

PROGRAMIRANJE 1

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

PROGRAMIRANJE 1

Zbirka zadataka

**Beograd
2016.**

Autori:

dr Milena Vujošević Jančić, docent na Matematičkom fakultetu u Beogradu

dr Jovana Kovačević, docent na Matematičkom fakultetu u Beogradu

Danijela Simić, asistent na Matematičkom fakultetu u Beogradu

Anđelka Zečević, asistent na Matematičkom fakultetu u Beogradu

PROGRAMIRANJE 1

Zbirka zadataka

Sadržaj

0.1	Strukture	v
0.2	Rešenja	xi

0.1 Strukture

Zadatak 0.1.1 Definirati strukturu kojom se predstavlja kompleksan broj. Napisati funkcije koje izračunavaju zbir, razliku, proizvod i količnik dva kompleksna broja, a zatim 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 0.1.1]

Zadatak 0.1.2 Definirati strukturu kojom se predstavlja razlomak. Napisati funkcije koje izračunavaju zbir i proizvod dva razlomka. Unosi se broj n a potom i n razlomaka sa standardnog ulaza. Ispisati njihov zbir i proizvod na standardni izlaz.

Primer 1

```
INTERAKCIJA SA PROGRAMOM:
Unesi broj razlomaka: 5
Uneti razlomke:
1 2
7 8
3 4
5 6
2 9
Suma svih razlomaka je 229/72.
Proizvod svih razlomaka je 35/576.
```

Primer 2

```
INTERAKCIJA SA PROGRAMOM:
Unesi broj razlomaka: 10
Uneti razlomke:
4 3
12 25
3 8
1 3
8 9
2 3
5 6
-24 50
7 18
-7 19
Suma svih razlomaka je 6089/1368.
Proizvod svih razlomaka je 1568/577125.
```

[Rešenje 0.1.2]

Zadatak 0.1.3 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 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 najviše C vitamina je: kivi
```

[Rešenje 0.1.3]

Zadatak 0.1.4 Definisati strukturu *Grad* u kojoj se nalazi ime grada (niska 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 0.1.4]

Zadatak 0.1.5 Definirati strukturu `ParReci` koja sadrži reč na srpskom jeziku i odgovarajući prevod na engleski jezik. Napisati program koji do kraja ulaza učitava sve parove reči, a potom za rečenicu koja se zadaje u jednoj liniji ispisati prevod. Ako je reč u rečenici nepoznata umesto nje ispisati odgovarajući broj zvezdica. Maksimalna dužina reči je 50 karaktera, ukupan broj parova reči je maksimalno 100, a maksimalna dužina rečenice je 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
```

Primer 2

```
INTERAKCIJA SA PROGRAMOM:
zima winter
pas dog
sreca happiness
prijatelj friend
solja cup
covek man
Unesite recenicu za prevod:
pas je covekov najbolji prijatelj
dog is ***** best friend
```

[Rešenje 0.1.5]

Zadatak 0.1.6 Cenoteka pomaže kupcima da pronađu najpