svake
        iteracije */
34         printf("%d ", i);    /* naredba i++ se izvrsava nakon svake
        iteracije */

36     printf("\n");

38     /* 3. nacin - do while petlja */
    printf("do while: "); /* uslov petlje se proverava na kraju svake
        iteracije */

40                                /* zbog toga se do while petlja izvrsava
        bar jednom, cak i u slucaju */
                                /* da uslov petlje nikada nije ispunjen */

42     i=n;
    do                        /* petlja se zapocinje bez provere uslova */
44     {
        printf("%d ",i); /* stampa se vrednost promenljive i */
46         i++;           /* uvecava se vrednost promenljive i */
    }

48     while(i<=m);          /* proverava se uslov i ukoliko je ispunjen,
        nastavlja se sa sledecom iteracijom */
                                /* u suprotnom, petlja se zavrsava i program
        se nastavlja od prve naredbe koja sledi za petljom */
50     printf("\n");
```

```

    return 0;
52 }

```

### Rešenje 2.43

```

/*
2 Program izracunava minimum n unetih brojeva.
  Npr. za n=4 i brojeve 3 8 2 9 program ispisuje 2
4 */
#include <stdio.h>
6 int main()
{
8     int n, i;
    float x, min;
10
12     printf("Unesi n>0:");
    scanf("%d", &n);
14     if (n<=0)                                /* ako je unos neispravan */
    {
16         printf("Neispravan unos\n");
        return -1;                                /* prekidamo izvršavanje
    programa pomocu naredbe return */
18     }
    /* u slucaju greske kao sto je
    neispravan unos vracamo vrednost -1 */
    printf("Unesi realan broj:");
20     scanf("%f", &x);                                /* prvi broj je unet izvan petlje */
    min=x;                                /* kako bi bio njegova vrednost bila
    dodeljena promenljivoj min */
22                                /* neophodno je da promenljiva min
    bude inicijalizovana pre ulaska u petlju */
                                /* da bi uslov x<min mogao da bude
    ispitan u prvoj iteraciji */
24     i=0;
    while(i<(n-1))
26     {
        printf("Unesi realan broj:");
28         scanf("%f", &x);
        if(x<min)
30             min=x;
        i++;
32     }
    printf("Minimum je: %f\n", min);
34     return 0;
}

```

### Rešenje 2.44

```

/*

```

## 2 Kontrola toka

```
2      a) Napisati program za uneti pozitivan ceo broj n ispisuje zbir
      s = 1+2^3+3^3+...+n^3. Na primer, za n=4, izlaz iz programa
4      treba da bude:
      Suma kubova od 1 do 4 je 100
6      b) Modifikovati program tako da ispisuje zbir s = 1+2^3+3^3+...+k
      ^3
      za svako i od 1 do n. Na primer, za n=4, izlaz iz programa
      treba da
8      bude:
      i=1, n=1
10     i=2, n=9
      i=3, n=36
12     i=4, n=100

14     */

16     #include <stdio.h>

18     int main()
19     {
20         int n;
21         int i;
22         int s;

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

28         if (n<0)
29             return -1;

30         i=1;
32         s=0; /* inicijalizacija promenljive u kojoj se cuva suma kubova */

34         for(i=1;i<=n;i++)
35         {
36             s+=i*i*i;
37             /* b) */
38             printf("i=%d, s=%d\n", i, s);
39         }
40         /* a) */
41         printf("Suma kubova od 1 do %d: %d\n", n, s);
42         return 0;
43     }
```

### Rešenje 2.45

```
1     /*
      Napisati program koji ispisuje sve prave delioce unetog pozitivnog
      celog broja.
3     */
```



```
5  */
6
7  #include<stdio.h>
8
9  #include<math.h>
10
11 int main()
12 {
13     int x;
14     int i;
15
16     printf("Unesi x>0:");
17     scanf("%d", &x);
18
19     if (x<=0)
20     {
21         printf("Neispravan unos\n");
22         return -1;
23     }
24
25     /* 1. nacin */
26     printf("----- 1. nacin ----- \n");
27     for(i=2;i<x;i++)
28     {
29         printf("proveravam za %d...\n",i);
30         if (x%i==0)
31             printf("\t delilac:%d \n",i);
32     }
33     /* 2. nacin (brzi) */
34     printf("----- 2. nacin ----- \n");
35     for(i=2;i<=sqrt(x);i++)
36     {
37         printf("proveravam za %d...\n",i);
38         if (x%i==0)
39             if (i==x/i) /* u slucaju kada je delilac koren broja, npr 4
40                        za 16, ispisujemo ga jednom */
41                 printf("\t delilac:%d \n",i);
42             else /* u suprotnom, npr 2 za 16, ispisujemo i 2 i 8
43                 */
44                 printf("\t delioci:%d %d \n",i,x/i);
45     }
46     return 0;
47 }
```

Rešenje 2.71

Rešenje 2.71

Rešenje 2.71

Rešenje 2.71

Rešenje 2.71

Rešenje 2.71

Rešenje 2.71

Rešenje 2.71

Rešenje 2.71

Rešenje 2.71

Rešenje 2.71

Rešenje 2.71

Rešenje 2.71

Rešenje 2.71

Rešenje 2.71

Rešenje 2.71

Rešenje 2.71

Rešenje 2.71

Rešenje 2.71

Rešenje 2.71

Rešenje 2.71

Rešenje 2.71

Rešenje 2.71

Rešenje 2.71

Rešenje 2.71

Rešenje 2.71

Rešenje 2.72

```
1  #include <stdio.h>
3  int kvadrat(int x)
4  {
5      /* promenljive u listi argumenata funkcije, kao i one
6       deklarisanе u samoj funkciji, lokalne su za tu funkciju
7       sto znaci da se promenljive x i y neće "videti" nigde izvan
8       funkcije kvadrat (ni u funkciji main ni u funkciji kub)
9       */
11     int y;
12     y = x*x;
13     return y;
14 }
15
16 int kub(int a)
17 {
18     /*
19      u listi argumenata funkcije mozemo, a ne moramo, imati
20      promenljivu
21      istog naziva kao promenljiva koja je deklarisanа u main
22      funkciji
23      (u ovom slucaju promenljiva a); ova promenljiva se razlikuje
24      od promenljive a deklarisanе u main funkciji i vidljiva je
25      samo unutar funkcije kub
26      */
27     return a*a*a;
28 }
29
30 int main()
31 {
32     int a, kv, kb;
33     printf("Unesi ceo broj:");
34     scanf("%d", &a);
```

## 2 Kontrola toka

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

37 printf("Kvadrat broja %d je %d, a njegov kub je %d\n", a, kv, kb);
    return 0;
39 }
```

### Rešenje 2.73

```
1 /*
    Napisati program koji za uneti realan broj x i ceo broj n ispisuje
    3 vrednost stepena x^n. Unosenje promenljivih, racunanje stepena i
    ispis promenljivih realizovati u posebnim funkcijama.
    */
5
7 #include <stdio.h>
    #include <stdlib.h>
9
11 float stepen(float a, int b)
    {
13     float s=1;
        int i;

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

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

21
23 int main()
    {
25     int n;
        float x;
        float s;

27
29     printf("Unesi jedan realan i jedan ceo broj:");
    scanf("%f%d",&x,&n);

31
33     s = stepen(x,n);

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

    return 0;
39 }
```

## Rešenje 2.74

```
1  /*
2  Napisati funkciju koja za dva data cela broja odredjuje
3  najveći zajednički delilac. Napisati potom glavni program
4  koji testira ovu funkciju.
5  */
6
7  #include <stdio.h>
8
9  int euklid(int x, int y)
10 {
11     int r;
12     /* Euklidov algoritam */
13     while(y) /* algoritam se zaustavlja kada vrednost */
14     {       /* promenljive y postane nula */
15         r=x%y;
16         x=y;
17         y=r;
18     }
19
20     return x; /* nzd je sacuvan u promenljivoj x */
21 }
22
23 int main()
24 {
25     int a,b;
26     int nzd;
27
28     printf("unesi dva cela broja:");
29     scanf("%d%d", &a,&b);
30
31     nzd = euklid(a,b); /* promenljivoj nzd dodeljujemo povratnu
32                        vrednost funkcije euklid */
33
34     printf("najveci zajednicki delilac za %d i %d je %d\n", a,b,nzd);
35
36     return 0;
37 }
```

## Rešenje 2.75

```
1  /*
2  Napisati funkciju koja za dato n vraca zbir reciprocnih vrednosti
3  brojeva od 1 do n.
4  Napisati program koji omogucava korisniku da unese prirodan broj n, a
5  potom ispisuje zbir reciprocnih
```

## 2 Kontrola toka

```
4 vrednosti brojeva od 1 do n koristeći funkciju float zbir_reciprocnih
   (int n). Rezultat zaokružiti
na dve decimalne.
6 */

8 #include <stdio.h>

10 float zbir_reciprocnih(int n)
{
12     float z=0;
    int i;
14     for(i=1;i<=n;i++)
        z+=1.0/i; /* da bismo dobili reciprocnu vrednost broja, vazno je
                   da izbegnemo celobrojno deljenje dva cela broja */
16     return z; /* tako sto ce npr deljenik biti 1.0 umesto 1 */
}

18
20 int main()
{
    int n;
22     printf("Unesi jedan pozitivan ceo broj:\n");
    scanf("%d", &n);
24     printf("Zbir reciprocnih vrednosti brojeva od 1 do %d je %.2f\n", n
        , zbir_reciprocnih(n));
    /* povratna vrednost funkcije zbir_reciprocnih je float; funkciju
       mozemo pozvati u okviru
26     naredbe printf i umesto specifikatora %.2f bice ispisana
       povratna vrednost funkcije
       zbir_reciprocnih zaokruzena na dve decimalne */
28     return 0;
}
```

### Rešenje 2.76

```
1 /*
   Napisati funkciju koja racuna aritmeticku sredinu cifara datog celog
   broja.
3 Napisati potom glavni program koji omogucava korisniku da unese ceo
   broj
   i racuna aritmeticku sredinu njegovih cifara primenom napisane
   funkcije. Ispisati
5 izracunatu vrednost zaokruzenu na dve decimalne.
   */

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

11 float aritmeticka_sredina(int x)
{
13     int zbir_cifara=0;
    int broj_cifara=0;
```

```

15  char cifra;

17  if (x==0)      /* u slucaju da je uneta 0 */
    return 0; /* aritmeticka sredina cifara iznosi 0 i tu vrednost
    vracamo */

19

21  x=abs(x); /* uzimamo apsolutnu vrednost broja za slucaj da je
    negativan */

23  while(x)
  {
25      cifra=x%10;

27      broj_cifara++;
      zbir_cifara+=cifra;

29      x/=10;
31  }

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

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

```

### Rešenje 2.77

```

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

7

#include <stdio.h>

```

## 2 Kontrola toka

---

```
9  #include <math.h>

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

15     for(i=x;i<=y;i+=korak)
17         printf("sin(%.4f)=%.4f\n", i,sin(i));
19 }

21 int main()
{
23     float a,b;
    int n;
25     float t;
    printf("Unesi dva realna broja:");
27     scanf("%f%f",&a,&b);
    printf("Unesi jedan ceo broj > 1:");
29     scanf("%u",&n);

31     if (n<=1 || a==b)
    {
33         printf("Nekorektan unos\n");
        return -1;
35     }
    if (b<a) /* u slucaju da je desni kraj intervala manji od levog */
37     { /* zamenimo im mesta */
        t=a;
39         a=b;
        b=t;
41     }

43

45     ispis(a,b,n);

47     return 0;
}
```

### Rešenje 2.78

```
/*
2  Napisati funkciju koja broji neparne cifre u zapisu datog celog broja
    . Napisati
    potom glavni program koji unosi cele brojeve dok se ne unese nula, i
    ispisuje
4  broj neparnih cifara svakog unetog broja koriscenjem napisane
    funkcije.
```



```

6  */
8  #include<stdio.h>
8  #include<stdlib.h>

10 int broj_ncifara(int x)
11 {
12     int s=0;
13     char cifra;
14     x = abs(x);

16     while(x)
17     {
18         cifra = x%10;
19         s+=(cifra%2); /* izraz cifra%2 ima vrednost 1 kada je cifra
20                        neparna,
21                        a 0 kada je cifra parna */
22         x/=10;
23     }

24     return s;
25 }

26 int main()
27 {
28     int x;
29     do
30     {
31         scanf("%d",&x);
32         printf("Broj neparnih cifara u zapisu broja %d: %d\n", x,
33               broj_ncifara(x));
34     } while(x!=0);

36     return 0;
37 }

```

### Rešenje 2.79

```

1  /*
2  Napisati funkciju koja ispituje da li je dati ceo broj prost.
3  Funkcija treba
4  da vrati 1 ako je broj prost i 0 u suprotnom. Napisati potom glavni
5  program
6  koji za uneti ceo broj n ispisuje prvih n prostih brojeva.
7  */

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

10 int prost (int x) /* 1-broj je prost, 0-broj nije prost */
11 {

```

## 2 Kontrola toka

```
13  int i;
15  if (x==2 || x==3) /* brojevi 2 i 3 su prosti */
    return 1;
17  if (x%2==0)        /* parni brojevi nisu prosti */
    return 0;
19
21  for (i=3; i<=sqrt(x);i+=2) /* trazimo delioca */
    if (x%i==0) /* ako je pronadjen, to znaci da broj nije prost */
        return 0; /* završavamo funkciju */
23
25  /* ukoliko izvršavanje funkcije dodje do poslednje naredbe return,
    to znaci da broj nije ispunio nijedan od prethodnih uslova
    (nije ni 2, ni 3, ni paran, niti ima ijednog delioca), odakle
    sledi da je prost i zbog toga vratamo 1
27  */
29  return 1;
31 }
33
35 int main()
36 {
37     int n;
38     scanf("%d",&n);
39     int i,j;
41
42     i=1; /* kandidat za prost broj */
43     j=0; /* brojac prostih brojeva */
44     while(j<n)
45     {
46         if (prost(i)) /* ako je broj prost */
47         {
48             printf("%d\n", i); /* stampamo ga i */
49             j++; /* uvecavamo brojac prostih brojeva */
50         }
51         i++; /* bilo da je i prost ili ne, uvecavamo ga za 1 i
        nastavljamo sa sledecom iteracijom */
52     }
53
54     return 0;
55 }
```

### Rešenje 2.80

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

```

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

```

### Rešenje 2.81

```

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

6   b) Napisati funkciju sve_cifre_jednake koja ispituje da li su sve
   cifre datog celog broja jednake. Funkcija treba
   da vrati 1 ako su sve cifre broja jednake i 0 u suprotnom.
8
c) Napisati potom glavni program koji na uneti ceo broj primenjuje
   napisane funkcije i ispisuje odgovarajuće poruke.
10
Na primer, za uneti broj 222, program treba da ispise:
12 Sve cifre broja su parne.
   Sve cifre broja su jednake.
14
A za uneti broj -284:
16 Sve cifre broja su parne.

```

```
Broj sadrzi razlicite cifre
18
20 */
20 #include <stdio.h>
20 #include <stdlib.h>
22
22 int sve_parne_cifre(int x) /* funkcija vraca 1 ako su sve cifre broja
    parne i 0 u suprotnom*/
24 {
24     char d;
26     x=abs(x);          /* uzimamo apsolutnu vrednost broja za slucaj da je
        broj negativan */
26     while (x>0)
28     {
28         d=x%10;          /* izdvajamo cifru broja */
30
30         if (d%2==1)      /* u slucaju da je neparna, to znaci da nisu sve
            cifre broja parne */
32             return 0;    /* vracamo 0 */
34
34         x/=10;           /* "uklanjamo" poslednju cifru broja celobrojnim
            deljenjem sa 10 */
36     }
36
36     return 1;           /* ukoliko se while petlja zavrсила, to znaci da
        uslov d%2==1 nije
38                             nijednom bio ispunjen i da su sve cifre broja
        parne; zbog toga
40                             vracamo 1
40                             */
42 }
42
44 int sve_cifre_jednake(int x) /* funkcija vraca 1 ako su sve cifre
    broja jednake i 0 u suprotnom*/
46 {
46     char d;
46     char prva_cifra;
48     x=abs(x);
48     prva_cifra = x%10; /* izdvajamo prvu cifru broja */
50     x/=10;             /* broj delimo sa 10 jer smo prvu cifru vec
        izdvojili */
52
52     while(x)
54     {
54         d = x%10;
56
56         if (d!=prva_cifra)
58             return 0;
58
58         x/=10;
60     }
```

```

62     return 1;
63 }
64 main()
65 {
66     int x;
67     int d;
68
69     printf("unesi ceo broj:");
70     scanf("%d", &x);
71
72     if (sve_parne_cifre(x))
73         printf("Sve cifre broja su parne\n");
74     else
75         printf("Broj sadrzi bar jednu neparnu cifru\n");
76
77     if (sve_cifre_jednake(x))
78         printf("Sve cifre broja su jednake\n");
79     else
80         printf("Broj sadrzi razlicite cifre \n");
81 }

```

### Rešenje 2.82

```

/*
2  Napisati funkciju koja za dva uneta neoznacena broja x i n utvrđuje
   da li je x neki stepen
   broja n. Ukoliko jeste, funkcija vraća izlozilac stepena, a u
   suprotnom vraća -1. Napisati
4  potom glavni program koji testira ovu funkciju.
   */
6
7  #include <stdio.h>
8
9  int je_stepen(unsigned x, unsigned n) /* funkcija vraća izlozilac
   stepena ukoliko broj x jeste neki stepen broja n */
10 {
11     int i=1;
12     int s=n;
13
14     while(s<x)
15     {
16         s=s*n;
17         i++;
18     }
19
20     if (s==x)
21         return i;
22
23     return -1;

```

```
24 }
26 int main()
27 {
28     unsigned x;
29     unsigned n;
30     int st;
31
32     scanf("%u%u",&x,&n);
33
34     st = je_stepen(x,n);
35
36     if (st!=-1)
37         printf("%u=%u^d\n",x,n,st);
38     else
39         printf("%u nije stepen broja %u\n",x,n);
40
41     return 0;
42 }
```

### Rešenje 2.83

```
/*
2
3  Napisati funkciju
4
5  double e_na_x(double x, double eps)
6
7  koja racuna vrednost e^x kao parcijalnu sumu reda
8  suma(x^n/n!), gde indeks n ide od
9  od 0 do beskonacno, pri cemu se sumiranje vrši dok
10 je razlika sabiraka u redu po apsolutnoj vrednosti
11 manja od eps. Napisati potom program koji omogućuje
12 korisniku da unese jedan realan broj x i ispisuje
13 vrednost e^x.
14
15 */
16
17 #include<stdio.h>
18 #include<math.h>
19
20 double e_na_x(double x, double eps)
21 {
22     double s=1;
23     double clan=1;
24     int n=1;
25
26     /*
27     parcijalnu sumu formiramo tako sto u svakoj iteraciji petlje
28     promenljivoj s dodamo jedan sabirak sume oblika (x^n)/n! koji
29     cuvamo u promenljivoj clan
30     */
31 }
```

```

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

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

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

42  do
43  {
44      clan = (clan*x)/n;
45      s += clan;
46      n++;
47  } while(fabs(clan)>eps);

48      return s;
50  }

52  int main()
53  {
54      double x,eps;
55      printf("x=");
56      scanf("%lf", &x);
57      printf("eps=");
58      scanf("%lf", &eps);

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

```

### Rešenje 2.84

```

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

```

## 2 Kontrola toka

---

```
10 - broj 372 nije srecan, zbir njegovih cifara je 12, zbir cifara broja
    12 je 3
12 Napisati funkciju koja vraca 1 ako je broj srecan, a 0 u suprotnom.
14 Napisati program koji omogucava korisniku da unese prirodan broj,
    poziva funkciju
    i ispisuje da li je dati broj srecan. Potom traziti od korisnika da
    unese prirodan
16 broj n i ispisati sve srecne brojeve od 1 do n.
    */
18
19 #include<stdio.h>
20
21 int zbir_cifara(int x)
22 {
23     int s=0;
24     char cifra;
25     while(x)
26     {
27         cifra = x%10;
28         s+=cifra;
29         x/=10;
30     }
31     return s;
32 }
33
34 int srecan(int x)
35 {
36     int s; /* promenljiva s sadrzi sumu cifara */
37
38     do
39     {
40         s=zbir_cifara(x);
41         x=s; /* kada izracunamo sumu cifara, dodeljujemo je promenljivoj
42             x jer iz te promenljive izdvajamo cifre u funkciji zbir_cifara
43             */
44     } while(x>=10);
45
46     return (x==1);
47 }
48
49 int main()
50 {
51     unsigned n;
52     int i;
53     printf("Unesi jedan neoznaceni broj:");
54     scanf("%u",&n);
55
56     for(i=1;i<=n;i++)
57         if (srecan(i))
```



```
        printf("%d je srecan\n", i);
58     return 0;
60 }
```

### Rešenje 2.85

```
/*
2  . a) Napisati funkciju
4  int konverzija (int c)
6  koja prebacuje veliko slovo u ekvivalentno malo i obrnuto.
8  b) Napisati program koji omogućava korisniku da unese niz karaktera
   sa tastature, a potom ispisuje uneseni niz konvertovanih karaktera.
10 Na primer, za uneti tekst "Kolokvijum iz Progi je 1.12." program
   treba da ispise "kOLOVKIJUM IZ pROG1 JE 1.12."
12
14 */
#include <stdio.h>
16 int konverzija(int c)
{
18     /* ključna rec return vraća povratnu vrednost funkcije (ako je ima)
        */
        /* i završava izvršavanje funkcije */
20     if (c>='A' && c<='Z')
22         return c+'a'-'A';
24     if (c>='a' && c<='z')
26         return c-'a'+'A';
28     return c;
}
30 int main()
{
32     int c;
34     while((c=getchar())!=EOF) /* korisnik unosi karakter po karakter
        do konstante EOF */
        putchar(konverzija(c)); /* funkcija putchar ispisuje jedan
36                                karakter na standardni izlaz */
38     return 0;
}
```

Rešenje [2.104](#)

Rešenje [2.104](#)

Rešenje [2.104](#)

Rešenje [2.104](#)

Rešenje [2.104](#)

Rešenje [2.104](#)

Rešenje [2.104](#)

Rešenje [2.104](#)

Rešenje [2.104](#)

Rešenje [2.104](#)

Rešenje [2.104](#)

Rešenje [2.104](#)

Rešenje [2.104](#)

Rešenje [2.104](#)

Rešenje [2.104](#)

Rešenje [2.104](#)

Rešenje [2.104](#)

Rešenje [2.104](#)

Rešenje [2.104](#)

# 3

## Predstavljanje podataka

### 3.1 Nizovi

**Zadatak 3.1** Tekst

[Rešenje [3.1](#)]

**Zadatak 3.2** Tekst

[Rešenje [3.2](#)]

**Zadatak 3.3** Tekst

[Rešenje [3.3](#)]

**Zadatak 3.4** Tekst

[Rešenje [3.4](#)]

**Zadatak 3.5** Tekst

[Rešenje [3.5](#)]

**Zadatak 3.6** Tekst

[Rešenje [3.6](#)]

**Zadatak 3.7** Tekst

[Rešenje [3.7](#)]

### 3 Predstavljanje podataka

---

#### Zadatak 3.8 Tekst

[Rešenje 3.8]

#### Zadatak 3.9 Tekst

[Rešenje 3.9]

#### Zadatak 3.10 Tekst

[Rešenje 3.10]

**Zadatak 3.11** Sa standardnog ulaza se unosi dimenzija niza (broj manji od 100), a zatim i njegovi elementi. Napisati program koji kvadrira sve negativne elemente niza i ispisuje rezultujući niz.

##### Primer 1

```
INTERAKCIJA SA PROGRAMOM:
Unesite broj elemenata niza: 6
Unesite elemente niza:
12.34 -6 1 8 32.4 -16
12.34 36 1 8 32.4 256
```

##### Primer 2

```
INTERAKCIJA SA PROGRAMOM:
Unesite broj elemenata niza: 9
Unesite elemente niza:
-8.25 6 17 2 -1.5 1 -7 2.65 -125.2
68.0625 6 17 2 2.25 1 49 2.65 15675.04
```

##### Primer 3

```
INTERAKCIJA SA PROGRAMOM:
Unesite broj elemenata niza: 4
Unesite elemente niza:
9.53 5 1 4.89
9.53 5 1 4.89
```

[Rešenje 3.26]

**Zadatak 3.12** Sa standardnog ulaza se učitava dimenzija niza (broj manji od 100), elemente niza i jedan ceo broj  $k$ . Napisati program koji štampa indekse elemenata koji su deljivi sa  $k$ .

##### Primer 1

```
INTERAKCIJA SA PROGRAMOM:
Unesite dimenziju niza: 4
Unesite elemente niza: 10 14 86 20
Unesite broj k: 5
0 3
```

##### Primer 2

```
INTERAKCIJA SA PROGRAMOM:
Unesite dimenziju niza: 4
Unesite elemente niza: 6 14 8 9
Unesite broj k: 5
U nizu nema elemenata koji su deljivi brojem 5!
```

*Primer 3*

```

|| INTERAKCIJA SA PROGRAMOM:
|| Unesite dimenziju niza: 6
|| Unesite elemente niza: 8 9 11 -4 8 11
|| Unesite broj k: 2
|| 0 3 4

```

[Rešenje 3.26]

**Zadatak 3.13** Napisati program koji sa standardnog ulaza učitava dimenziju niza (broj manji od 100) i elemente niza, a zatim štampa niz u kojem su najveći i najmanji element niza razmenili mesta.

*Primer 1*

```

|| INTERAKCIJA SA PROGRAMOM:
|| Unesite dimenziju niza: 5
|| Unesite elemente niza: 8 -2 11 19 4
|| 8 19 11 -2 4

```

*Primer 2*

```

|| INTERAKCIJA SA PROGRAMOM:
|| Unesite dimenziju niza: 10
|| Unesite elemente niza:
|| 46 -2 51 8 -5 66 2 8 3 14
|| 46 -2 51 8 66 -5 2 8 3 14

```

*Primer 3*

```

|| INTERAKCIJA SA PROGRAMOM:
|| Unesite dimenziju niza: 145
|| Greska: pogresan unos!

```

[Rešenje 3.26]

**Zadatak 3.14** Napisati program koji učitava karaktere sa ulaza (najviše njih 100) sve do pojave karaktera \*, a zatim ih ispisuje u redosledu suprotnom od redosleda čitanja.

*Primer 1*

```

|| INTERAKCIJA SA PROGRAMOM:
|| Unesite karakter: a
|| Unesite karakter: 8
|| Unesite karakter: 5
|| Unesite karakter: Y
|| Unesite karakter: I
|| Unesite karakter: o
|| Unesite karakter: ?
|| Unesite karakter: *
|| ? o I Y 5 8 a

```

*Primer 2*

```

|| INTERAKCIJA SA PROGRAMOM:
|| Unesite karakter: g
|| Unesite karakter: g
|| Unesite karakter: 2
|| Unesite karakter: 2
|| Unesite karakter: )
|| Unesite karakter: )
|| Unesite karakter: *
|| ) ) 2 2 g g

```

### 3 Predstavljanje podataka

---

#### Primer 3

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesite karakter: U  
|| Unesite karakter: 4  
|| Unesite karakter: a  
|| Unesite karakter: u  
|| Unesite karakter: *  
|| u a 4 U
```

[Rešenje 3.26]

**Zadatak 3.15** Napisati program koji za dva cela broja  $x$  i  $y$  koja se učitavaju sa standardnog ulaza proverava da li se zapisuju pomoću istih cifara. Napomena: iskoristiti niz za čuvanje broja pojavljivanja svake od cifara.

#### Primer 1

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesite dva broja: 251 125  
|| Brojevi se zapisuju istim ciframa!
```

#### Primer 2

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesite dva broja: 8898 9988  
|| Brojevi se ne zapisuju istim ciframa!
```

#### Primer 3

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesite dva broja: -7391 1397  
|| Brojevi se zapisuju istim ciframa!
```

[Rešenje 3.26]

**Zadatak 3.16** Sa standardnog ulaza se učitava dimenzija niza (broj manji od 100), zatim i elementi dvaju nizova  $a$  i  $b$ . Napisati program koji formira i ispisuje niz  $c$  čiju prvu polovinu čine elementi niza  $b$ , a drugu polovinu elementi niza  $a$ .

#### Primer 1

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesite broj n: 3  
|| Unesite elemente niza a: 4 -8 32  
|| Unesite elemente niza b: 5 2 11  
|| 5 2 11 4 -8 32
```

#### Primer 2

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesite broj n: 4  
|| Unesite elemente niza a: 1 0 -1 0  
|| Unesite elemente niza b: 5 5 5 3  
|| 5 5 5 3 1 0 -1 0
```

#### Primer 3

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesite dimenziju niza: 145  
|| Greska: pogresan unos!
```

[Rešenje 3.26]

**Zadatak 3.17** Sa standardnog ulaza se unosi dimenzija niza  $a$  (broj manji od 100), a zatim i njegovi elementi. Napisati program koji od datog niza formira niz  $b$  u koji ulaze elementi niza  $a$  koji se pojavljuju tačno 3 puta.

*Primer 1*

```
INTERAKCIJA SA PROGRAMOM:
Unesite broj elemenata niza: 8
Unesite elemente niza a:
4 11 4 6 8 4 6 6
Elementi niza b: 4 6
```

*Primer 2*

```
INTERAKCIJA SA PROGRAMOM:
Unesite broj elemenata niza: 13
Unesite elemente niza a:
-8 26 7 2 1 1 7 2 2 2 7 5 1
Elementi niza b: 7 1
```

*Primer 3*

```
INTERAKCIJA SA PROGRAMOM:
Unesite broj elemenata niza: 2
Unesite elemente niza a:
9 5
Elementi niza b:
```

[Rešenje 3.26]

**Zadatak 3.18** Sa standardnog ulaza se, redom, učitavaju dimenzija i elementi dva niza  $a$  i  $b$ . Napisati program koji određuje njihovu uniju, presek i razliku (redosled prikaza elemenata nije bitan). Pretpostaviti da će nizovi imati manje od 100 elemenata.

*Primer 1*

```
INTERAKCIJA SA PROGRAMOM:
Unesite broj elemenata niza a: 5
Unesite elemente niza a: 2 8 1 5 2
Unesite broj elemenata niza b: 3
Unesite elemente niza b: 5 7 8
Unija: 2 8 1 5 2 5 7 8
Presek: 5
Razlika: 2 1 2
```

*Primer 2*

```
INTERAKCIJA SA PROGRAMOM:
Unesite broj elemenata niza a: 3
Unesite elemente niza a: 11 4 4
Unesite broj elemenata niza b: 2
Unesite elemente niza b: 18 9
Unija: 11 4 4 18 9
Presek:
Razlika: 11 4 4
```

*Primer 3*

```
INTERAKCIJA SA PROGRAMOM:
Unesite broj elemenata niza a: 6
Unesite elemente niza a: 12 7 9 12 5 1
Unesite broj elemenata niza b: 4
Unesite elemente niza b: 1 12 22 12
Unija: 12 7 9 12 5 1 1 12 22 12
Presek: 12 12 1
Razlika: 7 9 5
```

[Rešenje 3.26]

### 3 Predstavljanje podataka

---

**Zadatak 3.19** Napisati program koji učitava dimenziju niza (broj manji od 100) i elemente niza, a zatim formira i ispisuje niz koji se dobija izbacivanjem svih neparnih elemenata niza. Zadatak rešiti na dva načina: korišćenjem pomoćnog niza i transformacijom polaznog niza.

#### Primer 1

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesite broj elemenata niza: 4  
|| Unesite elemente niza: 8 9 15 12  
|| 8 12
```

#### Primer 2

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesite broj elemenata niza: 6  
|| Unesite elemente niza: 21 5 3 22 19 188  
|| 22 188
```

#### Primer 3

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesite broj elemenata niza: 4  
|| Unesite elemente niza: 133 129 121 101
```

#### Primer 4

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesite broj elemenata niza: 8  
|| Unesite elemente niza: 15 -22 -23 13 18 46 14 -31  
|| -22 18 46 14
```

[Rešenje 3.26]

**Zadatak 3.20** Napisati program koji učitava dimenziju niza (broj manji od 100) i elemente niza, a zatim formira i ispisuje niz koji se dobija izbacivanjem svih elemenata koji su prosti brojevi. Zadatak rešiti na dva načina: korišćenjem pomoćnog niza i transformacijom polaznog niza. Napomena: brojeve -1 i 1 smatrati prostim.

#### Primer 1

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesite broj elemenata niza: 5  
|| Unesite elemente niza: 11 5 6 48 8  
|| 6 48 8
```

#### Primer 2

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesite broj elemenata niza: 4  
|| Unesite elemente niza: 11 5 19 21  
|| 21
```

#### Primer 3

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesite broj elemenata niza: 5  
|| Unesite elemente niza: 12 18 9 31 7  
|| 12 18 9
```



*Primer 4*

```

|| INTERAKCIJA SA PROGRAMOM:
|| Unesite broj elemenata niza: 3
|| Unesite elemente niza: -31 11 -19

```

*Primer 5*

```

|| INTERAKCIJA SA PROGRAMOM:
|| Unesite broj elemenata niza: 5
|| Unesite elemente niza: -2 15 -11 8 7
|| 15 8

```

[Rešenje 3.26]

**Zadatak 3.21** Napisati funkciju *int prebrojavanje(int a[], int n)* koja izračunava broj elemenata niza celih brojeva *a* dužine *n* koji su manji od poslednjeg elementa niza. Napisati i program koji testira rad funkcije. Pretpostaviti da dužina niza neće biti veća od 100.

*Primer 1*

```

|| INTERAKCIJA SA PROGRAMOM:
|| Unesite broj elemenata niza: 4
|| Unesite elemente niza: 11 2 4 9
|| 2

```

*Primer 2*

```

|| INTERAKCIJA SA PROGRAMOM:
|| Unesite broj elemenata niza: 7
|| Unesite elemente niza: 7 2 1 14 65 2 8
|| 4

```

*Primer 3*

```

|| INTERAKCIJA SA PROGRAMOM:
|| Unesite broj elemenata niza: 5
|| Unesite elemente niza: 25 18 29 30 14
|| 0

```

[Rešenje 3.26]

**Zadatak 3.22** Napisati funkciju *int prebrojavanje(int a[], int n)* koja izračunava broj parnih elemenata niza celih brojeva *a* dužine *n* koji prethode maksimalnom elementu niza. Napisati i program koji testira rad funkcije. Pretpostaviti da dužina niza neće biti veća od 100.

*Primer 1*

```

|| INTERAKCIJA SA PROGRAMOM:
|| Unesite broj elemenata niza: 4
|| Unesite elemente niza: 11 2 4 9
|| 0

```

*Primer 2*

```

|| INTERAKCIJA SA PROGRAMOM:
|| Unesite broj elemenata niza: 7
|| Unesite elemente niza: 7 2 1 14 65 2 8
|| 2

```

*Primer 3*

```

|| INTERAKCIJA SA PROGRAMOM:
|| Unesite broj elemenata niza: 5
|| Unesite elemente niza: 25 18 29 30 14
|| 1

```

### 3 Predstavljanje podataka

---

[Rešenje 3.26]

**Zadatak 3.23** Napisati funkciju *int prebrojavanje\_cifre(char s[], int n)* koja izračunava broj cifara u nizu karaktera *a* dužine *n*. Napisati i program koji testira rad funkcije. Pretpostaviti da dužina niza neće biti veća od 100.

#### Primer 1

```
INTERAKCIJA SA PROGRAMOM:
Unesite broj elemenata niza: 5
Unesite elemente niza:
4
+
4
u
8
Broj cifara je: 2
```

#### Primer 2

```
INTERAKCIJA SA PROGRAMOM:
Unesite broj elemenata niza: 7
Unesite elemente niza:
J
M
a
5
5
-
2
Broj cifara je: 3
```

#### Primer 3

```
INTERAKCIJA SA PROGRAMOM:
Unesite broj elemenata niza: 3
Unesite elemente niza:
e
k
F
Broj cifara je: 0
```

[Rešenje 3.26]

**Zadatak 3.24** Napisati funkciju *int zbir(int a[], int n, int i, int j)* koja računa zbir elemenata niza celih brojeva *a* dužine *n* od pozicije *i* do pozicije *j*. Napisati i program koji testira rad funkcije. Pretpostaviti da dužina niza neće biti veća od 100.

#### Primer 1

```
INTERAKCIJA SA PROGRAMOM:
Unesite broj elemenata niza: 5
Unesite elemente niza: 11 5 6 48 8
Unesite vrednosti za i i j: 0 2
Zbir je: 22
```

#### Primer 2

```
INTERAKCIJA SA PROGRAMOM:
Unesite broj elemenata niza: 3
Unesite elemente niza: -2 8 1
Unesite vrednosti za i i j: 8 12
Greska: nekorektne vrednosti granica!
```

#### Primer 3

```
INTERAKCIJA SA PROGRAMOM:
Unesite broj elemenata niza: 7
Unesite elemente niza: -2 5 9 11 6 -3 -4
Unesite vrednosti za i i j: 2 5
Zbir: 23
```

[Rešenje 3.26]

**Zadatak 3.25** Napisati funkciju *float zbir\_pozitivnih(float a[], int n, int k)* koja izračunava zbir prvih  $k$  pozitivnih elemenata realnog niza  $a$  dužine  $n$ . Napisati i program koji testira rad funkcije. Pretpostaviti da dužina niza neće biti veća od 100.

*Primer 1*

```
INTERAKCIJA SA PROGRAMOM:
Unesite broj elemenata niza: 8
Unesite elemente niza:
2.34 1 -12.7 5.2 -8 -6.2 7 14.2
Unesite vrednost za k: 3
Zbir je: 8.54
```

*Primer 2*

```
INTERAKCIJA SA PROGRAMOM:
Unesite broj elemenata niza: 3
Unesite elemente niza:
-6.598 -8.14 -15
Unesite vrednost za k: 4
Zbir je: 0.00
```

*Primer 3*

```
INTERAKCIJA SA PROGRAMOM:
Unesite broj elemenata niza: 7
Unesite elemente niza:
-35.11 5.29 -1.98 12.1 12.2 -3.33 -4.17
Unesite vrednost za k: 15
Zbir: 29.59
```

[Rešenje 3.26]

**Zadatak 3.26** Napisati funkciju *void kvadriranje(float a[], int n)* koja kvadrira elemente realnog niza  $a$  dužine  $n$  koji se nalaze na parnim pozicijama. Napisati i program koji testira rad funkcije. Pretpostaviti da dužina niza neće biti veća od 100.

*Primer 1*

```
INTERAKCIJA SA PROGRAMOM:
Unesite broj elemenata niza: 8
Unesite elemente niza:
2.34 1 -12.7 5.2 -8 -6.2 7 14.2
5.4756 1 161.29 5.2 64 -6.2 49 14.2
```

*Primer 2*

```
INTERAKCIJA SA PROGRAMOM:
Unesite broj elemenata niza: 3
Unesite elemente niza:
-6 -8.14 -15
36 -8.14 225
```

*Primer 3*

```
INTERAKCIJA SA PROGRAMOM:
Unesite broj elemenata niza: 1
Unesite elemente niza:
-35.11
1232.71
```

[Rešenje 3.26]

## 3.2 Pokazivači

**Zadatak 3.27** Tekst

[Rešenje [3.27](#)]

**Zadatak 3.28** Tekst

[Rešenje [3.28](#)]

**Zadatak 3.29** Tekst

[Rešenje [3.29](#)]

**Zadatak 3.30** Tekst

[Rešenje [3.30](#)]

**Zadatak 3.31** Tekst

[Rešenje [3.31](#)]

**Zadatak 3.32** Tekst

[Rešenje [3.32](#)]

**Zadatak 3.33** Napisati program koji ispisuje zbir numeričkih argumenata komandne linije. Napomena: može se koristiti funkcija *atoi*.

*Primer 1*

```
|| POKRETANJE: ./a.out 5 mkp 9 -2 11 a 4 2
|| INTERAKCIJA SA PROGRAMOM:
||   Zbir numerickih argumenata: 29
```

*Primer 2*

```
|| POKRETANJE: ./a.out ab u f hj
|| INTERAKCIJA SA PROGRAMOM:
||   Zbir numerickih argumenata: 0
```

*Primer 3*

```
|| POKRETANJE: ./a.out 33 1 p 44
|| INTERAKCIJA SA PROGRAMOM:
||   Zbir numerickih argumenata: 78
```

*Primer 4*

```
|| POKRETANJE: ./a.out
|| INTERAKCIJA SA PROGRAMOM:
||   Zbir numerickih argumenata: 0
```

[Rešenje [3.41](#)]

**Zadatak 3.34** Napisati program koji ispisuje argumente komandne linije koji počinju slovom z.

*Primer 1*

```
|| POKRETANJE: ./a.out zima jabuka zvezda Zrak
|| INTERAKCIJA SA PROGRAMOM:
|| zima zvezda
```

*Primer 2*

```
|| POKRETANJE: ./a.out bundeva pomorandza
|| INTERAKCIJA SA PROGRAMOM:
```

*Primer 3*

```
|| POKRETANJE: ./a.out sanke zapad zujanje
|| INTERAKCIJA SA PROGRAMOM:
|| zapad zujanje
```

*Primer 4*

```
|| POKRETANJE: ./a.out
|| INTERAKCIJA SA PROGRAMOM:
```

[Rešenje 3.41]

**Zadatak 3.35** Napisati program koji ispisuje broj argumenata komandne linije koji sadrže slovo z.

*Primer 1*

```
|| POKRETANJE: ./a.out zvezda grozd jesen kisa
|| INTERAKCIJA SA PROGRAMOM:
|| 2
```

*Primer 2*

```
|| POKRETANJE: ./a.out AZBUKA deda mraz
|| INTERAKCIJA SA PROGRAMOM:
|| 2
```

*Primer 3*

```
|| POKRETANJE: ./a.out japan caj
|| INTERAKCIJA SA PROGRAMOM:
|| 0
```

*Primer 4*

```
|| POKRETANJE: ./a.out
|| INTERAKCIJA SA PROGRAMOM:
|| 0
```

[Rešenje 3.41]

**Zadatak 3.36** Napisati program koji na osnovu broja  $n$  koji se zadaje kao argument komandne linije ispisuje cele brojeve iz intervala  $[-n, n]$ .

### 3 Predstavljanje podataka

---

#### Primer 1

```
|| POKRETANJE: ./a.out 2
|| INTERAKCIJA SA PROGRAMOM:
|| -2 -1 0 1 2
```

#### Primer 2

```
|| POKRETANJE: ./a.out 4
|| INTERAKCIJA SA PROGRAMOM:
|| -4 -3 -2 -1 0 1 2 3 4
```

#### Primer 3

```
|| POKRETANJE: ./a.out 0
|| INTERAKCIJA SA PROGRAMOM:
|| 0
```

#### Primer 4

```
|| POKRETANJE: ./a.out
|| INTERAKCIJA SA PROGRAMOM:
|| Greska: nedostaje argument komandne linije!
```

[Rešenje 3.41]

**Zadatak 3.37** Napisati program koji proverava da li se među zadatim argumentima komandne linije nalaze barem dva ista.

#### Primer 1

```
|| POKRETANJE: ./a.out pec zima deda mraz pec
|| INTERAKCIJA SA PROGRAMOM:
|| Medju argumentima ima istih.
```

#### Primer 2

```
|| POKRETANJE: ./a.out xyz abc abc abc efgh
|| INTERAKCIJA SA PROGRAMOM:
|| Medju argumentima ima istih.
```

#### Primer 3

```
|| POKRETANJE: ./a.out 11 15 abc 888
|| INTERAKCIJA SA PROGRAMOM:
|| Medju argumentima nema istih.
```

#### Primer 4

```
|| POKRETANJE: ./a.out
|| INTERAKCIJA SA PROGRAMOM:
|| Medju argumentima nema istih.
```

[Rešenje 3.41]

**Zadatak 3.38** Napisati funkciju *void modifikacija(char\* s, char\* t, int\* br\_modifikacija)* koja na osnovu niske *s* formira nisku *t* tako što svako malo slovo zamenjuje velikim. Broj izvršenih modifikacija se čuva u okviru argumenta *br\_modifikacija*. Pretpostaviti da niska *s* neće biti duža od 20 karaktera. Napisati i program koji testira rad napisane funkcije.

## Primer 1

```

|| INTERAKCIJA SA PROGRAMOM:
|| Unesite nisku: 123abc789XY
|| Modifikovana niska je: 123ABC789XY
|| Broj modifikacija je: 3

```

## Primer 2

```

|| INTERAKCIJA SA PROGRAMOM:
|| Unesite nisku: zimA
|| Modifikovana niska je: ZIMA
|| Broj modifikacija je: 3

```

## Primer 3

```

|| INTERAKCIJA SA PROGRAMOM:
|| Unesite nisku: SNEG
|| Modifikovana niska je: SNEG
|| Broj modifikacija je: 0

```

[Rešenje 3.41]

**Zadatak 3.39** Napisati funkciju *void interpunkcija(int\* br\_tacaka, int\* br\_zareza)* koja za tekst koji se unosi sa standardnog ulaza sve do kraja ulaza prebrojava broj tačaka i zareza. Napisati zatim program koji testira napisanu funkciju.

## Primer 1

```

|| INTERAKCIJA SA PROGRAMOM:
|| Unesite tekst:
|| a.b.c.d
|| a,b,,c,d,e
|| Broj tacaka: 3
|| Broj zareza: 5

```

## Primer 2

```

|| INTERAKCIJA SA PROGRAMOM:
|| Unesite tekst:
|| .....789.....
|| Broj tacaka: 10
|| Broj zareza: 0

```

## Primer 2

```

|| INTERAKCIJA SA PROGRAMOM:
|| Unesite tekst:
|| sunce
|| Broj tacaka: 0
|| Broj zareza: 0

```

[Rešenje 3.41]

**Zadatak 3.40** Napisati funkciju *void par\_nepar(int a[], int n, int parni[], int\* pn, int neparni[], int\* nn)* koja razbija niz *a* na niz parnih i niz neparnih brojeva. Pokazivači *pn* i *nn* redom treba da sadrže broj elemenata niza parnih tj. niza neparnih elemenata. Pretpostaviti da dužina niza *a* neće biti veća od 50. Napisati program koji testira napisanu funkciju.

#### Primer 1

```
INTERAKCIJA SA PROGRAMOM:
Unesite broj elemenata niza: 8
Unesite elemente niza:
1 8 9 -7 -16 24 77 4
Niz parnih brojeva: 8 -16 24 4
Niz neparnih brojeva: 1 9 -7 77
```

#### Primer 2

```
INTERAKCIJA SA PROGRAMOM:
Unesite broj elemenata niza: 5
Unesite elemente niza:
2 4 6 8 -11
Niz parnih brojeva: 2 4 6 8
Niz neparnih brojeva: -11
```

#### Primer 3

```
INTERAKCIJA SA PROGRAMOM:
Unesite broj elemenata niza: 2
Unesite elemente niza:
-15 15
Niz parnih brojeva:
Niz neparnih brojeva: -15 15
```

[Rešenje 3.41]

**Zadatak 3.41** Napisati funkciju `void min_max(float a[], int n, float *min, float *max)` koja izračunava minimalni i maksimalni element niza `a` dužine `n`. Napisati zatim i program koji učitava niz realnih brojeva maksimalne dužine 50 i ispisuje vrednosti minimuma i maksimuma na tri decimale.

#### Primer 1

```
INTERAKCIJA SA PROGRAMOM:
Unesite broj elemenata niza: 5
Unesite elemente niza:
24.16 -32.11 999.25 14.25 11
Minimum: -32.110
Maksimum: 999.250
```

#### Primer 2

```
INTERAKCIJA SA PROGRAMOM:
Unesite broj elemenata niza: 4
Unesite elemente niza:
-5.126 -18.29 44 29.268
Minimum: -18.290
Maksimum: 44.000
```

#### Primer 3

```
INTERAKCIJA SA PROGRAMOM:
Unesite broj elemenata niza: 1
Unesite elemente niza:
4.16
Minimum: 4.160
Maksimum: 4.160
```

[Rešenje 3.41]

## 3.3 Niske

**Zadatak 3.42** Tekst



[Rešenje [3.42](#)]

**Zadatak 3.43** Tekst

[Rešenje [3.43](#)]

**Zadatak 3.44** Tekst

[Rešenje [3.44](#)]

**Zadatak 3.45** Tekst

[Rešenje [3.45](#)]

**Zadatak 3.46** Tekst

[Rešenje [3.46](#)]

**Zadatak 3.47** Tekst

[Rešenje [3.47](#)]

**Zadatak 3.48** Tekst

[Rešenje [3.48](#)]

**Zadatak 3.49** Tekst

[Rešenje [3.49](#)]

**Zadatak 3.50**

- a) Napisati funkciju *int samoglasnik(char c)* koja proverava da li je zadati karakter samoglasnik. Funkcija treba da vrati vrednost 1 ako karakter *c* jeste samoglasnik, odnosno 0 ako nije.
- b) Napisati funkciju *int samoglasnik\_na\_kraju(char s[])* koja proverava da li se niska *s* završava samoglasnikom (koristiti funkciju iz tačke a)).
- c) Napisati program koji učitava nisku maksimalne dužine 20 karaktera i ispisuje da li završava samoglasnikom ili ne.

### 3 Predstavljanje podataka

---

#### Primer 1

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesite nisku: abcde  
|| Niska se završava samoglasnikom!
```

#### Primer 2

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesite nisku: AaBb+cCdD  
|| Niska se ne završava samoglasnikom!
```

#### Primer 3

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesite nisku: pRoGraMiranjE  
|| Niska se završava samoglasnikom!
```

[Rešenje 3.58]

**Zadatak 3.51** Napisati funkciju *void kopiraj\_n(char t[], char s[], int n)* koja kopira najviše  $n$  karaktera niske  $s$  u nisku  $t$ . Napisati i program koji učitava nisku maksimalne dužine 20 karaktera i jedan ceo broj i testira rad napisane funkcije.

#### Primer 1

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesite nisku: abcdef  
|| Unesite broj n: 3  
|| Rezultujuća niska: abc
```

#### Primer 2

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesite nisku: programiranje  
|| Unesite broj n: 5  
|| Rezultujuća niska: progr
```

#### Primer 3

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesite nisku: abc  
|| Unesite broj n: 15  
|| Rezultujuća niska: abc
```

[Rešenje 3.58]

**Zadatak 3.52** Napisati funkciju *void dupliranje(char t[], char s[])* koja na osnovu niske  $s$  formira nisku  $t$  tako što duplira svaki karakter niske  $s$ . Napisati i program koji učitava nisku maksimalne dužine 20 karaktera i testira rad napisane funkcije.

#### Primer 1

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesite nisku: zima  
|| zziimmaa
```

#### Primer 2

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesite nisku: A+B+C  
|| AA++BB++CC
```

*Primer 3*

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite nisku: C
|| CC
```

[Rešenje 3.58]

**Zadatak 3.53** Napisati funkciju *int heksa\_broj(char s[])* koja proverava da li je niskom *s* zadat korektan heksadekadni broj. Heksadekadni broj je korektno zadat ako počinje prefiksom *0x* ili *0X* i ako sadrži samo cifre i mala ili velika slova *A, B, C, D, E* i *F*. Funkcija treba da vrati vrednost 1 ako je niska korektan heksadekadni broj, odnosno 0 ako nije. Napisati i program koji učitava nisku maksimalne dužine 7 karaktera i ispisuje rezultat rada funkcije.

*Primer 1*

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite nisku: 0x12EF
|| Korektan heksadekadni broj!
```

*Primer 2*

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite nisku: 0X22af
|| Korektan heksadekadni broj!
```

*Primer 3*

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite nisku: 0xErA9
|| Nekorektan heksadekadni broj!
```

[Rešenje 3.58]

**Zadatak 3.54** Napisati funkciju *int heksa\_broj(char s[])* koja izračunava dekadnu vrednost heksadekadnog broja zadatog niskom *s*. Napisati i program koji učitava nisku maksimalne dužine 7 karaktera i ispisuje rezultat rada funkcije. Pretpostaviti da je uneta niska korektan heksadekadni broj.

*Primer 1*

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite nisku: 0x2A34
|| 10804
```

*Primer 2*

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite nisku: 0xfF2
|| 4082
```

*Primer 3*

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite nisku: 0xE1A9
|| 57769
```

[Rešenje 3.58]

### 3 Predstavljanje podataka

---

**Zadatak 3.55** Napisati funkciju *int podniska(char s[], char t[])* koja proverava da li je niska *t* podniska niske *s*. Napisati i program koji učitava dve niske maksimalne dužine 10 karaktera i testira rad napisane funkcije.

#### Primer 1

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesite nisku s: abcde  
|| Unesite nisku t: bcd  
|| t je podniska niske s!
```

#### Primer 2

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesite nisku s: abcde  
|| Unesite nisku t: bCd  
|| t nije podniska niske s!
```

#### Primer 3

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesite nisku s: abcde  
|| Unesite nisku t: def  
|| t nije podniska niske s!
```

[Rešenje 3.58]

**Zadatak 3.56** Napisati funkciju *void modifikacija(char \* s)* koja modifikuje nisku *s* tako što svaki drugi karakter zameni zvezdicom. Pretpostaviti da niska *s* neće biti duža od 20 karaktera. Napisati i program koji testira rad napisane funkcije.

#### Primer 1

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesite nisku: 123abc789XY  
|| Modifikovana niska je: 1*3*b*7*9*Y
```

#### Primer 2

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesite nisku: zima  
|| Modifikovana niska je: z*m*
```

#### Primer 3

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesite nisku: SNEG  
|| Modifikovana niska je: S*E*
```

[Rešenje 3.58]

**Zadatak 3.57** Napisati funkciju *int strspn\_klon(char \* t, char \* s)* koja izračunava dužinu prefiksa niske *t* sastavljenog od karaktera niske *s*. Napisati zatim i program koji učitava dve niske maksimalne dužine 20 karaktera i ispisuje rezultat poziva napisane funkcije.

*Primer 1*

```

|| INTERAKCIJA SA PROGRAMOM:
|| Unesite nisku t: programiranje
|| Unesite nisku s: opqr
|| 3

```

*Primer 2*

```

|| INTERAKCIJA SA PROGRAMOM:
|| Unesite nisku t: aaioo124
|| Unesite nisku s: aeioo
|| 6

```

*Primer 3*

```

|| INTERAKCIJA SA PROGRAMOM:
|| Unesite nisku t: 5296abc
|| Unesite nisku s: 0123456789
|| 4

```

[Rešenje 3.58]

**Zadatak 3.58** Napisati implementaciju funkcije `char * strchrklon(char * s, char c)` koja vraća pokazivač na prvo pojavljivanje karaktera `c` u niski `s` ili NULL ukoliko se karakter `c` ne pojavljuje u niski `s`. Učitati potom jednu nisku maksimalne dužine 20 karaktera i jedan dodatni karakter i testirati rad napisane funkcije.

*Primer 1*

```

|| INTERAKCIJA SA PROGRAMOM:
|| Unesite nisku s: programiranje
|| Unesite karakter c: a
|| Karakter se nalazi u niski!

```

*Primer 2*

```

|| INTERAKCIJA SA PROGRAMOM:
|| Unesite nisku s: 123456789
|| Unesite karakter c: y
|| Karakter se ne nalazi u niski!

```

[Rešenje 3.58]

## 3.4 Višedimenzioni nizovi

## 3.5 Strukture

**Zadatak 3.59** Tekst

[Rešenje 3.59]

**Zadatak 3.60** Tekst

[Rešenje 3.60]

**Zadatak 3.61** Tekst

[Rešenje 3.61]

**Zadatak 3.62** Tekst

[Rešenje 3.62]

**Zadatak 3.63** Tekst

[Rešenje 3.63]

**Zadatak 3.64** Definisati strukturu kojom se predstavlja kompleksan broj. Napisati funkcije koje izračunavaju zbir, razliku, proizvod i količnik dva kompleksna broja, a zati i program koji učitava dva kompleksna broja i ispisuje vrednost zbira, razlike, proizvoda i količnika.

*Primer 1*

```
INTERAKCIJA SA PROGRAMOM:
Unesite realni i imaginarni deo prvog broja:  1 2
Unesite realni i imaginarni deo drugog broja: -2 3
Zbir: -1.00+5.00*i
Razlika: 3.00-1.00*i
Proizvod: -8.00-1.00*i
Kolicnik: 0.31-0.54*i
```

[Rešenje 3.69]

**Zadatak 3.65** Definisati strukturu *Lopta* sa poljima *poluprecnik* (ceo broj u centimetrima) i *boja* (enumeracioni tip koji uključuje plavu, žutu, crvenu i zelenu boju). Zatim učitati informacije o  $n$  lopti ( $0 < n < 50$ ) i ispisati ukupnu zapreminu, kao i broj crvenih lopti. *Napomena: probati sa testiranjem zadataka pomoću preusmeravanja.*

*Primer 1*

```
INTERAKCIJA SA PROGRAMOM:
Unesite broj lopti: 4
Unesite dalje poluprecnike i boje lopti (1-plava, 2-zuta, 3-crvena, 4-zelena):
1.lopta: 4 1
2.lopta: 1 3
3.lopta: 2 3
4.lopta: 10 4
Ukupna zapremina: 4494.57
Broj crvenih lopti: 2
```

[Rešenje 3.69]

**Zadatak 3.66** Zimi su prehlade česte i treba unositi više vitamina C. Struktura *Vocka* sadrži ime voćke (nisku maksimalne dužine 20 karaktera) i količinu

vitamina C u miligramima (realan broj). Napisati program koji sa standardnog ulaza učitava podatke o voćkama sve do unosa reči KRAJ i ispisuje ime voćke sa najviše vitamina C. Pretpostaviti da broj voćki neće biti veći od 50. *Napomena: probati sa testiranjem zadataka pomoću preusmeravanja.*

#### Primer 1

```
INTERAKCIJA SA PROGRAMOM:
Unesite ime voćke i njenu količinu vitamina C: jabuka 4.6
Unesite ime voćke i njenu količinu vitamina C: limun 51
Unesite ime voćke i njenu količinu vitamina C: kivi 92.7
Unesite ime voćke i njenu količinu vitamina C: banana 8.7
Unesite ime voćke i njenu količinu vitamina C: pomorandža 53.2
Unesite ime voćke i njenu količinu vitamina C: KRAJ
Voce sa najvise C vitamina je: kivi
```

[Rešenje 3.69]

**Zadatak 3.67** Deda Mraz planira kupovinu poklona za studente koji su vredno učili C u toku godine. Na njegovoj listi se nalazi ime i prezime studenta (niske dužina do 50 karaktera) i njegova želja (niska maksimalne dužine 100 karaktera). Napisati program koji će služiti Deda Mrazu kao podsetnik: na osnovu liste koju je napravio, Deda Mraz može da unese ime i prezime studenta i da proveri njegovu želju. Ako ima više studenata sa istim imenom i prezimenom ispisati sve želje. *Napomena: probati sa testiranjem zadataka pomoću preusmeravanja.*

#### Primer 1

```
INTERAKCIJA SA PROGRAMOM:
Ime i prezime studenta:
Pera Peric
Njegova zelja:
privezak za kljucve
Jos vrednih studenata (da/ne)?
da
Ime i prezime studenta:
Zika Zikic
Njegova zelja:
stap za pecanje
Jos vrednih studenata (da/ne)?
da
Ime i prezime studenta:
Mara Maric
Njegova zelja:
komplet Knutovih knjiga
Jos vrednih studenata (da/ne)?
ne
Za podsecanje uneti ime i prezime:
Pera Peric
Novogodisnja zelja: privezak za kljucve
```

[Rešenje 3.69]

### 3 Predstavljanje podataka

---

**Zadatak 3.68** Definirati strukturu *Grad* u kojoj se nalazi ime grada (niška dužine 20 karaktera) i prosečna temperatura u toku decembra (realan broj). Napisati program koji učitava imena  $n$  ( $0 < n < 50$ ) gradova i njihove prosečne temperature, a zatim ispisuje one gradove koji imaju idealnu temperaturu za klizanje: od 3 do 8 stepeni. *Napomena: probati sa testiranjem zadataka pomoću preusmeravanja.*

#### Primer 1

```
INTERAKCIJA SA PROGRAMOM:
Unesite broj n: 4
Unesite grad i temperaturu: Beograd 7
Unesite grad i temperaturu: Uzice 1.5
Unesite grad i temperaturu: Subotica 4
Unesite grad i temperaturu: Zrenjanin 9
Gradovi sa idealnom temperaturom za klizanje u decembru:
Beograd
Subotica
```

#### Primer 2

```
INTERAKCIJA SA PROGRAMOM:
Unesite broj n: 2
Unesite grad i temperaturu: Varsava 11
Unesite grad i temperaturu: Prag 2
Gradovi sa idealnom temperaturom za klizanje u decembru:
```

[Rešenje 3.69]

**Zadatak 3.69** Definirati strukturu *ParReci* koja sadrži reč na srpskom jeziku i odgovarajući prevod na engleski jezik. Zatim sa standardnog ulaza sve do kraja ulaza učitavati parove reči i, posebno, za rečenicu koja se zadaje sa ulaza ispisati prevod - ako je reč u rečenici nepoznata umesto nje ispisati odgovarajući broj zvezdica. Reči neće biti duže od 50 karaktera, ukupan broj parova reči neće biti veći od 100, a ukupna dužina rečenice neće biti veća od 100 karaktera. *Napomena: probati sa testiranjem zadataka pomoću preusmeravanja.*

#### Primer 1

```
INTERAKCIJA SA PROGRAMOM:
zima winter
godina year
sreca happiness
programiranje programming
caj tea
Unesite recenicu za prevod:
piti caj zimi je sreca
**** tea **** ** happiness
```

[Rešenje 3.69]



## 3.6 Rešenja

### Rešenje 3.1

```
1  /*
   Napisati program koji racuna skalarni proizvod dva vektora. Svaki
   vektor
3  je zadat kao celobrojni niz sa najvise 100 elemenata. Program treba
   da
   ucita dimenziju nizova (oba niza su iste dimenzije), zatim jedan po
5  jedan element niza i da ispise njihov skalarni proizvod na
   standardni
   izlaz.
7  */

9  #include <stdio.h>
   #define MAX 100

11 /*
13 Pretprocesorskom direktivom define uvode se simbolicka imena (u ovom
   slucaju
15 MAX) kojima se pridruzuje nekakav tekst (u ovom slucaju 100). Pre
   kompilacije,
   sva pojavljivanja simbolickog imena MAX bice zamenjena pridruzenim
   tekstom
17 100. MAX nije promenljiva i za nju se tokom izvršavanja programa ne
   izdvaja
   memorijski prostor.

19 MAX se u ovom zadatku koristi kao maksimalni broj elemenata niza.
   Ukoliko bismo zeleli
21 da izmenimo ovu vrednost, npr. da povecamo sa 100 na 200, sve
   sto bi bilo neophodno uraditi je da izmenimo tekst sa 100 na 200. Sa
   druge
23 strane, da nismo koristili pretprocesorsku direktivu i da smo svaki
   put
   umesto MAX direktno navodili vrednost 100, morali bismo da je
   izmenimo na svakom
25 mestu u kodu.

27 */
   int main()
29 {
   int a[MAX];
   int b[MAX];
31 int n;
   int i;
33 int s;
35
```

### 3 Predstavljanje podataka

---

```
37 printf("Unesi dimenziju niza:");
   scanf("%d", &n);
39
   if (n<1 || n>100)
41   {
       printf("Neispravan unos\n");
43       return -1;
   }
45
   /*
47    prvi element niza ima indeks 0, a poslednji n-1,
       gde je n broj elemenata niza; elementima niza pristupamo
49    preko indeksa; na primer, ako niz a ima 5 elemenata, mozemo
       im pristupiti pomocu
51    a[0], a[1], a[2], a[3], a[4]
53
   */
55   for (i=0; i<n; i++)
   {
57       printf("a[%d]=",i);
       scanf("%d", &a[i]);
59   }
61   for (i=0; i<n; i++)
   {
63       printf("b[%d]=",i);
       scanf("%d", &b[i]);
65   }
67   s=0;
69   for (i=0; i<n; i++)
       s = s + a[i]*b[i];
71   printf("Skalarni proizvod: %d\n",s);
73   return 0;
}
```

#### Rešenje 3.2

```
/*
2   Napisati program koji učitava broj elemenata niza (n<=100),
   zatim učitava elemente niza i ispisuje:
4   a) elemente niza koji se nalaze na parnim indeksima
   b) parne elemente niza
6
   */
8
#include <stdio.h>
```

```
10 #define MAX 100
12 int main()
13 {
14     int a[MAX];
15     int n;
16     int i;
17
18     printf("Unesi dimenziju niza:");
19     scanf("%d", &n);
20
21     if (n<1 || n>MAX)
22     {
23         printf("Nekorektan unos\n");
24         return -1;
25     }
26
27
28
29     for (i=0; i<n; i++)
30     {
31         printf("a[%d]=",i);
32         scanf("%d", &a[i]); /* učitavamo jedan po jedan element niza */
33     }
34
35     printf("Elementi sa parnim indeksima:\n");
36     for (i=0; i<n; i+=2)
37         printf("a[%d]=%d\n",i,a[i]);
38
39     printf("Parni elementi:\n");
40     for (i=0; i<n; i++)
41         if (a[i]%2==0)
42             printf("a[%d]=%d\n",i,a[i]);
43
44
45     return 0;
46 }
```

### Rešenje 3.3

```
1 /*
2     Napisati program koji učitava jedan ceo broj a zatim ispisuje
3     koliko puta koja cifra ucestvuje
4     u zapisu tog broja. Nije potrebno ispisivati da se neka cifra
5     pojavila 0 puta.
6
7     Na primer, za uneti broj 4611, izlaz treba da bude:
8
9     U zapisu broja 4611, cifra 1 se pojaviljuje 2 puta
10    U zapisu broja 4611, cifra 4 se pojaviljuje 1 puta
```

### 3 Predstavljanje podataka

---

```
9      U zapisu broja 4611, cifra 6 se pojaviljuje 1 puta
11     A za uneti broj -252
13     U zapisu broja -252, cifra 2 se pojaviljuje 2 puta
14     U zapisu broja -252, cifra 5 se pojaviljuje 1 puta
15
16  */
17
18  #include<stdio.h>
19  #include<stdlib.h>
20  #define MAX 100
21
22  int main()
23  {
24      int x;
25      int brojac[10];
26      char cifra;
27      int original;
28      int i;
29
30      printf("Unesi jedan ceo broj:");
31      scanf("%d",&x);
32
33      /*
34         svaki element niza brojac predstavlja
35         brojac za jednu cifru:
36         brojac[0] sadrzi broj nula
37         brojac[1] sadrzi broj jedinica
38         ...
39         brojac[9] sadrzi broj devetki
40
41         brojac se inicijalizuju na vrednost 0
42     */
43
44     for(i=0;i<10;i++)
45         brojac[i]=0;
46
47     /*
48         vrednost promenljive x ce biti unistena
49         u while petlji jer je u svakom koraku delimo
50         sa 10; njenu vrednost cuvamo u promenljivoj
51         original kako bismo mogli da je iskoristimo
52         na kraju prilikom ispisa
53     */
54     original = x;
55
56     /*
57         Uzimamo apsolutnu vrednost broja za slucaj
58         da je uneti broj negativan
59     */
60     x=abs(x);
```

```

61      /* Izdvajanje cifara broja */
62      do
63      {
64          cifra = x%10;
65          brojaci[cifra]++; /* Uvecavamo brojac odgovarajuće cifre */
66          x/=10;
67      } while(x);
68
69      /* Ispis brojaca koji su razliciti od nule */
70      for(i=0;i<10;i++)
71          if(brojaci[i])
72              printf("U zapisu broja %d, cifra %d se pojaviljuje %d puta\n",
73                  original, i, brojaci[i]);
74
75      return 0;
76  }

```

### Rešenje 3.4

```

1  /* Napisati program koji ucitava karakter po karakter do EOF i
   ispisuje koliko se puta
   u unetom tekstu pojavila svaka cifra, svako malo slovo i svako
   veliko slovo. Ispisati
3  broj pojavljivanja samo za ona mala slova, velika slova i cifre
   koji su se u unetom
   tekstu pojavili >0 puta.
5  */
7  #include <stdio.h>
9
10 int main()
11 {
12     /* Za svaku dekadnu cifru definisemo jedan brojac (tj. imamo niz
13        od 10 brojaca): brojaci[0] broji koliko se puta pojavio karakter
14        '0', brojaci[1] broji koliko se puta pojavio karakter '1' i tako
15        dalje. Svi brojaci se inicijalizuju nulama.
16     */
17     int cifre[10];
18     int mala[26];
19     int velika[26];
20
21     int c, i;
22
23     for(i=0;i<10;i++)
24         cifre[i]=0;
25
26     for(i=0;i<26;i++)
27     {

```

### 3 Predstavljanje podataka

```
29     mala[i]=0;
    velika[i]=0;
31 }
33 while((c = getchar()) != EOF)
34 {
35     if (c>='A' && c<='Z')
36         velika[c-'A']++;
37     else if (c>='a' && c<='z')
38         mala[c-'a']++;
39     else if(c >='0' && c <= '9') /* Ako je karakter c dekadna cifra
... */
        cifre[c-'0']++;          /* Uvecavamo odgovarajuci brojac za
1 */
41
42     /*
43     Izraz c - '0' ce u slucaju da je c dekadna cifra imati
44     upravo
45     vrednost 0, 1, ..., 9 za karaktere '0', '1', ..., '9' respektivno,
46     a to su upravo indeksi u nizu brojac (jer niz ima 10 elemenata,
47     pa su indeksi od 0 do 9). Time postizemo da brojac[0] broji
48     karaktere '0', itd. Isto vazi i za brojace za mala i velika slova.
49     */
50 }
51
52 /* Prikazujemo elemente niza, tj. vrednosti brojaca: */
53 for(i = 0; i < 10; i++)
54     if (cifre[i]!=0)
55         printf("Karakter %c se pojavljuje %d puta\n", '0' + i,
56             cifre[i]);
57
58 for(i = 0; i < 26; i++)
59     if (mala[i]!=0)
60         printf("Karakter %c se pojavljuje %d puta\n", 'a' + i,
61             mala[i]);
62
63 for(i = 0; i < 26; i++)
64     if (velika[i]!=0)
65         printf("Karakter %c se pojavljuje %d puta\n", 'A' + i,
66             velika[i]);
67
68 return 0;
69 }
```

#### Rešenje 3.5

```
2 /*
    Napisati program koji učitava dimenziju n dva celobrojna niza a i b
    (oba niza su iste dimenzije),
```

```

    zatim ucitava elemente oba niza i formira treci niz c tako sto
    naizmenicno rasporedjuje
4   elemente nizova a i b unutar njega: a_0,b_0,a_1,b_1,...,a_(n-1),b_(
    n-1). Program treba
    da ispise elemente novog niza c na standardni izlaz. Mozemo
    pretpostaviti da je maksimalni
6   broj elemenata u nizovima a i b 100.
    */
8
    #include <stdio.h>
10  #define MAX 100

12  int main()
    {
14      int a[MAX];
        int b[MAX];
16      int c[2*MAX];

18      int n;
        int i,j;
20

        printf("Unesi dimenziju niza:");
22      scanf("%d", &n);

24      if (n<1 || n>MAX)
        {
26          printf("Neispravan unos\n");
            return -1;
28      }

30

        printf("\nUnesi elemente niza a:\n");
32      for(i=0;i<n;i++)
        {
34          printf("a[%d]=",i);
            scanf("%d", &a[i]);
36      }

38      printf("\nUnesi elemente niza b:\n");
        for(i=0;i<n;i++)
        {
40          printf("b[%d]=",i);
            scanf("%d", &b[i]);
42      }

44      /*
46      Koristimo dva indeksa:
        1. i, sa kojim pristupamo
48          elementima niza a i b, i koji uvecavamo za 1
            nakon svake iteracije,
        2. j, sa kojim pristupamo
50          elementima niza c; s obzirom da u svakoj

```

### 3 Predstavljanje podataka

---

```
52         iteraciji dodeljujemo vrednost za dva
           elementa niza c (c[j] i c[j+1]), indeks
54         j uvecavamo za 2 nakon svake iteracije
           */
56     for(i=0,j=0;i<n;i++,j+=2)
           {
58         c[j]=a[i];
           c[j+1]=b[i];
60     }

62     printf("\nNiz c:\n");
           for(i=0;i<2*n;i++)
64         printf("c[%d]=%d\n",i,c[i]);

66     return 0;
}
```

#### Rešenje 3.6

```
1  /*
           Napisati program koji ucitava dimenziju n celobrojnog niza a i
           njegove elemente, a zatim iz niza a izbacuje sve elemente
           koji nisu deljivi svojom poslednjom cifrom, izuzev elemenata
           cija je poslednja cifra 0 koji treba zadržati. Program treba da
           ispise
           izmenjeni niz na standardni izlaz. Mozemo pretpostaviti da niz a
           sadrzi najvise 100 elemenata.
           */
9
           #include <stdio.h>
11          #define MAX 100

13          int main()
           {
15
           int a[MAX];

17
           int n;
           int i,j;
           char poslednja_cifra;
           int novo_n;

23          printf("Unesi dimenziju niza:");
           scanf("%d", &n);

25
           if (n<1 || n>MAX)
27           {
           printf("Neispravan unos\n");
           return -1;
29         }

31
```



```

33 printf("\nUnesi elemente niza a:\n");
34 for(i=0;i<n;i++)
35 {
36     printf("a[%d]=",i);
37     scanf("%d", &a[i]);
38 }
39
40
41 /*
42  Dodatni indeks j se uvecava u slucaju da element na indeksu
43  i treba da ostane u nizu, tj da je deljiv svojim
44  indeksom i; u suprotnom, j se nece uvecati i
45  element i ce u narednoj iteraciji biti zamenjen elementom koji
46  jeste deljiv svojim indeksom
47 */
48
49 for(i=0,j=0;i<n;i++)
50 {
51     poslednja_cifra = a[i]%10;
52
53     /*
54      zbog lenjog izracunavanja, ako je prvi uslov
55      u disjunkciji tacan, drugi se nece ispitivati
56      (jer ce tada disjunkcija biti tacna bez obzira
57      da li je drugi uslov tacan ili ne)
58     */
59     if (poslednja_cifra==0 || a[i]%poslednja_cifra==0)
60     {
61         a[j]=a[i];
62         j++;
63     }
64 }
65 /*
66  Izbacivanjem elemenata dimenzija niza se menja, odnosno
67  smanjuje se za broj izbacenih elemenata
68 */
69 novo_n=j;
70
71 printf("Nakon izmena:\n");
72 for(i=0;i<novo_n;i++)
73     printf("a[%d]=%d\n",i,a[i]);
74
75 return 0;
76 }

```

### Rešenje 3.7

```

/*
2  a) Napisati funkciju koja učitava sadržaj niza.
   b) Napisati funkciju koja stampa sadržaj niza.

```

### 3 Predstavljanje podataka

---

```
4      c) Napisati funkciju koja racuna sumu elemenata niza.
5      d) Napisati funkciju koja racuna prosechnu vrednost elemenata niza.
6      e) Napisati funkciju koja izracunava minimum elemenata niza.
7      f) Napisati funkciju koja izracunava poziciju maksimalnog elementa
8          u nizu.
9      g) Napisati program koji testira prethodne funkcije.
10
11 */
12 #include <stdio.h>
13 #define MAX 100
14
15 /* a) */
16 void ucitaj(int a[], int n)
17 {
18     int i;
19     for(i=0;i<n;i++)
20     {
21         printf("Unesi element na poziciji %d:",i);
22         scanf("%d",&a[i]);
23     }
24 }
25
26 /* b) */
27 void stampaj(int a[], int n)
28 {
29     int i;
30     for(i=0;i<n;i++)
31         printf("%d\t",a[i]);
32     printf("\n");
33 }
34
35 /* c) */
36 int suma(int a[], int n)
37 {
38     int i;
39     int s=0;
40     for(i=0;i<n;i++)
41         s+=a[i];
42     return s;
43 }
44
45 /* d) */
46 float prosek(int a[], int n)
47 {
48     int i;
49     int s = suma(a,n);
50     return (float) s/n;
51 }
52
53
54
```

```
/* e) */
56 int minimum (int a[],int n)
{
58     int m;
    int i;
60     m = a[0];

62     /*
        minimum inicijalizujemo na prvi element niza (a[0])
64     u svakom koraku poredimo vrednost minimuma
        sa jednim elementom niza, iduci redom; s obzirom
66     da je minimum inicijalizovan na a[0], nema potrebe
        porediti a[0] sa a[0] i zbog toga indeksiranje kreće
68     od 1
    */

70     for(i=1;i<n;i++)
72         if (m>a[i])
            m = a[i];

74     return m;
76 }

78 /* f) */
80 int max_pozicija (int a[],int n)
{
82     int m;
    int m_poz;
84     int i;
    m = a[0];
86     m_poz=0;

88     for(i=1;i<n;i++)
89         if (m<a[i])
90             {
92                 m = a[i];
                m_poz=i;
94             }

96     return m_poz;
98 }

100 int main()
102 {
    int a[MAX];
    int n;
104     printf("Unesi dimenziju niza:");
106     scanf("%d",&n);
```

### 3 Predstavljanje podataka

```
108     if (n<1 || n>MAX)
109     {
110         printf("Nekorektan unos\n");
111         return -1;
112     }
113
114     učitaj(a,n);
115     printf("Učitani niz:");
116     stampaj(a,n);
117
118     printf("Suma elemenata niza: %d\n", suma(a,n));
119     printf("Prosečna vrednost elemenata niza: %.2f\n", prosek(a,n));
120     printf("Minimumalni element niza: %d\n", minimum(a,n));
121     printf("Indeks maksimalnog elementa niza: %d\n", max_pozicija(a,n)
122           );
123     return 0;
124 }
```

#### Rešenje 3.8

```
/*
2   a) Napisati funkciju koja učitava sadržaj niza.
3   b) Napisati funkciju koja stampa sadržaj niza.
4   c) Napisati funkciju koja proverava da li niz sadrži neku vrednost
5       m.
6   d) Napisati funkciju koja vraća vrednost prve pozicije na kojoj se
7       nalazi element koji ima vrednost m, ili -1 ukoliko element nije u
8       nizu.
9   e) Napisati funkciju koja vraća vrednost poslednje pozicije na
10      kojoj se
11      nalazi element koji ima vrednost m, ili -1 ukoliko element nije u
12      nizu.
13      f) Napisati funkciju koja proverava da li elementi niza čine
14          palindrom.
15      g) Napisati funkciju koja proverava da li su elementi niza
16          uređeni
17          neopadajuće.
18      h) Napisati funkciju koja izračunava najdužu uzastopnu seriju
19          jednakih
20          elemenata u nizu. Na primer, za uneti niz 1 2 3 4 4 4 5 6 7 8 9 9
21          funkcija
22          treba da vrati 3.
23      i) Napisati program koji testira prethodne funkcije.
24 */
#include <stdio.h>
#define MAX 100

/* a) */
void učitaj(int a[], int n)
```

```
22 {
23     int i;
24     for(i=0;i<n;i++)
25     {
26         printf("Unesi element na poziciji %d:",i);
27         scanf("%d",&a[i]);
28     }
29 }
30
31 /* b) */
32 void stampaj(int a[], int n)
33 {
34     int i;
35     for(i=0;i<n;i++)
36         printf("%d\t",a[i]);
37     printf("\n");
38 }
39
40 /* c) */
41 int sadrzi(int a[], int n, int m)
42 {
43     int i;
44     /*
45      * poredimo jedan po jedan element niza a sa datim m; ukoliko
46      * ustanovimo jednakost, to znaci da niz sadrzi element jednak
47      * m i vracamo 1
48      */
49     for(i=0;i<n;i++)
50         if (a[i]==m)
51             return 1;
52
53     /*
54      * ukoliko se petlja završi a uslov a[i]==m nijednom nije bio
55      * ispunjen,
56      * to znaci da se broj m ne nalazi u nizu a i da funkcija treba da
57      * vrati 0
58      */
59     return 0;
60 }
61
62 /* d) */
63 int prvo_pojavljivanje(int a[], int n, int m)
64 {
65     int i;
66     /*
67      * poredimo jedan po jedan element niza a sa datim m; ukoliko
68      * ustanovimo jednakost, vracamo indeks elementa niza a koji
69      * je jednak sa m
70      */
71     for(i=0;i<n;i++)
72         if (a[i]==m)
```

### 3 Predstavljanje podataka

---

```
72         return i;

74     /*
       ukoliko se petlja završi a uslov a[i]==m nijednom nije bio
       ispunjen,
76     to znaci da se broj m ne nalazi u nizu a i da funkcija treba da
       vrati -1
       */
78     return -1;
}

80
/* e) */
82 int poslednje_pojavljivanje(int a[], int n, int m)
{
84     int i;
86     /*
       krecemo od indeksa poslednjeg elementa, n-1
       */
88     for(i=n-1; i>=0; i--)
90         if (a[i]==m)
92             return i;
94     return -1;
}

96 /* f) */
97 int palindrom(int a[], int n)
98 {
99     int i, j;
100
101     /*
102     uporedjujemo element na poziciji 0 sa elementom na poziciji n-1
103     uporedjujemo element na poziciji 1 sa elementom na poziciji n-2
104     .
105     .
106     i tako redom dok je prva pozicija manja od druge
       */
108     for(i=0, j=n-1; i<j; i++, j--)
110         if (a[i]!=a[j])
112             return 0;
114     return 1;
}

116 /* g) */
117 int neopadajuci(int a[], int n)
118 {
119     int i;
120
121     /*
       Funkcija neopadajuci proverava da li je dati niz sortiran
```

```

122     neopadajuće i vraća
123     1 ako jeste, a 0 u suprotnom

124     Sortiranost proveravamo na sledeći način: za svaki par susednih
        elemenata
125     a[0] i a[1], a[1] i a[2], a[2] i a[3], ..., a[n-2] i a[n-1]
        proveravamo
126     da li važi da je drugi član para manji od prvog. Ako naidjemo na
        par za koji
        to ne važi, niz nije sortiran i funkcija vraća 0. Ukoliko se
        petlja završi
128     a da pritom uslov a[i]<a[i-1] nije nijednom bio ispunjen, to znaci
        da je
        niz sortiran i funkcija vraća 1

130
131     */
132     for(i=1; i<n; i++)
133         if (a[i]<a[i-1])
134             return 0;

136     return 1;
137 }

138 /* h) */
140 int najduza_konstanta(int a[], int n)
141 {
142     int i; /* indeks niza */
143     int j; /* dužina intervala */
144     int dužina;
145     int max_dužina=0;

146
147     for(i=0, j=0; i<n-1; i++)
148     {
149         if(a[i]==a[i+1]) /* nalazimo se unutar konstantnog intervala */
150         {
151             j++;          /* uvecavamo dužinu konstantnog intervala */

152
153             /*
154              * ako se niz završava konstantnim intervalom (nalazimo se u
155              * poslednjoj
156              * iteraciji petlje i tada je i==n-2), ispitujemo da li je
157              * taj konstantni
158              * interval maksimalne dužine
159              */
160             if(i==n-2)
161             {
162                 j++;
163                 if(j>max_dužina)
164                     max_dužina=j;
165             }
166         }
167     }

```

### 3 Predstavljanje podataka

---

```
166     else
167     {
168         /*
169             izasli smo iz konstantnog intervala
170
171             ukoliko smo imali bar dva elementa u konstantnom
172             intervalu,
173             vrednost promenljive j ce biti 1, a duzina tog intervala
174             je 2;
175             zbog toga je neophodno takve (pozitivne) j uvecati za 1;
176
177             sa druge strane, ako su a[i] i a[i+1] razliciti,
178             duzina tog intervala je 0
179         */
180
181         if (j>0)
182             j++;
183
184         /* azuriramo maksimalnu duzinu uspona */
185         if(j>max_duzina)
186             max_duzina=j;
187         /*
188             duzina uspona se postavlja na nulu
189             kako bi mogli da je iskoristimo
190             za naredni uspon
191         */
192         j=0;
193     }
194
195 }
196
197 return max_duzina;
198 }
199
200 int main()
201 {
202     int a[MAX];
203     int n;
204     int m;
205     int i;
206
207     printf("Unesi dimenziju niza:");
208     scanf("%d",&n);
209
210     if (n<1 || n>MAX)
211     {
212         printf("Nekorektan unos\n");
213         return -1;
214     }
215 }
```



```

216     ucitaj(a,n);
218     printf("Ucitani niz:");
219     stampaj(a,n);
220
221     printf("Unesi jedan ceo broj:");
222     scanf("%d",&m);
223
224     if(sadrzi(a,n,m))
225         printf("Niz sadrzi element cija je vrednost %d\n", m);
226     else
227         printf("Niz ne sadrzi element cija je vrednost %d\n", m);
228
229     i = prvo_pojavljivanje(a,n,m);
230     if(i!=-1)
231         printf("Niz sadrzi element cija je vrednost %d. Indeks njegovog
232             prvog pojavljivanja u nizu je %d\n", m,i);
233     else
234         printf("Niz ne sadrzi element cija je vrednost %d\n", m);
235
236     i = poslednje_pojavljivanje(a,n,m);
237     if(i!=-1)
238         printf("Niz sadrzi element cija je vrednost %d. Indeks njegovog
239             poslednjeg pojavljivanja u nizu je %d\n", m,i);
240     else
241         printf("Niz ne sadrzi element cija je vrednost %d\n", m);
242
243     if(palindrom(a,n))
244         printf("Elementi niza cine palindrom\n");
245     else
246         printf("Elementi niza ne cine palindrom\n");
247
248     if(neopadajuci(a,n))
249         printf("Niz je sortiran neopadajuće\n");
250     else
251         printf("Niz nije sortiran neopadajuće\n");
252
253     printf("Duzina najduzeg konstantnog intervala: %d\n",
254         najduza_konstanta(a,n));
255
256     return 0;
}

```

### Rešenje 3.9

```

1  /*
2
3  a) Napisati funkciju koja ucitava sadrzaj niza.
4  b) Napisati funkciju koja stampa sadrzaj niza.

```

### 3 Predstavljanje podataka

---

```

    c) Napisati funkciju koja proverava da li niz sadrzi neku vrednost
       m.
5   d) Napisati funkciju koja vraća vrednost prve pozicije na kojoj se
    nalazi element koji ima vrednost m, ili -1 ukoliko element nije u
       nizu.
7   e) Napisati funkciju koja vraća vrednost poslednje pozicije na
    kojoj se
    nalazi element koji ima vrednost m, ili -1 ukoliko element nije u
       nizu.
9   f) Napisati funkciju koja proverava da li elementi niza cine
    palindrom.
    g) Napisati funkciju koja proverava da li su elementi niza
       uredjeni
11  neopadajuće.
    h) Napisati funkciju koja izracunava najduzu uzastopnu seriju
       jednakih
13  elemenata u nizu. Na primer, za uneti niz 1 2 3 4 4 4 5 6 7 8 9 9
    funkcija
    treba da vrati 3.
15  i) Napisati program koji testira prethodne funkcije.
    */
17  #include <stdio.h>
    #define MAX 100
19
    /* a) */
21  void ucitaj(int a[], int n)
    {
23      int i;
      for(i=0;i<n;i++)
25      {
          printf("Unesi element na poziciji %d:",i);
27          scanf("%d",&a[i]);
      }
29  }

    /* b) */
31  void stampaj(int a[], int n)
33  {
      int i;
35      for(i=0;i<n;i++)
          printf("%d\t",a[i]);
37      printf("\n");
    }
39

    /* c) */
41  int sadrzi(int a[], int n, int m)
43  {
      int i;
45      /*
       poredimo jedan po jedan element niza a sa datim m; ukoliko
47      ustanovimo jednakost, to znaci da niz sadrzi element jednak

```

```

    m i vratamo 1
49 */
    for(i=0;i<n;i++)
51         if (a[i]==m)
                return 1;
53
    /*
55     ukoliko se petlja završi a uslov a[i]==m nijednom nije bio
        ispunjen,
        to znači da se broj m ne nalazi u nizu a i da funkcija treba da
        vrati 0
57 */
    return 0;
59 }

61 /* d) */
int prvo_pojavljivanje(int a[], int n, int m)
63 {
    int i;
65     /*
        poredimo jedan po jedan element niza a sa datim m; ukoliko
67     ustanovimo jednakost, vratamo indeks elementa niza a koji
        je jednak sa m
69     */
    for(i=0;i<n;i++)
71         if (a[i]==m)
                return i;
73
    /*
75     ukoliko se petlja završi a uslov a[i]==m nijednom nije bio
        ispunjen,
        to znači da se broj m ne nalazi u nizu a i da funkcija treba da
        vrati -1
77 */
    return -1;
79 }

81 /* e) */
int poslednje_pojavljivanje(int a[], int n, int m)
83 {
    int i;
85     /*
        krecemo od indeksa poslednjeg elementa, n-1
87     */
    for(i=n-1;i>=0;i--)
89         if (a[i]==m)
                return i;
91
    return -1;
93 }

95 /* f) */
```

### 3 Predstavljanje podataka

---

```
int palindrom(int a[], int n)
{
    int i,j;

    /*
    uporedjujemo element na poziciji 0 sa elementom na poziciji n-1
    uporedjujemo element na poziciji 1 sa elementom na poziciji n-2
    .
    .
    i tako redom dok je prva pozicija manja od druge
    */
    for(i=0,j=n-1;i<j;i++,j--)
        if(a[i]!=a[j])
            return 0;

    return 1;
}

/* g) */
int neopadajuci(int a[], int n)
{
    int i;

    /*
    Funkcija neopadajuci proverava da li je dati niz sortiran
    neopadajuće i vraća
    1 ako jeste, a 0 u suprotnom

    Sortiranost proveravamo na sledeći način: za svaki par susednih
    elemenata
    a[0] i a[1], a[1] i a[2], a[2] i a[3], ..., a[n-2] i a[n-1]
    proveravamo
    da li važi da je drugi član para manji od prvog. Ako nađemo na
    par za koji
    to ne važi, niz nije sortiran i funkcija vraća 0. Ukoliko se
    petlja završi
    a da pritom uslov a[i]<a[i-1] nije nijednom bio ispunjen, to znači
    da je
    niz sortiran i funkcija vraća 1
    */
    for(i=1; i<n; i++)
        if (a[i]<a[i-1])
            return 0;

    return 1;
}

/* h) */
int najduza_konstanta(int a[], int n)
{
```

```
143 int i; /* indeks niza */
144 int j; /* duzina intervala */
145 int duzina;
146 int max_duzina=0;

147
148 for(i=0,j=0;i<n-1;i++)
149 {
150     if(a[i]==a[i+1]) /* nalazimo se unutar konstantnog intervala */
151     {
152         j++;          /* uvecavamo duzinu konstantnog intervala */
153
154         /*
155         ako se niz zavrшава konstantnim intervalom (nalazimo se u
156 poslednjoj
157 iteraciji petlje i tada je i==n-2), ispitujemo da li je
158 taj konstantni
159 interval maksimalne duzine
160 */
161         if(i==n-2)
162         {
163             j++;
164             if(j>max_duzina)
165                 max_duzina=j;
166         }
167     }
168     else
169     {
170         /*
171         izašli smo iz konstantnog intervala
172
173         ukoliko smo imali bar dva elementa u konstantnom
174 intervalu,
175 vrednost promenljive j ce biti 1, a duzina tog intervala
176 je 2;
177 zbog toga je neophodno takve (pozitivne) j uvecati za 1;
178
179 sa druge strane, ako su a[i] i a[i+1] razliciti,
180 duzina tog intervala je 0
181 */
182         if (j>0)
183             j++;
184
185         /* azuriramo maksimalnu duzinu uspona */
186         if(j>max_duzina)
187             max_duzina=j;
188
189         /*
190         duzina uspona se postavlja na nulu
191 kako bi mogli da je iskoristimo
192 za naredni uspon
193 */
194     }
```

```

        j=0;
191     }
193
195 }
197     return max_duzina;
199 }
201 int main()
202 {
203     int a[MAX];
204     int n;
205     int m;
206     int i;
207
208     printf("Unesi dimenziju niza:");
209     scanf("%d",&n);
210
211     if (n<1 || n>MAX)
212     {
213         printf("Nekorektan unos\n");
214         return -1;
215     }
216
217     učitaj(a,n);
218     printf("Učitani niz:");
219     stampaj(a,n);
220
221     printf("Unesi jedan ceo broj:");
222     scanf("%d",&m);
223
224
225     if(sadrzi(a,n,m))
226         printf("Niz sadrzi element cija je vrednost %d\n", m);
227     else
228         printf("Niz ne sadrzi element cija je vrednost %d\n", m);
229
230     i = prvo_pojavljivanje(a,n,m);
231     if(i!=-1)
232         printf("Niz sadrzi element cija je vrednost %d. Indeks njegovog
233         prvog pojavljivanja u nizu je %d\n", m,i);
234     else
235         printf("Niz ne sadrzi element cija je vrednost %d\n", m);
236
237     i = poslednje_pojavljivanje(a,n,m);
238     if(i!=-1)
239         printf("Niz sadrzi element cija je vrednost %d. Indeks njegovog
240         poslednjeg pojavljivanja u nizu je %d\n", m,i);

```

```

241     else
242         printf("Niz ne sadrzi element cija je vrednost %d\n", m);
243
244     if(palindrom(a,n))
245         printf("Elementi niza cine palindrom\n");
246     else
247         printf("Elementi niza ne cine palindrom\n");
248
249     if(neopadajuci(a,n))
250         printf("Niz je sortiran neopadajuće\n");
251     else
252         printf("Niz nije sortiran neopadajuće\n");
253
254     printf("Duzina najduzeg konstantnog intervala: %d\n",
255           najduza_konstanta(a,n));
256
257     return 0;
258 }

```

### Rešenje 3.10

```

1  /*
2  a) Napisati funkciju koja sve vrednosti niza uvecava za vrednost m.
3  b) Napisati funkciju koja obrće vrednosti elementima niza.
4  c) Napisati funkciju koja rotira niz ciklicno za jedno mesto u levo
5  d) Napisati funkciju koja rotira niz ciklicno za k mesta u levo.
6  e) Napisati program koji testira prethodne funkcije.
7
8  Napisati potom glavni program koji testira ovu funkciju.
9  */
10
11 #include<stdio.h>
12 #define MAX 100
13
14 void ucitaj(int a[], int n)
15 {
16     int i;
17     for(i=0;i<n;i++)
18     {
19         printf("Unesi element na poziciji %d:",i);
20         scanf("%d",&a[i]);
21     }
22 }
23
24 void stampaj(int a[], int n)
25 {
26     int i;
27     for(i=0;i<n;i++)
28         printf("%d\t",a[i]);

```

### 3 Predstavljanje podataka

---

```
29     printf("\n");
30 }
31
32 void uvecaj(int a[], int n, int m)
33 {
34     int i;
35     for(i=0;i<n;i++)
36         a[i]+=m;
37 }
38
39
40 void obrni(int a[], int n)
41 {
42     int t;
43     int i,j;
44     /*
45      * Niz obrnemo tako sto razmenimo vrednosti elemenata na pozicijama
46      * 0 i n-1,
47      * zatim 1 i n-2, 2 i n-3 i tako redom dok je prva pozicija manja od
48      * druge
49      */
50
51     for(i=0,j=n-1;i<j;i++, j--)
52     {
53         t = a[i];
54         a[i] = a[j];
55         a[j] = t;
56     }
57 }
58
59 void rotiraj1(int a[], int n)
60 {
61     int i;
62     int tmp;
63     tmp=a[0]; /* izdvajamo prvi element */
64     for(i=0;i<n-1;i++)
65         a[i]=a[i+1]; /* pomeramo preostale elemente */
66     a[n-1] = tmp; /* poslednjem elementu dodeljujemo
67                  * sacuvanu vrednost prvog elementa */
68 }
69
70 void rotirajk(int a[], int n, int k)
71 {
72     int i;
73     /*
74      * k puta rotiramo niz za jednu poziciju
75      * ulevo
76      */
77     for(i=0;i<k;i++)
```



```
79     rotiraj1(a,n);
80 }
81
82 int main()
83 {
84     int a[MAX];
85     int n;
86     int i;
87     int k;
88     int m;
89
90     printf("Unesi dimenziju niza:");
91     scanf("%d",&n);
92
93     if (n<1 || n>MAX)
94     {
95         printf("Nekorektan unos\n");
96         return -1;
97     }
98
99     ucitaj(a,n);
100
101     printf("Unesi jedan ceo broj:");
102     scanf("%d", &m);
103
104     uvecaj(a,n,m);
105     printf("Elementi niza nakon uvecanja za %d:\n",m);
106     stampaj(a,n);
107
108     obrni(a,n);
109     printf("Elementi niza nakon obrtanja:\n");
110     stampaj(a,n);
111
112     printf("Unesi jedan pozitivan ceo broj:");
113     scanf("%d",&k);
114
115     if (k<=0)
116     {
117         printf("Nekorektan unos\n");
118         return -1;
119     }
120
121     rotiraj1(a,n);
122     printf("Elementi niza nakon rotiranja za 1 mesto ulevo:\n");
123     stampaj(a,n);
124
125     rotirajk(a,n,k);
126     printf("Elementi niza nakon rotiranja za %d mesto ulevo:\n",k);
127     stampaj(a,n);
128
129     return 0;
130 }
```

Rešenje 3.26

Rešenje 3.26

Rešenje 3.26

Rešenje 3.26

Rešenje 3.26

Rešenje 3.26

Rešenje 3.26

Rešenje 3.26

Rešenje 3.26

Rešenje 3.26

Rešenje 3.26

Rešenje 3.26

Rešenje 3.26

Rešenje 3.26

Rešenje 3.26

Rešenje 3.26

Rešenje 3.27

```
1  /*  
3  Napisati funkciju uredi koja uredjuje svoja dva  
celobrojna argumenta tako da se u prvom nalazi manji  
a u drugom veci. Napisati potom glavni program koji
```

```
5   ucitava dva cela broja i uredjuje njihove vrednosti
   primenom napisane funkcije. Na primer, ako su učitane
7   promenljive x=5 i y=2, njihove vrednosti nakon
   primene funkcije uredi treba da budu x=2 i y=5.
9  */

11 #include <stdio.h>

13 /*
   Argumenti funkcije uredi_pogresno, promenljive a i b,
15   predstavljaju lokalne promenljive za ovu funkciju
   i prestaju da postoje po završetku funkcije. Zbog toga
17   se efekti razmene vrednosti promenljivih a i b u slucaju
   da je a>b vide u funkciji, ali se ne vide u glavnom programu.
19  */
void uredi_pogresno(int a, int b)
21 {
   int t;
23
   if (a>b)
25   {
       t = a;
27       a = b;
       b = t;
29   }
   printf("uredi_pogresno :: a=%d, b=%d\n", a, b);
31   printf("uredi_pogresno :: &a=%p, &b=%p\n", &a, &b);
   }

33 /*
35   Argumenti funkcije uredi_tacno, promenljive pa i pb,
   takodje su lokalne promenljive za ovu funkciju i
37   prestaju da postoje kada se funkcija završi.
   Njima prosledjujemo adrese promenljivih a i b koje zelimo
39   da razmenimo u slucaju da je a>b.

41   Promenljivoj a pristupamo preko pokazivacke promenljive
   pa sa *pa i slicno, promenljivoj pb pristupamo sa *pb.
43
45   Vrednosti promenljivih *pa i *pb razmenjujemo kao
   i vrednosti bilo koje dve celobrojne promenljive.
47  */
void uredi_tacno(int * pa, int * pb)
49 {
   int t;
51   if (*pa>*pb)
   {
53       t = *pa;
       *pa = *pb;
55       *pb = t;
   }
```

### 3 Predstavljanje podataka

```
57 printf("uredi_tacno :: *pa=%d, *pb=%d\n ", *pa, *pb);
   printf("uredi_tacno :: pa=%p, pb=%p\n ", pa, pb);
59
   }
61 int main()
   {
63     int a,b;

65     printf("Unesi dve celobrojne promenljive:");
     scanf("%d%d",&a,&b);

67
     printf("main :: a=%d, b=%d\n", a,b);
69     printf("main :: &a=%p, &b=%p\n", &a, &b);
     uredi_pogresno(a,b);
71     printf("main :: nakon uredi_pogresno, a=%d, b=%d\n", a, b);

73     /*
       Funkcija uredi_tacno kao argument ima dve pokazivacke
       promenljive
75     (int*,int*). Zbog toga joj je u pozivu funkcije neophodno
       proslediti
       adrese promenljivih koje zelimo da uredimo rastuce, &a i &b.
77     */

79     uredi_tacno(&a, &b);
     printf("main :: nakon uredi_tacno, a=%d, b=%d\n", a, b);

81     return 0;
83 }
```

#### Rešenje 3.28

```
1 /*
   Napisati funkciju koja za boju datu u rgb formatu
3   racuna cmy format po formulama:
   C = 1 - ( R / 255 )
5   M = 1 - ( G / 255 )
   Y = 1 - ( B / 255 )

7
   Napisati program koji učitava boju u rgb formatu,
9   primenjuje odgovarajucu funkciju i ispisuje boju u cmy formatu.

11 */

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

15
   void rgb_to_cmy(float* a, float* b, float* c)
17 {
   /* Zagrade su neophodne jer aritmetickie operacije
19   imaju veci prioritet od operatora dereferenciranja (*).
```

```
21  */
22  *a=1-(*a)/255;
23  *b=1-(*b)/255;
24  *c=1-(*c)/255;
25
26  /*
27  Pomocu return ne mozemo vratiti vise od jedne vrednosti.
28
29  Ceste greske:
30  return a,b,c;      return vraca samo jednu vrednost
31  return a; return b; return c; return ce vratiti samo a
32
33  Zato je neophodno da promenljive ciju vrednost
34  zelimo da promenimo prenesemo preko pokazivaca.
35  */
36 }
37
38 int rgb_korektno(float a)
39 {
40     if(a<0 || a>255)
41         return 0;
42     return 1;
43 }
44
45 int main()
46 {
47     float a,b,c;
48
49     /*
50     Argumenti funkcije rgb_to_cmy su
51     pokazivaci na float. Njima prosledjujemo
52     adrese promenljivih a, b i c.
53     */
54
55     printf("Unesi boju u rgb formatu (vrednosti izmedju 0 i 255:");
56     scanf("%f%f%f",&a,&b,&c);
57
58     if(rgb_korektno(a) && rgb_korektno(b) && rgb_korektno(c))
59         rgb_to_cmy(&a,&b,&c);
60     else
61     {
62         printf("Nekorektan unos\n");
63         return -1;
64     }
65
66     printf("Nakon konverzije: %.2f,%.2f,%.2f\n", a,b,c);
67
68     return 0;
69 }
```

#### Rešenje 3.29

```
2  /*
3     Napisati funkciju koja za dve prave date svojim koeficijentima
4     pravca i slobodnim clanovima odredjuje njihovu tacku preseka.
5     Funkcija treba da vrati 1 ako se prave seku i 0 ako nemaju
6     tacku preseka (ako su paralelne). Napisati glavni program
7     koji ucitava podatke o pravama, poziva napisanu funkciju i
8     ispisuje odgovarajucu poruku.
9 */
10 #include<stdio.h>
11
12 /*
13     Funkcija presek treba da izracuna tri vrednosti:
14     1. indikator da li su koeficijenti pravca jednaki ili ne
15     2. prvu koordinatu presečne tacke (ukoliko prave nisu paralelne)
16     3. drugu koordinatu presečne tacke (ukoliko prave nisu paralelne)
17
18     Indikator funkcija vraca kao povratnu vrednost, preko kljucne reci
19     return.
20
21     Koordinate presečne tacke (ako postoji) funkcija vraca preko
22     liste argumenata, zbog cega promenljive kojima ce koordinate
23     biti dodeljene prenosimo preko pokazivaca (promenljive px i py)
24
25     Promenljive koje sadrze podatke o pravama (k1,n1,k2,n2) se ne
26     menjaju u funkciji i zbog toga ih ne moramo prenositi preko
27     pokazivaca.
28 */
29
30 int presek(float k1, float n1, float k2, float n2, float* px, float*
31            py)
32 {
33     if (k1==k2)
34         return 0;
35
36     *px = -(n1-n2)/(k1-k2);
37     *py = k1*(px)+n1;
38     return 1;
39 }
40
41 int main()
42 {
43     float k1,k2,n1,n2;
44     float x,y;
45
46     printf("Unesi k i n za prvu pravu:");
47     scanf("%f%f",&k1,&n1);
48
49     printf("Unesi k i n za drugu pravu:");
50     scanf("%f%f",&k2,&n2);
```

```

50     if(presek(k1,n1,k2,n2,&x,&y))
52         printf("Prave se seku u tacki (%.2f,%.2f)\n", x,y);
    else
54         printf("Prave su paralelne\n");
56     return 0;
}

```

### Rešenje 3.30

```

1  /*
    Napisati program koji ispisuje broj navedenih argumenata komandne
    linije,
3   a zatim i same argumente i njihove redne brojeve.
    */
5
    #include <stdio.h>
7
    /*
    Argumenti komandne linije cuvaju se u nizu niski pod nazivom
    argv. Svaki element tog niza odgovara jednom argumentu komandne
    linije pri cemu prvi element predstavlja naziv programa koji
    pokrecemo. Celobrojna promenljiva argc predstavlja ukupan
    broj argumenata komandne linije ukljucujuci i argument koji
    odgovara nazivu programa.
13  */
15
17  int main(int argc, char *argv[])
    {
19      int i;

21      printf("Broj argumenata je: %d\n",argc);

23      for(i=0; i<argc; i++)
          printf("%d: %s\n",i,argv[i]);
25
27      return 0;
    }

```

### Rešenje 3.31

```

1  /*
    Napisati funkciju koja za dva data stringa str i
3   accept odredjuje koliko se uzastopnih karaktera stringa str
    nalazi u stringu accept pocev od pocetka niza str. Napisati
    potom program koji testira napisanu funkciju za dva stringa
    koji se unose kao argumenti komandne linije. Primeri upotrebe:
5
7

```

```
1:
9  ./a.out aladin bal
3
11
12 2:
13 ./a.out aladin lad
4
15
16 3:
17 ./a.out Aladin ala
0
19
20 */
21
22 #include <stdio.h>
23 #include <string.h>
24
25 /*
26 Funkcija strspn(str,accept) je ugradjena funkcija koja vraca broj
27 karaktera
28 stringa str koji se nalaze u stringu accept, pocev od pocetka
29 stringa str.
30
31 Funkcija strspn se nalazi u zaglavlju string.h.
32
33 Funkcija strspn_klon je jedna implementacija funkcije strspn.
34
35 U zadacima cemo uvek koristiti ugradjenu funkciju strspn osim ako
36 u tekstu zadatka
37 nije naglaseno da se ona ne sme koristiti. Funkcija strspn_klon
38 sluzi da pokaze na koji
39 nacin radi ugradjena funkcija strspn.
40
41 Ugradjena funkcija strspn poziva se na isti nacin kao funkcija
42 strspn_klon:
43 strspn(s1,s2)
44
45 */
46
47 int strspn_klon(char str[], char accept[])
48 {
49     int br=0;
50     int i;
51
52     for(i=0; str[i];i++)
53         if(strchr(accept, str[i])!=NULL)
54             br++;
55     else /* ako pronadjemo karakter u stringu str koji nije */
56         break; /* u stringu accept, prekidamo petlju */
57
58     return br;
59 }
```



```

55 int main(int argc, char* argv[])
56 {
57     int br;
58
59     if(argc<3)
60     {
61         printf("Nekorektan poziv\nProgram treba pozvati sa ./a.out arg1
62             arg2\n");
63         return -1;
64     }
65
66     br = strstrn_klon(argv[1],argv[2]);
67     printf("Broj karaktera stringa %s koji se nalaze u stringu %s,
68         pocev od pocetka stringa %s: %d\n", argv[1],argv[2],argv[1],br);
69     return 0;
70 }

```

### Rešenje 3.32

```

/*
2  Napisati funkciju void sifruj(char s[], char c, int k) koja
   sifruje
   string s na sledeci nacin: svako malo i veliko slovo stringa s
   konvertuje u
4  slovo koje je u abecedi od njega udaljeno k pozicija, i to
   k pozicija ulevo, ako je karakter c jednak karakteru 'L' ili
   udesno
6  ako je karakter c jednak karakteru 'D'. Sifrovanje treba da bude
   kruzno. Ako string
   s sadrzi karakter koji nije alfanumericki, ostaviti ga
   nesifriranog.
8
10 Napisati potom glavni program koji testira napisanu funkciju za
   string i prirodan
   broj koji se unose kao argumenti komandne linije dok se pravac
   sifrovanja unosi
   kao opcija -p koja moze imati vrednosti 'L' ili 'D'. Ukoliko
   opcija -p nije
12 navedena, podrazumevani pravac je udesno.
14
   Mozemo podrazumevati da string sadrzi najvise 30 karaktera.
16
   Primeri upotrebe:
18
   1:
   ./a.out abcd 2
   cdef
20
   2:
22

```

```
24     ./a.out abcd 2 -p D
      cdef
26
      3:
      ./a.out abcd 2 -p L
28     yzab
30
      4:
      ./a.out abcd -3 -p L
32     Nekorektan unos
34
      5:
      ./a.out abcd 3 -p X
36     Nekorektan unos
38
      6:
      ./a.out ab12cd 2 -p D
40     cd12ef
42 */
44 #include <stdio.h>
45 #include <string.h>
46 #include <stdlib.h>
47 #define MAX 31
48
49 void sifruj(char s[], char c, int k)
50 {
51     int i;
52     int znak;
53     char t;
54
55     /*
56      S obzirom da ce korektnost unosa podataka
57      biti ispitana pre poziva funkcije, promenljiva
58      c ce imati vrednost 'L' ili 'D'.
59
60      Promenljiva znak ima vrednost 1 ili -1
61      i sluzi kao pomocna promenljiva u slucaju
62      da prilikom sifriranja konvertovani
63      karakter izadje iz opsega malih ili velikih slova.
64
65      */
66     znak=1;
67     if (c=='L')
68         znak = -1;
69
70     for(i=0; s[i];i++)
71         if(isalpha(s[i]))
72             {
73
74                 /*
```

```

76         Promenljiva t predstavlja sifrirani karakter s[i].
           Ako je promenljiva t izvan opsega malih ili velikih slova
,
           dodajemo joj ili oduzimamo ukupan broj slova u abecedi
(26),
78         u zavisnosti od pravca sifriranja, kako bismo omogucili
           kruzno sifriranje.
80     */
           t = s[i]+znak*k;
82     if((islower(s[i]) && (t<'a' || t>'z')) || (isupper(s[i]) &&
(t<'A' || t>'Z'))))
           s[i]=t-znak*26;
84     else
           s[i]=t;
86     }
}
88
89 int main(int argc, char* argv[])
90 {
91
92     int k;
93     char pravac;
94     char rec[MAX];
95
96     /*
           Program mozemo pozivati na dva nacina:
98     ./a.out abcd 2
           ili
100    ./a.out abcd 2 -p D
101
102    Zbog toga, broj argumenata moze biti 3 ili 5.
103    */
104
105     if (argc!=3 && argc!=5)
106     {
107         printf("Nekorektan unos: broj argumenata moze biti 3 ili 5\n");
108         return -1;
109     }
110
111     /*
112     Argumenti komandne linije su stringovi. Ako program pokrecemo
           na sledeci nacin:
114     ./a.out abcd 2 -p D
           to znaci da je argument koji odgovara dvojci u stvari
116     string "2". Da bismo string konvertovali u ceo broj,
           koristimo ugradjenu funkciju atoi iz biblioteke stdlib.h.
118     */
119
120     k = atoi(argv[2]);
121
122     /*
           Ispitujemo korektnost datih podataka:

```

### 3 Predstavljanje podataka

---

```
124  */
125  if (k<=0)
126  {
127      printf("Nekorektan unos: broj pozicija mora biti pozitivan ceo
128      broj\n");
129      return -1;
130  }
131
132  /* Korektnost unosa je ispitana, sto znaci da
133  argc moze biti 3 ili 5 */
134
135  if (argc==3) /* Ako je argc 3: */
136      pravac='D';
137  else /* Ako argc nije 3, tada je sigurno 5, jer je */
138  { /* korektnost unosa ispitana, a unos je korektan
139      jedino za argc==3 ili argc==5 */
140      /*
141          Ispitujemo korektnost preposlednjeg argumenta koji mora da
142          bude u formatu "-p".
143          Ovaj argument je string argv[3]. Njegovom prvom karakteru (
144          koji treba
145          da bude '-') pristupamo sa argv[3][0] a drugom sa argv
146          [3][1].
147      */
148      if (argv[3][0] != '-')
149      {
150          printf("Nekorektan unos: pri zadavanju opcija prvi karakter
151          mora biti '-' \n");
152          return -1;
153      }
154
155      if (argv[3][1]!='p')
156      {
157          printf("Nekorektan unos: nedozvoljena opcija\n");
158          return -1;
159      }
160
161      /*
162          Nakon argumenta -p sledi argument koji zadaje vrednost ove
163          opcije. To je
164          poslednji argument kome pristupamo sa argv[4]. Ovaj argument
165          treba
166          da sadrzi samo jedan karakter - 'L' ili 'D' i njemu
167          pristupamo sa
168          argv[4][0].
169      */
170      if (argv[4][0]=='L' || argv[4][0]=='D')
171          pravac=argv[4][0];
172      else
173      {
174          printf("Nekorektan unos: pravac moze biti L ili D\n");
175          return -1;
176      }
```

```
168     }  
169 }  
170 strcpy(rec, argv[1]);  
171 sifruj(rec,pravic,k);  
172 printf("Sifrovana rec: %s\n", rec);  
173  
174 return 0;  
175 }  
176 }
```

Rešenje 3.41

Rešenje 3.41

Rešenje 3.41

Rešenje 3.41

Rešenje 3.41

Rešenje 3.41

Rešenje 3.41

Rešenje 3.41

Rešenje 3.41

Rešenje 3.42

```
2  /*  
3   Napisati funkciju koja konvertuje dati string tako sto  
4   mala slova menja u velika a velika u mala. Napisati  
5   potom glavni program koji učitava string, poziva napisanu  
6   funkciju i ispisuje konvertovani string. Mozemo pretpostaviti  
7   da string ne sadrzi vise od 10 karaktera.  
8   */  
9  
10 #include <stdio.h>  
11 #include <ctype.h>  
12  
13 /*
```

### 3 Predstavljanje podataka

---

```
14      Kada je niz argument funkcije, dodatni argument je obavezno
16      njegova dimenzija. Kod stringova to nije slucaj jer svaki string
18      ima isti poslednji element - terminirajucu nulu - i to je oznaka
16      kraja stringa.
18      */
18      void konvertuj(char s[])
19      {
20          int i;
21
22          for(i=0; s[i]!='\0'; i++)
23              if (s[i]>='a' && s[i]<='z')
24                  s[i] = toupper(s[i]); /* toupper - konvertuje malo slovo u
25                  odgovarajuce veliko */
26              else if (s[i]>='A' && s[i]<='Z')
27                  s[i] = tolower(s[i]); /* tolower - konvertuje veliko slovo
28                  u odgovarajuce malo */
29          /*
30             Funkcije toupper i tolower se nalaze u zaglavlju ctype.h.
31
32             Konverzija malog slova u veliko bez upotrebe funkcije toupper:
33             s[i] = s[i]-'a'+'A';
34             Konverzija velikog slova u malo bez upotrebe funkcije tolower:
35             s[i] = s[i]+'a'-'A';
36
37          */
38      }
39
40      int main()
41      {
42          /*
43             Poslednji karakter svakog stringa je terminirajuca
44             nula '\0', specijalni karakter ciji je ASCII kod 0.
45
46             Ukoliko pretpostavljamo da string sadrzi najvise 30
47             karaktera, neophodno je deklarirati niz od 31 karaktera,
48             pri cemu se dodatni izdvaja za terminirajucu nulu.
49
50          */
51          char s[31];
52          printf("Unesi string:");
53
54          /*
55             Za razliku od nizova koji se ucitavaju i stampaju
56             element po element, stringovi se mogu ucitati i
57             odstampati pomocu jedne scanf/printf naredbe koriscenjem
58             specifikatora %s.
59
60             Funkcija scanf ucitava string do prvog pojavljivanja razmaka.
61          */
62          scanf("%s", s);
63
64          konvertuj(s);
```

```
64     printf("Konvertovani string: %s\n", s);
66     return 0;
68 }
```

### Rešenje 3.43

```
/*
2   Napisati funkciju skрати koja uklanja beline sa
   kraja datog stringa.

4   Napisati glavni program koji testira napisanu
6   funkciju na stringu "rep belina".

8 */

10 #include <stdio.h>
12 #include <ctype.h>

/*
14   Funkcija koja racuna duzinu niza
   ne racunajuci '\0'.

16   U biblioteci string.h definisan je veliki
18   broj funkcija za rad sa stringovima,
   ukljucujuci i funkciju strlen koja racunana
20   duzinu stringa.

22   Funkcija strlen_klon predstavlja jednu
   implementaciju funkcije strlen.

24   U zadacima cemo uvek koristiti ugradjenu
26   funkciju strlen osim ako u tekstu zadatka
   nije naglaseno da se ona ne sme koristiti.
28   Funkcija strlen_klon služi da pokaze na koji
   nacin radi ugradjena funkcija strlen.

30   Ugradjena funkcija strlen poziva se na
32   isti nacin kao funkcija strlen_klon:
   strlen(s1)

34 */
36 int strlen_klon(char s[])
38 {
   int i=0;
   while(s[i]) /* ASCII kod karaktera '\0' je jednak nuli. */
40     i++;

42     return i;
}
```

### 3 Predstavljanje podataka

---

```

}
44
void skрати(char s[])
46
{
    /*
48     Poslednji karakter stringa s (ne racunajuci '\0') ima
        indeks strlen_klon(s)-1; ideja je da pocnemo od poslednjeg
50     karaktera stringa i da smanjujemo indeks dokle god
        je karakter na poziciji i blanko znak.

52     */
    /*
54     int i;
        for(i=strlen_klon(s)-1; i>=0; i--)
56         if (!isspace (s[i])) /* Ako s[i] nije blanko znak, prekidamo
            petlju. */
            break;

58     s[i+1]='\0'; /* Dodajemo terminirajucu nulu iza indeksa i (prvi
        neblanko karakter gledano sdesna nalevo).*/

60     /*
62     Ugradjena funkcija isspace nalazi se u biblioteci ctype.h i
        vraca 1 ako
        je dati karakter blanko znak a 0 u suprotnom.

64     Unarni logicki operator ! oznacava negaciju.

66     */
68 }

70 int main()
{
72     /*
74     Ukoliko string ne zelimo da učitavamo po pokretanju programa
        vec da ga unapred zadamo, to mozemo uraditi na sledeci nacin:

76     */
    char s[]="rep belina";
78     /* U ovom slucaju nije neophodno navoditi dimenziju stringa vec
        ce ona biti automatski postavljena na broj karaktera u stringu +
        1 za
80     terminirajucu nulu. */

82     printf("Pre skracivanja: %s\n", s);
84     skрати(s);
86     printf("Posle skracivanja: %s\n", s);

    return 0;
88 }

```



## Rešenje 3.44

```
2  /*
3  Napisati program koji ucitava string src i formira string dst
4  trostrukim nadovezivanjem stringa src. Program treba da ispise
5  string dst. Na primer, za uneti string "dan", string dst treba
6  da bude "dandandan". Pretpostaviti da string src nije duzi od
7  30 karaktera.
8  */
9
10 #include <stdio.h>
11 #include <string.h>
12
13 #define MAX 30
14 /*
15 Na stringove ne mozemo primeniti naredbu dodele.
16 Ukoliko zelimo da jedan string "dodelimo" drugom,
17 mozemo koristiti ugradjenu funkciju strcpy(s,t)
18 koja kopira karaktere stringa t
19 u string s zajedno za terminirajucom nulom.
20
21 Funkcija strcpy se nalazi u biblioteci string.h.
22
23 Funkcija strcpy_klon predstavlja jednu
24 implementaciju funkcije strcpy.
25
26 Karakteri stringa original se, jedan po jedan,
27 kopiraju u string kopija. Nakon kopiranja,
28 na kraj stringa kopija dodaje se terminalna
29 nula.
30
31 U zadacima cemo uvek koristiti ugradjenu
32 funkciju strcpy osim ako u tekstu zadatka
33 nije naglaseno da se ona ne sme koristiti.
34 Funkcija strcpy_klon sluzi da pokaze na koji
35 nacin radi ugradjena funkcija strcpy.
36
37 Ugradjena funkcija strcpy poziva se na
38 isti nacin kao funkcija strcpy_klon:
39 strcpy(dst,src)
40 gde karaktere stringa src kopiramo
41 u string dst.
42 */
43
44 void strcpy_klon(char kopija[], char original[])
45 {
46     int i;
47     for(i=0; original[i]; i++)
48         kopija[i]=original[i];
49
50     kopija[i] = '\0';
```

### 3 Predstavljanje podataka

---

```
52 }
53
54 int main()
55 {
56     char src[MAX+1];    /* src, skraceno od source (izvor, odnosno sta
                           kopiramo) */
57     char dst[3*MAX+1];  /* dst, skraceno od destination (odrediste,
                           odnosno gde kopiramo) */
58
59     /*
60      Vazno je izdvojiti dovoljno memorijskog prostora
61      za string dst: on treba da bude tri puta veci od
62      maksimalne duzine stringa src + jedan karakter za
63      terminirajucu nulu.
64     */
65
66     printf("Unesi jedan string:");
67     scanf("%s", src);
68
69     strcpy_klon(dst,src);
70
71     /*
72      Funkcija strcat(s,t) nadovezuje karaktere stringa
73      t na kraj stringa s i novi string terminira
74      karakterom '\0' .
75
76      Funkcija strcat se nalazi u biblioteci string.h.
77     */
78     strcat(dst,src);
79     strcat(dst,src);
80
81     printf("Kada nadovezemo string %s triput: %s\n",src,dst);
82
83     return 0;
84 }
```

#### Rešenje 3.45

```
1  /*
2     Napisati funkciju int ucitaj_liniju(char s[], int n)
3     koja ucitava liniju maksimalne duzine n u string s
4     i vraca duzinu učitane linije. Linija moze da sadrzi
5     blanko znakove ali ne moze da sadrzi \n ili EOF.
6
7     Napisati potom glavni program koji ucitava linije
8     do EOF i ispisuje najduzu liniju i njenu duzinu. Ukoliko
9     ima vise linija maksimalne duzine, ispisati prvu. Mozemo
10    pretpostviti da svaka linija sadrzi najvise 80 karaktera,
11    zajedno sa \n.
12
13    */
```

```
15 #include<stdio.h>
16 #include<string.h>
17 #define MAX 81
18
19 /*
20    Ukoliko zelimo da učitamo string koji sadrzi beline
21    (npr liniju teksta), ne mozemo koristiti funkciju
22    scanf jer ona učitava string do prvog blanko znaka.
23
24    Zbog toga je neophodno napisati funkciju koja učitava
25    string karakter po karakter.
26
27    Ova funkcija ne dopusta unosenje vise karaktera od
28    unapred odredjene granice (argument n).
29
30    U standardnoj biblioteci stdio.h postoji definisana
31    funkcija char *gets(char *s) koja učitava karaktere
32    dok se ne pojavi novi red ili EOF. Ova funkcija
33    dopusta unosenje vise karaktera nego sto string
34    s sadrzi, sto moze dovesti do neocekivanog ponasanja
35    programa.
36
37    Pored funkcije gets, koja vrsi učitavanje sa standardnog
38    ulaza, u standardnoj biblioteci stdio.h postoji
39    i ugradjena funkcija fgets koja vrsi učitavanje iz
40    datoteke. Nju cemo koristiti za nekoliko casova
41    kada budemo radili datoteke. Prototim funkcije fgets je
42    ovakav:
43
44    char *fgets(char *s, int size, FILE *stream);
45
46    Argumenti funkcije fgets su:
47    s - string u koji vrsimo učitavanje
48    size - maksimalna duzina unetog stringa
49    stream - datoteka iz koje vrsimo učitavanje
50
51    Funkcija fgets, za razliku od funkcije gets, ne dopusta
52    unos vise karaktera od date vrednosti size. Zbog toga
53    je ona sigurnija nego funkcija gets. Funkciju fgets
54    mozemo koristiti i za unos sa standardnog ulaza
55    ukoliko kao treci argument navedemo stdin.
56
57 */
58 int ucitaj_liniju(char s[], int n)
59 {
60     int i=0;
61     int c;
62
63     while((c=getchar())!='\n' && i<n-2 && c!=EOF)
64     {
65         s[i] = c;
```

### 3 Predstavljanje podataka

```
        i++;
67    }

69    /* Ucitavamo najvise n-2 karaktera jer na kraju dodajemo jos
        dva: '\n' i '\0' */
71
73    s[i]='\n';
75    s[i+1]='\0';
77
79    return i;
81 }

83 int main()
85 {
87     char linija[MAX];
89     char najduza_linija[MAX];
91     int max_duzina=0;
93     int duzina;
95
97     /*
99     Petlja se zavrsava ukoliko je promenljiva duzina
101     jednaka nuli, sto cemo postici zadavanjem linije koja ne sadrzi
103     nijedan karakter osim EOF.
105     */
107     while ((duzina=ucitaj_liniju(linija, MAX))>0)
109     {
111         /*
113         Proveravamo da li je uneta linija duza od trenutnog
115         maksimuma i azuriramo promenljive max_duzina i najduza_linija
117         */
119         if (max_duzina<duzina)
121         {
123             max_duzina = duzina;
125             strcpy(najduza_linija,linija);
127         }
129     }

131     printf("Najduza linija: %s duzine: %d\n", najduza_linija,
133           max_duzina);

135     return 0;
137 }
```

#### Rešenje 3.46

```
2  /*
    Napisati program koji pretvara nisku u ceo broj.
    Npr. za ulaz "-1238" se generise rezultat -1238
    */
```

```
4   Pogledati funkcije atoi i atof koje postoje u biblioteci stdlib.h
   */
6
8   #include <stdio.h>
   #include <ctype.h>
   #define MAX 10
10  /*
   String b se sastoji od karaktera koji
12  cine jedan ceo broj, onim redom kojim
   se karakteri pojavljuju u zapisu broja.
14
   Ako je prvi karakter stringa b '-',
16  to znaci da je broj negativan i
   funkcija znak_broja vraca -1
18
   U suprotnom, broj je pozitivan i
20  funkcija znak_broja vraca 1
22  */
24  int znak_broja(char b[])
   {
26      if(b[0]=='-')
           return -1;
28      return 1;
   }
30
32  /*
   Funkcija formiraj_broj na osnovu
34  karaktera koji cine broj iz stringa
   b vraca ceo broj koji odgovara
36  zapisu datom u stringu b.
38
   Ako su cifre broja a,b,c i d, tada
   broj mozemo kreirati kao:
40   $a \cdot 10^3 + b \cdot 10^2 + c \cdot 10^1 + d \cdot 10^0$ 
42
   Medjutim, efikasnije je koristiti
   Hornerovu semu:
44
    $10 \cdot (10 \cdot (10 \cdot (10 \cdot 0 + a) + b) + c) + d$ 
46
   */
48
50  int formiraj_broj(char b[])
   {
       int i;
52      int n=0;
       int znak = znak_broja(b);
54
       /*
```

### 3 Predstavljanje podataka

---

```
56     Ako je broj negativan, cifre u nizu b
    pocinju od indeksa 1
58 */
60     i=0;
    if(znak== -1)
62         i=1;
64
    /*
    Funkcija isdigit proverava da li je broj
66     cifra. Nalazi se u biblioteci ctype.h
68
    Proveravamo da li je karakter u zapisu
    broja cifra kako bismo se osigurali
70     od nekorektnog unosa, npr ako korisnik
    unese -123abc. Ovaj unos je moguc jer
72     se vrsi sa scanf("%s",broj), gde unosimo
    karaktere do prvog blanko znaka
74
    Ako naidjemo na karakter koji nije cifra,
76     prekidamo petlju
78
    */
    for(; b[i]!='\0'; i++)
80        if(isdigit(b[i]))
            n = n*10 + b[i] - '0';
82        else
            break;
84
    /* Formirani broj mnozimo znakom: */
86
    n*=znak;
88    return n;
90 }
92 int main()
{
94     char broj[MAX];
    int n;
96
    /* Ucitavamo broj: */
98     scanf("%s", broj);
100
    /* Ispisujemo rezultat: */
    printf("Broj zapisan kao int: %d\n", formiraj_broj(broj));
102
    return 0;
104 }
```

#### Rešenje 3.47

```
2  /*
   Napisati program koji pretvara zadatu broj u nisku.
   Npr. za broj -453 treba generisati nisku "-453"
4  */

6  #include <stdio.h>
   #include <string.h>
8  #define MAX 10
   /*

10     Funkcija transformisi_negativan vraca
12     1 ako je broj negativan i 0 u suprotnom, a
   uz to, ako broj jeste negativan, funkcija
14     treba da ga konvertuje u njegovu apsolutnu
   vrednost. S obzirom da funkcija treba da vrati dve
16     vrednosti, to realizujemo na sledeci nacin:
   1. indikator da li je broj negativan
18     ce vratiti kao povratnu vrednost
   2. apsolutnu vrednost broja ce vratiti
20     preko liste argumenata, zbog cega broj
   prenosimo preko pokazivaca
22
   */
24  int transformisi_negativan(int* pn)
   {
26     if(*pn<0)
       {
28         *pn = -(*pn);
         return 1;
30     }
     return 0;
32  }

34  int formiraj_niz_cifara(int n, char b[], int neg)
   {
36     int i=0;
     char cifra;

38     do
40     {
         cifra = n%10;

42
         /* Promenljiva b predstavlja string.
44         Da bismo na neku poziciju u stringu
         upisali karakter koji odgovara nekoj
46         cifri, npr '2', neophodno je da
         odgovarajucoj poziciji dodelimo vrednost
48         ASCII koda te cifre, konkretno za '2'
         ASCII kod je '0'+2.

50         Greska bi bila navesti b[i]=2
```

### 3 Predstavljanje podataka

---

```
52         jer 2 nije ASCII kod koji odgovara karakteru
           '2'.
54     */
           b[i]=cifra+'0';
56
           n/=10;
58           i++;
           } while(n);
60
           /* Ako je broj negativan, dodajemo znak minus: */
62           if(neg)
           {
64               b[i]='-';
               i++;
66           }

           /* Svaki string se završava terminirajućom nulom: */
           b[i]='\0';
70     }

72 void obrni(char s[])
       {
74
           char t;
           int i,j;
           /*
78             Karaktere stringa obrćemo tako što razmenimo karaktere na
               pozicijama 0 i n-1,
               zatim 1 i n-2, 2 i n-3 i tako redom dok je prva pozicija manja od
               druge
80             */

           for(i=0,j=strlen(s)-1;i<j;i++, j--)
           {
84               t = s[i];
               s[i] = s[j];
86               s[j] = t;
           }
88     }

90 void broj_u_niz_cifara(int n, char broj[])
       {
92     int negativan;

           /* Odredjujemo znak broja: */
           negativan=transformisi_negativan(&n);

           /* Izdvajamo cifre broja i smestamo ih u niz: */
           formiraj_niz_cifara(n, broj, negativan);
100
           /* S obzirom da cifre izdvajamo sa kraja broja, u nizu ce biti u
```



```

    obrnutom redosledu.
102     Na primer, za broj 234 niz ce sadrzati cifre 4 3 2. */
    obrni(broj);
104 }

106 int main()
{
108     int n;
    char broj[MAX];
110     int negativan;

112     /* Ucitavamo broj: */
    scanf("%d", &n);
114

    /* Kreiramo broj na osnovu niza cifara: */
116     broj_u_niz_cifara(n,broj);

118     /* Ispisujemo rezultat: */
    printf("Broj zapisan kao string: %s\n", broj);
120

    return 0;
122 }

```

### Rešenje 3.48

```

2  /*
    Napisati program koji ucitava dva stringa i ispituje najpre da li
    su jednaki. Ako jesu, program
    treba da izda odgovarajucu poruku, a ako nisu, treba da ispita da
    li je drugi podstring
4  prvog. Ukoliko jeste, program treba da ispise pocev od kog indeksa
    prvog
    stringa pocinje drugi string. U suprotnom, ispisati odgovarajucu
    poruku. Mozemo
6  pretpostaviti da stringovi ne sadrze vise od 20 karaktera.
    */
8
    #include <stdio.h>
10  #include <string.h>

12  /*
    Funkcija strcmp(s,t) je ugradjena funkcija koja utvrdjuje da
    li su stringovi
14  s i t jednaki. Ukoliko jesu, vraca 0, a u suprotnom vraca
    razliku
    ASCII kodova prva dva razlicita karaktera na istim pozicijama
16  (npr strcmp("aa","ab") ce vratiti -1 a strcmp("ab","aa") 1).

    Funkcija strcmp se nalazi u zaglavlju string.h.
18

    Funkcija strcmp_klon je jedna implementacija funkcije strcmp.
20

```

### 3 Predstavljanje podataka

---

```
22      U zadacima cemo uvek koristiti ugradjenu funkciju strcmp osim
      ako u tekstu zadatka
      nije naglaseno da se ona ne sme koristiti. Funkcija
      strcmp_klon služi da pokazuje na koji
24      način radi ugradjena funkcija strcmp.

26      Ugradjena funkcija strcmp poziva se na isti način kao funkcija
      strcmp_klon:
      strcmp(s1,s2)
28      gde poredimo stringove s1 i s2.

30  */

32  int strcmp_klon(char s1[], char s2[])
  {
34      int i;
      for(i=0; s1[i]==s2[i];i++)
36          if (s1[i]=='\0')
              return 0;

38      return s1[i] - s2[i];
40  }

42  int main()
  {
44      char s1[21];
      char s2[21];
46      char* p;

48      printf("Unesi dva stringa:");
      scanf("%s%s",s1,s2);
50

52      /*
      Funkcija strstr(s,t) je ugradjena funkcija koja utvrđuje da
      li je string t
      podstring stringa t i ako jeste, vraća pokazivac (char*) na
      karakter
54      stringa s odakle počinje prvo pojavljivanje stringa t, a NULL
      u suprotnom.

56      NULL je pokazivac koji ne pokazuje ni na šta, odnosno ne
      sadrži adresu
      nijedne promenljive.

58

      Podsetimo se veze nizova(a time i stringova) i pokazivaca:
      ako je string deklarisan sa s1[21], tada je njegov naziv s1
      ekvivalentan adresi prvog karaktera stringa:
60      s1 <=> &s1[0]
      i nadalje redom:
62      s1+1 <=> &s1[1]
64      ...

```

```

66     u opstem slucaju:
        s1+i <=> &s1[i]
68
        To znaci da se indeks elementa na koji pokazuje s1+i moze
70        dobiti tako sto od s1+i oduzmemo pokazivac na pocetak niza:
        s1+i-s1 <=> i. Ovako od pokazivaca na karakter u stringu
72        dobijamo njegov indeks u stringu.
74
        */
76     p = strstr(s1,s2);
78     if (strcmp_klon(s1,s2)==0)
        printf("Uneti stringovi su jednaki\n");
80     else if (p!=NULL)
        printf("%s jeste podstring od %s pocev od pozicije : %d\n", s2,
            s1, p-s1);
82     else
        printf("%s NIJE podstring od %s\n", s2,s1);
84
        return 0;
86 }

```

### Rešenje 3.49

```

/*
2   Napisati program koji za uneti string s i karakter c utvrđuje
   da li se c pojavljuje u stringu s i ukoliko se pojavljuje,
4   ispisuje indeks prvog pojavljivanja a u suprotnom ispisuje
   odgovarajucu poruku. Mozemo pretpostaviti da string ima najviše
6   20 karaktera.
   */
8
#include <stdio.h>
10 #include <string.h>
12
int main()
{
14     char s[21];
        char c;
16     char* p;

18     printf("Unesi karakter:");
        c=getchar();
20     printf("Unesi string:");
        scanf("%s", s);
22
        /*
24         Da smo ucitavali obrnutim redom (prvo string pa karakter)
            to bismo realizovali na sledeci nacin:
26         printf("Unesi string:");

```

### 3 Predstavljanje podataka

---

```
scanf("%s",s);
28 getchar();
printf("Unesi karakter:");
30 c=getchar();

32 Dodatni getchar() bi sluzio da "pokupi" karakter kojim
razdvajamo unos stringa i karaktera (razmak, novi red ili
34 slicno).

36 */
37
38 /*
Funkcija strchr(s,c) je ugradjena funkcija koja vraca pokazivac
40 na prvi karakter u stringu s koji je jednak karakteru c, ako
takav
postoji, a NULL u suprotnom.
42
Indeks od pokazivaca dobijamo na isti nacin kao u prethodnom
zadatku
44 sa strstr.
*/
46
47 p = strchr(s,c);
48 if(p!=NULL)
printf("%c se pojavljuje u %s na poziciji %d\n", c, s, p-s);
50 else
printf("%c se NE pojavljuje u %s\n",c, s);
52
return 0;
54 }
```

Rešenje 3.58

Rešenje 3.58

Rešenje 3.58

Rešenje 3.58

Rešenje 3.58

Rešenje 3.58

Rešenje 3.58

## Rešenje 3.58

## Rešenje 3.58

## Rešenje 3.59

```

2  /*
4  Data je struktura koja opisuje koordinate
   tacke u ravni:

6  typedef struct point
   {
8      float x;
      float y;
10 } POINT;

12 U glavnom programu date su dve tacke: tacka
   A sa fiksiranim koordinatama (1,2) i tacka B
14 cije koordinate zadaje korisnik. Napisati
   funkcije:
16 a) za racunanje rastojanja izmedju dve date tacke
   b) za odredjivanje tacke koja se nalazi na
18 sredini duzi odredjene dvema datim tackama

20 Testirati napisane funkcije u glavnom programu.

22 */

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

26 typedef struct point
28 {
   float x;
   float y;
30 } POINT;

32 /*
34 Poljima strukture pristupamo pomocu
   operatora .

   Ako je promenljiva a tipa POINT,
38 njenim koordinatama pristupamo
   pomocu a.x i a.y
40 */

42 float rastojanje (POINT a, POINT b)
   {
44     return sqrt(pow(a.x-b.x,2)+pow(a.y-b.y,2));

```

### 3 Predstavljanje podataka

```

46 }
47
48 POINT sredina (POINT a, POINT b)
49 {
50     POINT s;
51     s.x = (a.x+b.x)/2;
52     s.y = (a.y+b.y)/2;
53     return s;
54 }
55
56 int main()
57 {
58     POINT a = {1,2};
59     POINT b;
60     POINT sredina_a_b;
61
62     /* Ispisujemo koordinate tacke a. */
63     printf("Tacka a ima koordinate %.2f,%.2f\n", a.x, a.y);
64
65     /* Ucitavamo koordinate tacke b. */
66     printf("Unesi prvu koordinatu tacke: ");
67     scanf("%f", &b.x);
68     printf("Unesi drugu koordinatu tacke: ");
69     scanf("%f", &b.y);
70     printf("Tacka b ima koordinate %.2f,%.2f\n", b.x, b.y);
71
72     /* Strukture kao argumenti funkcije - prenos po vrednosti. */
73     printf("Rastojanje izmedju tacaka a i b je %.2f\n", rastojanje(a,b)
74 );
75
76     /* Struktura kao povratna vrednost funkcije. */
77     sredina_a_b=sredina(a,b);
78     printf("Tacka na sredini izmedju tacaka a i b je %.2f,%.2f\n",
79         sredina_a_b.x, sredina_a_b.y);
80
81     return 0;
82 }
```

#### Rešenje 3.60

```

1  /*
2     Data je struktura
3     typedef struct Student
4     {
5         char ime[MAX];
6         char prezime[MAX];
7         char smer;
8         float prosek;
9     } STUDENT;
```

```

11 I   Napisati funkciju koja ucitava sa standardnog ulaza podatke o
      studentu. Mozemo pretpostaviti da
      ime i prezime studenta ne sadrže više od 30 karaktera.
13 II  Napisati funkciju koja ispisuje podatke o studentu na standardni
      izlaz.
      III Ucitati niz od n studenata i :
15         a) ispisati imena i prezimena onih koji su na smeru R
            b) ispisati podatke za studenta sa najvećim prosekom; ako ima
               više takvih studenata, ispisati
17                 1) sve njih
                   2) prvog
19                 3) poslednjeg
      */
21
22 #include <stdio.h>
23 #define MAXST 100
24 #define MAX 31
25
26 typedef struct Student
27 {
28     char ime[MAX];
29     char prezime[MAX];
30     char smer;
31     float prosek;
32 } STUDENT;
33
34 /*
35 I
36
37 Ako je dat pokazivac na strukturnu promenljivu s,
38 poljima ove strukture pristupamo sa
39 (*s).ime, (*s).prezime, itd.
40
41 Zagrade su neophodne zbog prioriteta operatora:
42 operator * ima veći prioritet nego operator . .
43
44 Operator -> pruža skraćeni zapis za prethodno
45 navedeni pristup poljima:
46 s->ime je skraćeno za (*s).ime
47 s->prezime je skraćeno za (*s).prezime
48 itd.
49
50 */
51 void ucitaj(STUDENT* s)
52 {
53     /* printf("Ime:"); */
54     scanf("%s", s->ime);
55     /* printf("Prezime:"); */
56     scanf("%s", s->prezime);
57     getchar();
58     /* printf("Smer:"); */

```

### 3 Predstavljanje podataka

---

```
59  scanf("%c",&s->smer);
    /* printf("Prosek:");*/
61  scanf("%f", &s->prosek);
    }
63
64  /* II */
65  /*
    Kada neku promenljivu prenosimo u funkciju kao argument, obicno
67  je prenosimo po vrednosti (bez pokazivaca), ako se ona nece menjati
    u funkciji
    ili po adresi (preko pokazivaca), ako ce se njena vrednost
    promeniti u funkciji.
69
    Prilikom poziva funkcije, za svaki argument funkcije kreira se
    promenljiva
71  koja predstavlja lokalnu kopiju argumenta i koja prestaje da
    postoji po zavrsetku
    funkcije. S obzirom da se strukture sastoje od vise polja,
    zauzimaju
73  vise memorije nego nestrukturne promenljive. Zbog toga je za
    njihovo kopiranje
    potrebno vise vremena i vise memorijskih resursa nego za kopiranje
    nestrukturnih
75  promenljivih.
77
    Da bismo ucinili program efikasnijim, korisno je da strukturu uvek
    kao
    argument funkcije prenosimo po adresi (preko pokazivaca), bez
    obzira
79  da li ce se struktura u toj funkciji menjati ili ne. Pokazivac na
    strukturu
    zauzima manje memorije nego sama struktura pa je izrada njegove
    kopije
81  brza a kopija pokazivaca uzima manji memorijski prostor nego kopija
    strukture.
83
    Kada prenosimo strukturnu promenljivu u funkciju po adresi (preko
    pokazivaca), tada
85  imamo mogucnost da je u funkciji menjamo. Ukoliko zelimo da
    onemogucimo promenu,
    uz argument dodajemo kljucnu rec const. Ako pokusamo da promenimo
    argument
87  funkcije prenesen kao const (npr u funkciji ispisi navedemo naredbu
    s->smer='X');),
    kompajler ce prijaviti gresku. Na ovaj nacin obezbedjujemo da
    promenljiva
89  koju smo preneli po adresi ne da bismo je promenili vec radi
    povecanja efikasnosti programa,
    ne bude, cak ni slucajno, izmenjena u funkciji.
91
    */
93
```



```
void ispisi(const STUDENT* s)
95 {
    printf("%s %s, %c, %.2f\n", s->ime, s->prezime, s->smer, s->prosek);
97 }

99 float najveci_prosek(STUDENT studenti[], int n)
101 {
    float m;
103     int i;
    /* Pretstavimo da student sa indeksom 0 ima
105     maksimalni prosek. */
    m = studenti[0].prosek;
107     for(i=1; i<n; i++)
        if(m<studenti[i].prosek) /* Ako student sa indeksom i ima veci
109         prosek od maksimalnog, */
            m=studenti[i].prosek; /* menjamo maksimalni prosek */
    return m;
111 }

113 /*
    Struktura moze da bude povratna vrednost funkcije.
115 */
STUDENT prvi_student_sa_najvecim_prosekom(STUDENT studenti[], int n,
    float m)
117 {
    STUDENT s;
119     int i;
    for(i=0; i<n; i++)
121         if(m==studenti[i].prosek) /* Ako naidjemo na studenta sa
            maksimalnim prosekom, prekidamo petlju. */
            {
123                 /*
                    Na strukturu se moze primenjivati
125                 naredba dodele.
                */
                s = studenti[i];
127                 break;
            }
129     return s;
131 }

133 /*
    Strukturu mozemo preneti u funkciju preko pokazivaca. Strukture se
    obavezno
135     prenose preko pokazivaca ukoliko je neophodno promeniti vrednosti
    njihovih
    polja u funkciji.
137 */
void poslednji_student_sa_najvecim_prosekom(STUDENT studenti[], int n
    , float m, STUDENT* s)
139 {
```

### 3 Predstavljanje podataka

---

```
141     int i;
142     for(i=0;i<n;i++)
143         if(m==studenti[i].prosek)
144             *s = studenti[i];
145 }
146
147 /*
148     Napomena: funkcije
149     1)prvi_student_sa_najvecim_prosekom
150     2)poslednji_student_sa_najvecim_prosekom
151     odredjuju studenta sa najvecim prosekom po određenom kriterijumu.
152     Funkcija su realizovane na razlicite nacine kako bi ilustrovale:
153     - strukturu kao povratnu vrednost
154     - prenos strukture preko pokazivaca u funkciju, s obzirom da ce se
155       promeniti u funkciji
156
157     Prilikom izrade zadataka moze biti izabran bilo koji od opisanih
158     nacina rada, osim
159     ako neki nacin nije posebno naglasen u tekstu zadatka.
160 */
161 int main()
162 {
163     STUDENT studenti[MAXST];
164     int n;
165     int i;
166     float max_prosek;
167     STUDENT student_sa_max_prosekom;
168     int indeks;
169
170     /* printf("Unesi broj studenata:"); */
171     scanf("%d", &n);
172
173     if (n<0 || n>MAXST)
174     {
175         printf("Nekorektan unos\n");
176         return -1;
177     }
178
179     /* printf("Unesi podatke o studentima:"); */
180     for(i=0;i<n;i++)
181     {
182         /* printf("%d. student:\n", i); */
183         ucitaj(&studenti[i]);
184     }
185
186     printf("Studenti sa R smerom:\n");
187     for(i=0;i<n;i++)
188     {
189         if(studenti[i].smer == 'R')
190             ispisi(&studenti[i]);
191     }
192     printf("-----\n");
```

```

191  /* b)1)
193      Stampamo podatke o svim studentima sa
195      maksimalnim prosekom.
196  */
197
198  max_prosek = najveći_prosek(studenti, n);
199  printf("Svi studenti koji imaju maksimalni prosek:");
200  for(i=0;i<n;i++)
201      if(studenti[i].prosek==max_prosek)
202          ispisi(&studenti[i]);
203
204  /* b)2) */
205  student_sa_max_prosekom = prvi_student_sa_najvecim_prosekom(
206      studenti,n,max_prosek);
207
208  printf("Prvi student u nizu sa najvećim prosekom: ");
209  ispisi(&student_sa_max_prosekom);
210
211  /* b)3) */
212  poslednji_student_sa_najvecim_prosekom(studenti,n,max_prosek,&
213      student_sa_max_prosekom);
214
215  printf("Poslednji student u nizu sa najvećim prosekom: ");
216  ispisi(&student_sa_max_prosekom);
217
218  return 0;
219 }

```

### Rešenje 3.61

```

1  /*
3  Napisati program koji učitava reci sa standardnog ulaza dok korisnik
   ne zada EOF i ispisuje
   ih na standardni izlaz svaku u posebnom redu, poravnatu udesno u
   odnosu
5  na poslednji karakter najduže reci. Koristiti
   strukturu typedef struct rec
7      {
           char s[21];
           int duzina;
           }REC;
11  Na primer, ako su unesene sledeće reci:
   Danas imamo ispit iz programiranja1.
13  Nadam se da neće biti tesko!
   onda ispis izgleda ovako:
15      Danas
        imamo

```

### 3 Predstavljanje podataka

---

```
17         ispit
18         iz
19 programiranja1.
20         Nadam
21         se
22         da
23         nece
24         biti
25         tesko!

27 Program realizovati kroz sledece funkcije:
28 a) Funkciju za ucitavanje jedne reci u strukturu REC.
29 b) Funkciju za ucitavanje niza struktura koja vraca dimenziju niza
30 c) Funkciju koja odredjuje maksimalnu duzinu reci u datom nizu
31 d) Funkciju koja ispisuje reci u trazenom formatu

33 Mozemo pretpostaviti da nijedna rec ne sadrzi vise od 30 karaktera i
34     da nece biti
35     uneto vise od 1000 reci.

36
37 /*
38
39 #include<stdio.h>
40 #include<string.h>
41 #define MAXRECI 100
42 #define MAX 31
43
44 typedef struct rec
45 {
46     char s[MAX];
47     int duzina;
48 }REC;
49
50 void ucitaj_rec(REC* rec)
51 {
52     scanf("%s", rec->s);
53     rec->duzina = strlen(rec->s);
54 }
55
56 /*
57     U funkciji ucitaj_niz_reci argument n oznacava broj
58     elemenata niza reci, koji ce biti poznat tek po
59     zavrsetku funkcije. Ova promenljiva ce dobiti svoju
60     vrednost u funkciji i zbog toga mora biti prenesena
61     preko pokazivaca.
62
63 */
64
65 void ucitaj_niz_reci(REC reci[], int* pn, int granica)
66 {
67     int i=0;
```

```

69     do
70     {
71         ucitaj_rec(&reci[i]);
72         i++;
73     }
74     while(reci[i-1].duzina>0 && (i-1)<granica);
75
76     /*
77      S obzirom da se promenljiva i ucitava
78      pre ispitivanja uslova, uslov ispitujemo
79      za rec sa indeksom i-1
80     */
81
82     *pn = i-1;
83
84     /*
85      S obzirom da se vrednost promenljive i
86      ucitava i kada je unesen EOF, dimenzija
87      niza odgovarace vrednosti i-1
88     */
89 }
90
91 int max_duzina(REC reci[], int n)
92 {
93     int najveca_duzina;
94     int i;
95
96     /*
97      Najvecu duzinu inicijalizujemo na duzinu
98      prve reci.
99     */
100    najveca_duzina = reci[0].duzina;
101
102    for(i=1;i<n;i++)
103        if(reci[i].duzina>najveca_duzina) /* Ukoliko u nizu naidjemo
104            na rec duzine vece od najvece duzine, */
105            najveca_duzina = reci[i].duzina; /* menjamo vrednost
106                promenljive najveca_duzina. */
107
108    return najveca_duzina;
109 }
110
111 /*
112 Da bismo realizovali ispis u trazenom formatu, pre
113 svake reci ispisujemo onoliko razmaka koliko iznosi
114 razlika maksimalne duzine i duzine date reci.
115 */
116
117 void ispis(REC reci[], int n, int max_d)
118 {
119     int i,j;

```

```
119     for(i=0;i<n;i++)
120     {
121         for(j=0;j<max_d-recv[i].duzina;j++)
122             printf(" ");
123         printf("%s\n", recv[i].s);
124     }
125 }
126
127 int main(int argc, char* argv[])
128 {
129     REC recv[MAXRECI];
130     int najveca_duzina;
131     int n;
132
133     ucitaj_niz_recv(recv, &n, MAXRECI);
134     najveca_duzina = max_duzina(recv,n);
135     ispis(recv, n, najveca_duzina);
136
137     return 0;
138 }
```

#### Rešenje 3.62

```
/*
2   Napisati program koji izracunava prosečnu cenu jedne potrosacke
3   korpe. Potrosacka korpa se sastoji od broja kupljenih artikala i
4   niza kupljenih artikala. Svaki artikal odredjen je svojim nazivom,
5   kolicinom i cenom. Program treba da ucita broj potrosaca n (
6   najvise 100),
7   zatim podatke za n potrosackih korpi i da na osnovu ucitanih
8   podataka
9   izracuna prosečnu cenu potrosacke korpe. Ucitavanje se vrši sa
10  standardnog
11  ulaza pri čemu se prvo zada broj artikala, a zatim za svaki
12  artikal naziv,
13  kolicina i cena.  Možemo pretpostaviti da nijedan
14  potrosac neće kupiti više od 20 artikala, kao i da naziv svakog
15  artikla
16  sadrži maksimalno 30 karaktera.
17
18  */
19
20 #include <stdio.h>
21 #define MAXART 20
22 #define MAXPOT 100
23 #define MAXNAZIV 31
24
25 typedef struct artikal
26 {
27     char naziv[MAXNAZIV];
```

```
24     int kolicina;
25     float cena;
26 } ARTIKAL;
27
28 typedef struct korpa
29 {
30     int br_art;
31     ARTIKAL artikli[MAXART];
32 } KORPA;
33
34 /*
35 Funkcija ucitaj_artikal ucitava podatke za jedan
36 artikal i vraca 1 ako je ucitavanje bilo uspesno
37 a 0 u suprotnom. Ucitavanje je neuspesno ukoliko
38 kolicina nekog artikla ili njegova cena nisu pozitivni
39 brojevi.
40
41 S obzirom da funkcija ucitaj_artikal treba da vrati
42 dve vrednosti (ucitanu strukturu i indikator uspesnosti),
43 strukturu ARTIKAL prenosimo preko pokazivaca a
44 indikator uspesnosti vracamo kao povratnu vrednost.
45
46 */
47 int ucitaj_artikal(ARTIKAL* a)
48 {
49     scanf("%s", a->naziv);
50     scanf("%d", &a->kolicina);
51
52     if (a->kolicina<=0)
53     {
54         printf("Nekorektan unos za kolicinu artikla: %d\n", a->kolicina);
55         return 0;
56     }
57
58     scanf("%f",&a->cena);
59     if (a->cena<0)
60     {
61         printf("Nekorektan unos za cenu artikla: %f\n", a->cena);
62         return 0;
63     }
64
65     return 1;
66 }
67
68 /*
69 Funkcija izracunaj_racun izracunava racun date
70 potrosacke korpe u kojoj su inicijalizovani
71 podaci o broju artikala i o svakom pojedinacnom
72 artiklu.
```

```
74  */
75  float izracunaj_racun(const KORPA* k)
76  {
77      int i;
78      float racun=0;
79      for(i=0;i<k->br_art;i++)
80          racun+=k->artikli[i].kolicina * k->artikli[i].cena;
81      return racun;
82  }

83  /*
84   Pri učitavanju korpe, zadaje se broj artikala a zatim
85   podaci za svaki artikal.

86   Funkcija učitaj_korpu vraća 1 ako je učitavanje uspesno
87   i 0 u suprotnom. Do neuspesnog učitavanja može doći
88   ako broj artikala u korpi nije pozitivan ili ako dodje
89   do neuspesnog učitavanja nekog artikla.
90   */
91  */
92  int učitaj_korpu(KORPA* k)
93  {
94      int i;
95      scanf("%d", &k->br_art);
96      if (k->br_art<=0)
97      {
98          printf("Nekorektan unos za broj artikala: %d\n", k->br_art);
99          return 0;
100      }
101      for(i=0; i<k->br_art;i++)
102          if (učitaj_artikal(&k->artikli[i])==0)
103              return 0;
104      return 1;
105  }

106  /*
107   Funkcija učitaj_niz_korpi učitava podatke
108   za niz od n potrosackih korpi. Funkcija
109   vraća 1 ako je učitavanje uspesno i 0 ako
110   nije. Učitavanje je neuspesno ukoliko ne uspe
111   učitavanje jedne od korpi.
112   */
113  */
114  int učitaj_niz_korpi(KORPA korpe[], int n)
115  {
116      int i,j;
117      for(i=0; i<n; i++)
118          if(učitaj_korpu(&korpe[i])==0)
119              return 0;
120      return 1;
121  }
```



```
126 }
128
129 /*
130 Funkcija stampaj_racun ispisuje na
131 standardni izlaz racun za datu korpu
132 tako sto za svaki artikal ispise
133 naziv, cenu i kolicinu i na kraju
134 ukupnu cenu za kupljene artikle.
135 */
136 void stampaj_racun(const KORPA* k)
137 {
138     int i,j;
139     for(i=0;i<k->br_art;i++)
140         printf("\t%s %d %.2f\n", k->artikli[i].naziv, k->artikli[i].
141             kolicina, k->artikli[i].cena);
142     printf("-----\n");
143     printf("\tukupno: %.2f\n", izracunaj_racun(k));
144 }
145
146 /*
147 Funkcija stampaj_racune_za_korpe
148 ispisuje na standardni izlaz racune
149 za svaku korpu u nizu potrosackih
150 korpi
151 */
152
153 void stampaj_racune_za_korpe(KORPA korpe[], int n)
154 {
155     int i;
156     for (i=0;i<n;i++)
157     {
158         printf("\nKorpa %d:\n",i);
159         stampaj_racun(&korpe[i]);
160     }
161 }
162
163 /*
164 Funkcija prosek racuna prosečnu cenu
165 potrosacke korpe za dati niz potrosackih
166 korpi
167 */
168 float prosek(KORPA korpe[], int n)
169 {
170     int i;
171     float p;
172
173     for(i=0;i<n;i++)
174         p+=izracunaj_racun(&korpe[i]);
175
176 }
```

### 3 Predstavljanje podataka

```
178     return p/n;
180 }
182 int main()
183 {
184     int n;
185     KORPA korpe[MAXPOT];
186
187     printf("Unesi broj potrosackih korpi:");
188     scanf("%d", &n);
189
190     if(n<0 || n>MAXPOT)
191     {
192         printf("Nekorektan unos broja potrosackih korpi: %d\n",n);
193         return -1;
194     }
195
196     if (ucitaj_niz_korpi(korpe, n)==0)
197         return -1;
198
199     stampaj_racune_za_korpe(korpe,n);
200     printf("Prosecna cena potrosacke korpe: %.2f\n", prosek(korpe, n))
201     ;
202
203     return 0;
204 }
```

#### Rešenje 3.63

```
1  /*
2   Struktura IZRAZ opisuje numericki izraz nad celim brojevima koji
3   se sastoji
4   od dva celobrojna operanda, numericke operacije nad celim
5   brojevima i
6   vrednosti izraza:
7
8   typedef struct izraz
9   {
10     char o;
11     int x;
12     int y;
13     } IZRAZ;
14
15     a) Napisati funkciju koja ispituje da li je dati izraz korektno
16     zadat i vraca 1 ako jeste a 0 u suprotnom. Podrazumevamo da je
17     izraz korektno zadat ako operacija odgovara +,-,* ili / i u
18     slucaju
19     deljenja drugi operand je razlicit od 0.
20
21     b) Napisati funkciju koja za dati izraz odredjuje vrednost izraza.
```

```

21 c) Napisati funkciju koja ucitava dati izraz. Funkcija
    treba da ucita sa standardnog ulaza operaciju i dva
23 operanda u polja o, x i y strukture IZRAZ. Funkcija vraca
    1 ako je ucitavanje bilo uspesno, tj. ako je izraz bio
    korektno zadat ili 0 u suprotnom.

25
    d) Napisati funkciju koja stampa dati izraz infiksno, u obliku
27 x o y = vr. Na primer, za izraz + 4 17 ispis treba
    da bude 4+17=21

29
    e) Napisati glavni program koji ucitava prirodan broj n<1000 a
    zatim n izraza
    u notaciji
33 + 4 17
    - 8 -16
35 Program treba da ispise maksimalnu vrednost medju unetim izrazima
    i da ispise one
    izraze cija je vrednost manja od polovine maksimalne vrednosti.

37
39
41 */
43 #include <stdio.h>
45 #define MAX 1000
47 typedef struct izraz
49 {
    char o;
    int x;
    int y;
} IZRAZ;

51
53 /*
    Funkcija korektan_izraz vraca 1 ako je izraz korektan a 0
55 u suprotnom. Izraz je korektan ukoliko se sastoji od
    aritmetickih operacija +,-,* ili /, i ukoliko je u slucaju
57 operacije deljenja drugi operand razlicit od nule.
    */
59 int korektan_izraz(const IZRAZ* izraz)
61 {
    if(izraz->o!='+' && izraz->o!='-' && izraz->o!='*' && izraz->o!='/'
        '&')
    {
63         printf("Nedozvoljena operacija!\n");
        return 0;
65     }
    if(izraz->o=='/' && izraz->y==0)
67     {
        printf("Deljenje nulom!\n");
    }
}

```

### 3 Predstavljanje podataka

---

```
69     return 0;
70 }
71 return 1;
72 }
73
74 /*
75 Promenljiva izraz ce se promeniti u funkciji
76 vrednost tako sto ce njenom neinicijalizovanom
77 polju vr biti dodeljena vrednost izraza. Zbog
78 toga ovu promenljivu funkciji prosledjujemo
79 po adresi, preko pokazivaca
80 */
81
82 int vrednost(const IZRAZ* izraz)
83 {
84     int v;
85
86     switch (izraz->o)
87     {
88         case '+':
89             v=izraz->x+izraz->y;
90             break;
91         case '-':
92             v=izraz->x-izraz->y;
93             break;
94         case '*':
95             v=izraz->x*izraz->y;
96             break;
97         case '/':
98             v=izraz->x/izraz->y;
99             break;
100     }
101     return v;
102 }
103
104 /*
105 Promenljiva izraz ce se promeniti u funkciji
106 ucitaj_izraz tako sto ce njenim neinicijalizovanim
107 poljima o,x,y biti dodeljene vrednosti učitane
108 sa standardnog ulaza. Zbog toga ovu promenljivu
109 funkciji prosledjujemo po adresi, preko pokazivaca.
110
111 S obzirom da učitavanje karaktera nije prvo
112 učitavanje koje se obavlja u programu, funkcijom
113 getchar() "pokupimo" karakter kojim razdvajamo
114 unos karaktera od prethodnog unosa (najcesce blanko
115 znak)
116 */
117
118 int ucitaj_izraz(IZRAZ* izraz)
```

```
121 {
122     getchar();
123     scanf("%c%d%d",&izraz->o, &izraz->x, &izraz->y);
124     if (!korektan_izraz(izraz))
125         return 0;
126     return 1;
127 }
128
129 void stampa_izraz(const IZRAZ* izraz)
130 {
131     printf("%d %c %d = %d\n", izraz->x, izraz->o, izraz->y, vrednost(
132         izraz));
133 }
134
135 int max_vr(IZRAZ izrazi[], int n)
136 {
137     int i;
138     int max;
139     /* Trazimo maksimalnu vrednost izraza */
140     max=vrednost(&izrazi[0]);
141
142     /* U petlji... */
143     for(i=1; i<n; i++)
144     /* Ako je ona veca od maksimalne: */
145         if(vrednost(&izrazi[i])>max)
146             /* Azuriramo max: */
147             max=vrednost(&izrazi[i]);
148     return max;
149 }
150
151 int main()
152 {
153     int n;
154     IZRAZ izrazi[MAX];
155     int max;
156     int i;
157
158     /* Ucitavamo broj izraza: */
159     scanf("%d", &n);
160     if(n<0 || n>MAX)
161     {
162         printf("Nekorektna vrednost broja n!\n");
163         return -1;
164     }
165
166     /* U petlji ucitavamo jedan po jedan izraz: */
167     for(i=0; i<n; i++)
168         if(ucitaj_izraz(&izrazi[i])==0)
169         {
170             printf("Nekorektan unos\n");
171         }
```

```
173         return -1;
174     }
175
176     printf("Svi izrazi:\n");
177     for(i=0; i<n; i++)
178         stampaj_izraz(&izrazi[i]);
179
180     max = max_vr(izrazi, n);
181     printf("Maksimalna vrednost izraza:%d\n", max);
182
183     printf("Izrazi cija je vrednost manja od polovine maksimalne
184           vrednosti:\n");
185
186     for(i=0; i<n; i++)
187         if(vrednost(&izrazi[i])<max/2)/* Ako je vrednost tekuceg izraza
188           manja od polovine maksimalne, ispisujemo ga. */
189             stampaj_izraz(&izrazi[i]);
190
191     return 0;
192 }
```

Rešenje 3.69

Rešenje 3.69

Rešenje 3.69

Rešenje 3.69

Rešenje 3.69

Rešenje 3.69

# 4

## Ulaz i izlaz programa

### 4.1 Standardni tokovi

### 4.2 Argumenti komandne linije

### 4.3 Datoteke

**Zadatak 4.1** Tekst

[Rešenje [4.1](#)]

**Zadatak 4.2** Tekst

[Rešenje [4.2](#)]

**Zadatak 4.3** Tekst

[Rešenje [4.3](#)]

**Zadatak 4.4** Tekst

[Rešenje [4.4](#)]

**Zadatak 4.5** Tekst

[Rešenje [4.5](#)]

**Zadatak 4.6** Tekst

[Rešenje [4.6](#)]

**Zadatak 4.7** Napisati program koji prebrojava mala slova u datoteci *test.txt*.

*Primer 1*

```
TEST.TXT
Abcd EFGH+ijKLMN

IZLAZ:
Broj malih slova je: 5
```

*Primer 2*

```
TEST.TXT
PrograMiranje

IZLAZ:
Broj malih slova je: 11
```

[Rešenje 4.15]

**Zadatak 4.8** Napisati program koji prepisuje svaki treći karakter datoteke *ulaz.txt* u datoteku *izlaz.txt*.

*Primer 1*

```
ULAZ.TXT
Volim programiranje.
IZLAZ.TXT
Vipgmae
```

[Rešenje 4.15]

**Zadatak 4.9** Kao argumenti komandne linije se zadaju ime datoteke i ceo broj  $k$ . Napisati program koji na standardni izlaz ispisuje sve linije zadate datoteke čija je dužina veća od  $k$ . Može se pretpostaviti da dužina linije neće biti veća od 80 karaktera.

*Primer 1*

```
POKRETANJE: ./a.out test.txt 7
TEST.TXT
Teme koje su obradjuvane:
Petlje
Funkcije
Nizovi
Strukture

IZLAZ:
Teme koje su obradjuvane:
Funkcije
Strukture
```

*Primer 2*

```
POKRETANJE: ./a.out test.txt

IZLAZ:
Greska: Pogresan broj argumenata!
```

[Rešenje 4.15]

**Zadatak 4.10** Napisati program koji prebrojava koliko se linija datoteke *ulaz.txt* završava niskom  $s$  koja se učitava sa standardnog ulaza. Može se pretpostaviti da dužina linije neće biti veća od 80 karaktera, kao i da dužina niske  $s$



neće biti veća od 20 karaktera.

#### Primer 1

```

ULAZ.TXT
abcde abcde
abcde aab
abcde abcde abcde
abcde abcde Aab
abcde abcde ab
abcde abcde abcde abcde

INTERAKCIJA SA PROGRAMOM:
Unesite nisku s: ab
Broj linija: 3

```

#### Primer 2

```

ULAZ.TXT
abcde abcde
abcde
abcde abcde AB

INTERAKCIJA SA PROGRAMOM:
Unesite nisku s: ab
Broj linija: 0

```

[Rešenje 4.15]

**Zadatak 4.11** Napisati program koji pronalazi maksimum brojeva zapisanih u datoteci *brojevi.txt*.

#### Primer 1

```

BROJEVI.TXT
2.36 -16.11 5.96 8.88
-265.31 54.96 38.4

IZLAZ:
Najveci broj je: 54.96

```

[Rešenje 4.15]

**Zadatak 4.12** U datoteci *studenti.txt* se nalaze informacije o studentima: prvo broj studenata, a zatim u pojedinačnim linijama korisničko ime i pet poslednjih ocena koje je student dobio. Napisati program koji pronalazi studenta koji je ostvario najbolji uspeh i ispisuje njegove podatke. Pretpostaviti da broj studenata neće biti veći od 100.

#### Primer 1

```

STUDENTI.TXT
mr15239 10 9 9 8 10
m114005 8 8 9 8 10
m115112 9 8 8 7 10
mr15007 10 10 10 10 10
mn13208 7 7 9 6 10

IZLAZ:
korisnicko ime: mr15007, prosek ocena: 10.00

```

[Rešenje 4.15]

**Zadatak 4.13** U datoteci *tacke.txt* se nalazi prvo broj tačaka, a zatim u pojedinačnim linijama  $x$  i  $y$  koordinate tačke. Napisati program koji u datoteku *rastojanja.txt* upisuje rastojanje svake od pročitanih tačaka od koordinatnog početka, a na standardni izlaz koordinate tačke koja je najudaljenija. Koristiti strukturu *Tacka* sa poljima  $x$  i  $y$ , kao i funkciju kojom se računa rastojanje. Pretpostaviti da broj tačaka u datoteci neće biti veći od 50.

### Primer 1

```
TACKE.TXT
4
11 -2
3 5
8 -8
0 4

RASTOJANJA.TXT
11.18
5.29
11.31
4.00

IZLAZ:
Najudaljenija je tačka: 8 -8
```

### Primer 1

```
TACKE.TXT
-2
0 0
9 -8

IZLAZ:
Greska: Nedozvoljen broj tacaka!
```

[Rešenje 4.15]

**Zadatak 4.14** Napisati program koji za reč  $s$  maksimalne dužine 20 karaktera koja se zadaje sa standardnog ulaza u datoteku *rotacije.txt* upisuje sve rotacije reči  $s$ .

### Primer 1

```
INTERAKCIJA SA PROGRAMOM:
Unesite rec: abcde

ROTACIJE.TXT
abcde
bcdea
cdeab
deabc
eabcd
```

[Rešenje 4.15]

**Zadatak 4.15** Napisati program koji linije koji se učitavaju sa standardnog ulaza sve do kraja ulaza prepisuje u datoteku *izlaz.txt* i to, ako je prilikom pokretanja zadata opcija  $-v$  ili  $-V$  samo one linije koje počinju velikim slovom, ako je zadata opcija  $-m$  ili  $-M$  samo one linije koje počinju malim slovom, a ako je opcija izostavljena sve linije. Pretpostaviti da linije neće biti duže od 80 karak-

tera.

#### Primer 1

```
POKRETANJE: ./a.out -m
INTERAKCIJA SA PROGRAMOM:
Unesite recenice:
programiranje u C-u je zanimljivo
Volim programiranje!
Kada porastem bicu programer!
u slobodno vreme programiram

IZLAZ.TXT
programiranje u C-u je zanimljivo
u slobodno vreme programiram
```

#### Primer 2

```
POKRETANJE: ./a.out -V
INTERAKCIJA SA PROGRAMOM:
Unesite recenice:
programiranje u C-u je zanimljivo
Volim programiranje!
Kada porastem bicu programer!
u slobodno vreme programiram

IZLAZ.TXT
Volim programiranje!
Kada porastem bicu programer!
```

#### Primer 3

```
POKRETANJE: ./a.out -k
INTERAKCIJA SA PROGRAMOM:
Greska: Pogresno pokretanje programa!
```

[Rešenje 4.15]

## 4.4 Rešenja

### Rešenje 4.1

```
1  /*
2     Napisati program koji prepisuje sadrzaj datoteke ulaz.txt u
3     datoteku izlaz.txt karakter po karakter.
4  */
5
6  #include <stdio.h>
7  #include <stdlib.h>
8
9  int main()
10 {
11     int c;
12     FILE *ulaz, *izlaz;
13
14     /*
15      Promenljive ulaz i izlaz predstavljaju
16      pokazivace na ugradjenu strukturu FILE.
17      Unutar ove strukture nalaze se polja neophodna
18      za rad sa datotekama.
19
20      Kada zelimo da radimo sa nekom datotekom,
21      moramo je prvo otvoriti. Ugradjena funkcija
22      fopen(dat, mode) otvara datoteku sa nazivom
```

```
23     dat. Datoteka moze biti otvorena za citanje,  
24     pisanje ili nadovezivanje, sto odredjuje  
25     argument mode koji moze imati vrednost "r" (read),  
26     "w"(write) ili "a"(append).  
27 */  
28  
29 ulaz=fopen("ulaz.txt","r");  
30  
31 /*  
32     Do neuspesnog otvaranja datoteke moze doci  
33     ukoliko ne postoji datoteka sa datim nazivom  
34     ili je putanja do datoteke pogresna. U tom  
35     slucaju, funkcija fopen vraca pokazivac na NULL  
36     i tada treba prijaviti gresku. Datoteka stderr  
37     predstavlja standardnu datoteku u koju se upisuju  
38     greske. Stderr je podrazumevano postavljen  
39     na standardni izlaz.  
40  
41     Ugradjena funkcija exit prouzrokuje zavrsetak programa.  
42     Argument ove funkcije je jedna od konstanti definisanih  
43     u biblioteci stdlib.h koje pokazuju da li se program  
44     zavrrio uspesno (EXIT_SUCCESS) ili neuspesno (EXIT_FAILURE).  
45  
46 */  
47 if(ulaz==NULL)  
48 {  
49     fprintf(stderr,"error fopen(): Neuspelo otvaranje datoteke ulaz  
50     .txt za citanje.\n");  
51     exit(EXIT_FAILURE);  
52 }  
53  
54 izlaz= fopen("izlaz.txt", "w");  
55 if(izlaz==NULL)  
56 {  
57     fprintf(stderr,"error fopen(): Neuspelo otvaranje datoteke  
58     izlaz.txt za citanje.\n");  
59     exit(EXIT_FAILURE);  
60 }  
61  
62 /*  
63     Funkcija fgetc ucitava jedan karakter iz datoteke ulaz.  
64     Povratna vrednost ove funkcije je ascii kod unetog  
65     karaktera.  
66  
67     Funkcija fputc ispisuje karakter c u datoteku izlaz.  
68 */  
69 while((c=fgetc(ulaz))!=EOF)  
70     fputc(c,izlaz);  
71  
72 /*  
73     Nakon zavrsetka rada sa datotekama, neophodno ih je
```

```

73     zatvoriti pomocu ugradjene funkcije fclose.
    */
75     fclose(ulaz);
    fclose(izlaz);
77     return 0;
}

```

## Rešenje 4.2

```

/*
2   Napisati program koji u datoteci cije se ime navodi kao prvi
   argument komandne linije odredjuje liniju maksimalne duzine i
4   ispisuje je na standarni izlaz. Ukoliko ima vise takvih linija,
   ispisati onu koja je leksikografski prva. Mozemo pretpostaviti
6   da datoteka ne sadrzi linije duze od 80 karaktera.
*/
8   #include <stdio.h>
   #include <stdlib.h>
10  #include <string.h>
   #define MAX_LEN 81
12
   int main(int argc, char* argv[])
14 {
   char linija[MAX_LEN];
16   char max_linija[MAX_LEN];
   int duzina;
18   int max_duzina;

20   FILE *ulaz, *izlaz;

22   /*
   Proveravamo da li poziv programa ima dovoljan broj argumenata.
24   */
   if(argc!=2)
26   {
       fprintf(stderr, "Greska: program se pokrece sa: %s
           ime_ulazne_datoteke\n", argv[0]);
28       exit(EXIT_FAILURE);
   }

30   ulaz=fopen(argv[1], "r");
32   if(ulaz==NULL)
   {
34       fprintf(stderr, "error fopen(): Neuspelo otvorenje datoteke %s
           za citanje.\n", argv[1]);
           exit(EXIT_FAILURE);
36   }

38   /*
       Funkcija fgets ucitava jednu liniju teksta maksimalne duzine
       MAX_LEN

```

## 4 Ulaz i izlaz programa

---

```
40     iz datoteke ulaz u string linija. Ukoliko učitavanje ne uspe (
na primer,
    zato sto smo dosli do kraja datoteke), povratna vrednost ove
funkcije
42     bice prazan pokazivac (NULL).
*/
44
max_duzina=0;
46 while(fgets(linija, MAX_LEN, ulaz)!=NULL)
{
48     duzina = strlen(linija);
    /*
50     Promenljivu max_duzina inicijalizovali smo na 0 pre ulaska u
petlju.
    Ovu promenljivu menjamo kada je duzina učitana linije
52     veca od max_duzina ili kada su jednake, ali je učitana
linija
    leksikografski ispred trenutne linije sa maksimalnom duzinom
    .
54
    Setimo se da funkcija strcmp(s1,s2) vraca razliku ascii
kodova prva dva
56     razlicita karaktera stringova s1 i s2 na istim indeksima,
ukoliko oni
    postoje, ili 0 ukoliko su jednaki. Ova funkcija je stoga
osetljiva
58     na mala i velika slova (npr 'D' je leksikografski ispred 'p
').
60
    */
62     if(duzina>max_duzina || (duzina==max_duzina && strcmp(linija,
max_linija)<0))
    {
64         strcpy(max_linija, linija);
        max_duzina=duzina;
66     }
}
68
/*
70     Funkcija fputs ispisuje string koji je njen prvi argument u
datoteku
    koja je njen drugi argument. Sve funkcije za učitavanje iz
datoteka i
72     upis u datoteke (fgetc, fputc, fgets, fputs, ...) mozemo
koristiti
    i kada radimo sa standardnim ulazom i standardnim izlazom. Kao
nazive
74     datoteka tada navodimo stdin i stdout.
*/
76 fputs(max_linija, stdout);
```

```
78     fclose(ulaz);  
    return 0;  
80 }
```

### Rešenje 4.3

```
2  /*  
    U datoteci cije se ime zadaje kao prvi argument komandne linije  
    nalazi se  
    prirodan broj n a zatim i n celih brojeva. Napisati program koji  
    prebrojava  
4  koliko k-tocifrenih brojeva postoji u datoteci, pri cemu se  
    prirodan broj k  
    zadaje kao drugi argument komandne linije.  
6  */  
8  #include <stdio.h>  
    #include <stdlib.h>  
10 #include <math.h>  
12 /*  
    Funkcija ucitaj_i_prebroj ucitava brojeve  
    iz datoteke na koju pokazuje f i prebrojava  
    koliko je medju njima k-tocifrenih brojeva  
16 */  
    int ucitaj_i_prebroj(FILE* f, int k)  
18 {  
    int n;  
    int x;  
    int i;  
    int br;  
24  
    /* U datoteci je prvo naveden ukupan broj brojeva. */  
    fscanf(f, "%d", &n);  
26  
    /* Ako je taj broj negativan ili nula, izdajemo poruku o gresci.  
    */  
28    if(n<=0)  
    {  
30        fprintf(stderr, "Greska: broj n mora biti prirodan\n");  
        exit(EXIT_FAILURE);  
32    }  
34    br=0;  
    for(i=0; i<n; i++)  
36    {  
        fscanf(f, "%d", &x);  
38        if(broj_cifara(x)==k)  
            br++;  
40    }  
}
```

```
42     return br;
43 }
44
45 int broj_cifara(int x)
46 {
47     int br_c;
48
49     br_c=0;
50
51     /*
52      Do while petlja je pogodnija od petlji sa preduslovom
53      jer tacno racuna broj cifara i za broj 0.
54     */
55     do
56     {
57         br_c++;
58         x/=10;
59     } while(x);
60
61     return br_c;
62 }
63
64 int main(int argc, char* argv[])
65 {
66     int n;
67     int k;
68     FILE* f;
69     int br;
70
71     if(argc!=3)
72     {
73         fprintf(stderr, "Greska: program se pokrece sa: %s
74         naziv_datoteke k \n", argv[0]);
75         exit(EXIT_FAILURE);
76     }
77
78     f=fopen(argv[1], "r");
79
80     if(f==NULL)
81     {
82         fprintf(stderr, "Greska fopen(): Neuspelo otvaranje datoteke %s
83         .\n", argv[1]);
84         exit(EXIT_FAILURE);
85     }
86
87     /* Argumenti komandne linije su stringovi. Da bismo konvertovali
88     string
89     u ceo broj koristimo ugradjenu funkciju atoi. */
90     k = atoi(argv[2]);
91
92     if (k<=0)
93     {
```



```

    fprintf(stderr, "Greska: broj k mora biti prirodan\n");
92     exit(EXIT_FAILURE);
}

94     printf("Broj %d-cifrenih brojeva u datoteci: %d\n", k,
        ucitaj_i_prebroj(f,k));

96     fclose(f);
98     return 0;
}

```

### Rešenje 4.4

```

1  /*
   U datoteci cije se ime navodi kao prvi argument komandne
3  linije navedena je rec r i niz linija. Napisati
   program koji u datoteku cije se ime navodi kao
5  drugi argument komandne linije upisuje sve linije
   u kojima se rec r pojavljuje bar n puta, gde je
7  n prirodan broj koji se unosi sa standardnog ulaza. Ispis
   treba da bude u formatu broj_pojavljivanja: linija.
9  */

11 #include <stdio.h>
   #include <stdlib.h>
13 #define MAXL 81
   #define MAXR 31

15 /*
17  Funkcija broj_pojavljivanja broji koliko
   se puta pojavio string t u stringu s
19  */
   int broj_pojavljivanja(char s[], char t[])
21 {
   int br;
23   int i,j;
   /*
25   i - indeks karaktera u s
   j - indeks karaktera u t
27   br - brojac koliko se puta t javlja u s
   */
29   br=0;
   for(i=0;s[i];i++)
31   {
       for(j=0;t[j];j++)
33           if(s[i+j]!=t[j]) /* Ako naidjemo na razlicite karaktere, */
               break; /* prekidamo petlju. */
35   }
   /*
37   Do prekida petlje moze doci ili zbog toga sto su pronadjeni
   razliciti karakteri i usledio je break ili zbog toga sto
   je prestao da vazi uslov petlje, odnosno karakter t[j] je

```

```
39         jednak '\0'. Ako vazi drugi slucaj, to znaci da se string
        t nalazi u stringu s pocev od indeksa i i potrebno je
        uvecati
41         brojac br.
        */
43         if (t[j]=='\0')
            br++;
45     }

47     return br;
}
49 int main(int argc, char* argv[])
{
51     char rec[MAXR];
    char linija[MAXL];
53     FILE* in, *out;
    int n;
55     int br;

57     if(argc!=3)
    {
59         fprintf(stderr, "Greska: program se pokrece sa: %s
        ime_ulazne_datoteke ime_izlazne_datoteke\n", argv[0]);
        exit(EXIT_FAILURE);
61     }

63     in= fopen(argv[1], "r");
    if(in==NULL)
65     {
        fprintf(stderr, "Greska fopen(): Neuspelo otvaranje datoteke %s
        .\n", argv[1] );
        exit(EXIT_FAILURE);
67     }

69     out= fopen(argv[2], "w");
    if(out==NULL)
71     {
        fprintf(stderr, "Greska fopen(): Neuspelo otvaranje datoteke %s
        .\n", argv[2] );
        exit(EXIT_FAILURE);
73     }

75     printf("Unesi n:");
    scanf("%d", &n);
77
79     if(n<=0)
81     {
        fprintf(stderr, "Greska: n treba da bude prirodan broj.\n");
        exit(EXIT_FAILURE);
83     }

85     fscanf(in, "%s", rec);
```

```

87     while(fgets(linija,MAXL,in)!=NULL)
88     {
89         br = broj_pojavljivanja(linija,rec);
90         if (br>=n)
91             fprintf(out,"%d: %s\n", br, linija);
92     }
93     fclose(in);
94     fclose(out);
95     return 0;
96 }

```

### Rešenje 4.5

```

1  /* Program se pokrece tako sto se navedu nazivi dve datoteke(ulazna i
2     izlazna) i opcije.
3     U datoteci cije se ime navodi kao prvi argument komandne linije
4     nalaze se podaci o razlomcima:
5     u prvom redu se nalazi broj razlomaka, a u svakom sledecem redu
6     brojilac i imenilac jednog razlomka.
7     Potrebno je kreirati strukturu koja opisuje razlomak i učitati niz
8     razlomaka
9     iz datoteke, a potom:
10    a) ukoliko je navedena opcija x, upisati u datoteku cije je ime
11    drugi argument komandne linije
12    recipročni razlomak za svaki razlomak iz niza (npr. za 2/3
13    treba upisati 3/2)
14    b) ukoliko je navedena opcija y, upisati u datoteku cije je ime
15    drugi argument komandne linije
16    realnu vrednost reciprocnog razlomka svakog razlomka iz niza
17    (npr. za 2/3 treba upisati 1.5)
18    Mozemo pretpostaviti da se u datoteci sa podacima o razlomcima
19    nalazi najvise 100 razlomaka.
20 */
21
22 /*
23    Prilikom pokretanja programa se, pored naziva ulazne i izlazne
24    datoteke, navode i opcije -x i -y. Moguce je navesti jednu ili
25    obe opcije, sto znaci da je minimalni broj argumenata 3.
26
27    Moguci nacini pokretanja:
28    ./a.out ulaz.txt izlaz.txt -x
29    ./a.out ulaz.txt izlaz.txt -y
30    ./a.out ulaz.txt izlaz.txt -yx
31    ./a.out ulaz.txt izlaz.txt -xy
32
33 */
34
35 #include <stdio.h>
36 #include <stdlib.h>
37 #include <ctype.h>

```

```
29  #define MAX 100
31
32  typedef struct razlomak
33  {
34      int br;
35      int im;
36  } RAZLOMAK;
37
38  /*
39   Funkcija ucitaj_razlomke ucitava razlomke iz datoteke
40   na koju pokazuje f u niz. Dimenzija niza, na koju
41   pokazuje pokazivac dim, nije poznata. Prva vrednost
42   u datoteci je ukupan broj razlomaka i tu vrednost
43   ucitavamo u promenljivu dim.
44
45   Funkcija fscanf se koristi isto kao i funkcija scanf
46   uz dodatni prvi argument koji predstavlja naziv
47   datoteke iz koje se vrsi ucitavanje.
48
49  */
50  int ucitaj_razlomke(RAZLOMAK niz[], int* dim, FILE* f)
51  {
52      int i;
53
54      fscanf(f, "%d", dim);
55      for (i=0; i< *dim; i++)
56      {
57          fscanf(f, "%d %d", &niz[i].br, &niz[i].im);
58          if (niz[i].im==0)
59              return 0;
60      }
61      return 1;
62  }
63
64  RAZLOMAK recipročni(RAZLOMAK* r)
65  {
66      RAZLOMAK rec;
67      rec.im = r->br;
68      rec.br = r->im;
69      return rec;
70  }
71
72  float vrednost(RAZLOMAK* r)
73  {
74      return 1.0*r->br/r->im;
75  }
76
77  int main(int argc, char* argv[])
78  {
79      FILE *in, *out;
80      char c;
```

```
81  int i;
82  int j;
83  int xoption=0;
84  int yoption=0;
85  int dim;
86  RAZLOMAK razlomci[MAX];
87  RAZLOMAK r;

89  /*
90   Prilikom pokretanja programa se, pored naziva ulazne i izlazne
91   datoteke, navode i opcije -x i -y. Moguce je navesti jednu ili
92   obe opcije, sto znaci da je minimalni broj argumenata 3.
93
94   Moguci nacini pokretanja:
95   ./a.out ulaz.txt izlaz.txt -x
96   ./a.out ulaz.txt izlaz.txt -y
97   ./a.out ulaz.txt izlaz.txt -yx
98   ./a.out ulaz.txt izlaz.txt -xy
99
100  */
101
102  if(argc!=4)
103  {
104      fprintf(stderr,"Greska: program se pokrece sa: %s
105      ime_ulazne_datoteke ime_izlazne_datoteke [ -x | -y]\n", argv[0]);
106      exit(EXIT_FAILURE);
107  }

108
109  in= fopen(argv[1],"r");
110  if(in==NULL)
111  {
112      fprintf(stderr, "Greska fopen(): Neuspelo otvaranje datoteke %s
113      .\n", argv[1] );
114      exit(EXIT_FAILURE);
115  }

116
117  out= fopen(argv[2],"w");
118  if(out==NULL)
119  {
120      fprintf(stderr, "Greska fopen(): Neuspelo otvaranje datoteke %s
121      .\n", argv[2] );
122      exit(EXIT_FAILURE);
123  }

124  /* Ispitujemo da li je treca opcija u trazenom formatu. Prvi
125   karakter mora biti '-' */
126
127  if (argv[3][0] != '-')
128  {
129      fprintf(stderr, "Greska u zadavanju opcija: program se pokrece
130      sa: %s ime_ulazne_datoteke ime_izlazne_datoteke [ -x | -y]\n",
```

```
    argv[0]);
    exit(EXIT_FAILURE);
129 }

131 /* Ostali karakteri mogu biti 'x' ili 'y'. U zavisnosti od date
    opcije, postavljamo
    vrednosti indikatorskih promenljivih xoption i yoption. */
133
135 for(j=1;argv[3][j]!='\0';j++)
    switch(argv[3][j])
    {
137         case 'x': xoption=1;
                    break;
139         case 'y': yoption=1;
                    break;
141         default:
                    fprintf(stderr, "Greska: nedozvoljeni karakter\n"
    );
                    exit(EXIT_FAILURE);
143     }
145
147 if(ucitaj_razlomke(razlomci, &dim, in)==0)
    {
149         fprintf(stderr, "Greska pri zadavanju razlomaka\n");
            exit(EXIT_FAILURE);
151     }
153
155 /*
    U zavisnosti od datih opcija, vrsimo upis reciprocni
    razlomaka u trazenom formatu.

157 Funkcija fprintf se koristi na isti nacin kao
    funkcija printf uz dodatni prvi argument koji
    oznacava naziv datoteke u koju se vrsi upis.
159 */
161 for (i=0; i<dim;i++)
    {
163         /*
            Ukoliko je brojilac razlomka jednak nuli,
            nema smisla traziti njegovu reciprocnu vrednost
165         */
            if (razlomci[i].br==0)
                continue;
167
            r = reciprocni(&razlomci[i]);
169
171             if (xoption)
                fprintf(out,"%d/%d ", r.br, r.im);
173
175             if (yoption)
                fprintf(out, "%f ", vrednost(&r));
```

```

177     fprintf(out, "\n");
179 }

181 fclose(in);
182 fclose(out);
183
184     return 0;
185 }

```

### Rešenje 4.6

```

1  /*
2     Za svaki automobil poznati su marka, model i cena. Iz datoteke cije
3     se ime zadaje sa standardnog ulaza ucitava se broj automobila a
4     potom
5     i podaci za svaki automobil. Program treba da:
6     a) izracuna proseccnu cenu po marki kola
7     b) za maksimalnu cenu koju je kupac spreman da plati, a koja se
8     zadaje
9     kao argument komandne linije, da ispise automobile u tom cenovnom
10    rangu zajednu sa proseccnom cenom odgovarajuce marke
11
12    Mozemo pretpostaviti da se model i marka sastoje od jedne reci i
13    da svaka od njih sadrzi najvise 30 karaktera kao i da se u datoteci
14    nalaze podaci za najvise 100 automobila.
15
16    */
17
18    #include <stdio.h>
19    #include <stdlib.h>
20    #include <string.h>
21    #define MAX 31
22    #define MAXA 100
23
24    typedef struct automobil
25    {
26        char marka[MAX];
27        char model[MAX];
28        float cena;
29    } AUTOMOBIL;
30
31    /*
32     Struktura INFO sadrzi naziv
33     marke automobila, proseccna cena
34     za tu marku i broj automobila
35     te marke
36    */
37    typedef struct info
38    {
39        char marka[MAX];

```

```

    float vrednost;
39    int n;
} INFO;

41
42 int ucitaj_podatke(FILE* f, AUTOMOBIL a[], int* pn, int max)
43 {
44     int i;
45
46     fscanf(f, "%d", pn);
47     if (*pn <= 0 || *pn > max)
48     {
49         printf("Nekorektan unos dimenzije niza automobila\n");
50         return 0;
51     }
52     for(i=0; i < *pn; i++)
53         fscanf(f, "%s %s %f", a[i].marka, a[i].model, &a[i].cena);
54
55     return 1;
56 }
57
58 /*
59 Funkcija sadrzi ispituje da li se u nizu proseka po marki
60 nalazi prosek za marku m. Posto podatak o marki automobila
61 predstavlja string, poredjenje vrsimo pomocu funkcije strcmp.
62
63 Povratna vrednost ove funkcije je indeks pojavljivanja, ukoliko
64 se marka m pojavljuje u nizu proseka, ili -1 u suprotnom.
65 */
66
67 int sadrzi(INFO p[], int n, char m[])
68 {
69     int i;
70     for(i=0; i < n; i++)
71         if(strcmp(p[i].marka, m) == 0)
72             return i;
73
74     return -1;
75 }
76
77 /*
78 Funkcija informacije_o_markama za niz automobila a dimenzije n
79 racuna proseke cena automobila po markama i smesta ih u niz
80 p. Na dimenziju niza p pokazuje pokazivac pn.
81
82 Ideja je da jednim prolaskom kroz niz sa svaku marku izracunamo
83 sumu cena automobila te marke (koju cemo smestiti u polje vrednost
84 strukture
85 INFO), i broj automobila te marke (koju cemo smestiti u polje
86 n strukture INFO) i da na kraju podelimo ove dve promenljive
87 i tako dobijemo prosečnu vrednost cene.
88
89 Za svaki automobil a[i] proveravamo da li se njegova marka vec

```



```

89     nalazi u nizu p. Ukoliko se nalazi, nadjenom elementu dodajemo
vredost cene automobila a[i] i uvecavamo broj automobila sa
91     tom markom. U suprotnom, dodajemo novi element u niz p. Posto
ga dodajemo na kraj, njegov indeks odgovara dimenziji niza p
93     na koju pokazuje pokazivac *pn.
*/
95 void informacije_o_markama(AUTOMOBIL a[], int n, INFO p[], int* pn1)
{
97     int i,j;
int ind;
99     for(i=0;i<n;i++)
    {
101         /* Proveravamo da li se marka automobila a[i] vec nalazi u
nizu p (niz proseka po markama) */
103         ind = sadrzi(p,*pn1,a[i].marka);
if(ind!=-1) /* Ako se ne nalazi, uvodimo novi element niza na
kraj, na poziciju *pn. */
105         {
            strcpy(p[*pn1].marka, a[i].marka);
107             p[*pn1].vrednost = a[i].cena;
p[*pn1].n = 1;
109             (*pn1)++; /* Zagrade su neophodne zbog prioriteta operatora.
*/
        }
111         else /* Ako se nalazi, azuriramo polja strukture. */
        {
113             p[ind].vrednost+=a[i].cena;
p[ind].n++;
115         }
    }
117
    /* Na osnovu sume cena i broja automobila racunamo prosečnu
vrednost. */
119     for(i=0;i<*pn1;i++)
        p[i].vrednost = p[i].vrednost/p[i].n;
121 }
123
void stampaj_informacije(INFO p[], int n)
125 {
    printf("Informacije o broju automobila i prosečnoj ceni po markama
:\n");
127     int i;
for(i=0;i<n;i++)
129         printf("%s %.2f %d\n", p[i].marka, p[i].vrednost, p[i].n);
}
131
/*
133 Funkcija stampa automobile čija je cena manja od maksimalne
cene koju je korisnik naveo u komandnoj liniji da je spreman
135 da plati, zajedno sa prosečnom cenom za tu marku automobila
*/

```

## 4 Ulaz i izlaz programa

---

```
137 void stampaj_kandidate(AUTOMOBIL a[], int n, float g, INFO p[], int
    n1)
139 {
    /*
    S obzirom da je niz p formiran na osnovu niza a, marka svakog
141 automobila iz niza a se sigurno nalazi u nizu p. Zbog toga
    nije neophodno proveravati da li je povratna vrednost funkcije
143 sadrzi razlicita od -1.
    */
145 int i;
    printf("Kola u vasem cenovnom rangu:\n");
147 for(i=0;i<n;i++)
    if(a[i].cena<g)
149     printf("%s %s %.2f\n", a[i].marka, a[i].model, p[sadrzi(p,n1
        ,a[i].marka)].vrednost);
    }
151
153 int main(int argc, char* argv[])
    {
        AUTOMOBIL kola[MAXA];
155 FILE* f;
        char dat[MAX]; /* Naziv datoteke koji se unosi sa standardnog
            ulaza. */
157 float granica; /* Maksimalna cena koju je korisnik spreman da
            plati.
            Zadaje se kao argument komandne linije.
159 */
        INFO infos[MAXA];
161 int dim_kola,dim_infos;
        int i;
163
165 if(argc!=2)
    {
        fprintf(stderr,"Greska: program se pokrece sa: %s
167 gornja_granica_cene \n", argv[0]);
        exit(EXIT_FAILURE);
    }
169
    /* Argumenti komandne linije su stringovi. Da bismo od stringa
        dobili
171 realan broj, koristimo ugradjenu funkciju atof. */
    granica = atof(argv[1]);
173
175 printf("Unesi naziv datoteke:");
    scanf("%s", dat);
177
    f=fopen(dat, "r");
179
    if(f==NULL)
    {
181         fprintf(stderr, "Greska fopen(): Neuspelo otvaranje datoteke %s
            .\n", dat);
    }
```

```
183     exit(EXIT_FAILURE);
184 }
185 if (ucitaj_podatke(f,kola,&dim_kola,MAXA)==0)
186 {
187     fprintf(stderr, "Greska pri učitavanju podataka\n");
188     exit(EXIT_FAILURE);
189 }
190
191 informacije_o_markama(kola, dim_kola, infos, &dim_infos);
192
193 stampaj_informacije(infos,dim_infos);
194
195 stampaj_kandidate(kola, dim_kola, granica, infos, dim_infos);
196
197 fclose(f);
198 return 0;
199 }
```

Rešenje 4.15

Rešenje 4.15

Rešenje 4.15

Rešenje 4.15

Rešenje 4.15

Rešenje 4.15

Rešenje 4.15

Rešenje 4.15

Rešenje 4.15



# 5

## Razni zadaci

### 5.1 Rešenja



## Dodatak A

# Ispitni zadaci

### A.1 Kvalifikacioni zadaci

### A.2 Ispitni rokovi

#### A.2.1 Praktični deo ispita, jun ...

### A.3 Rešenja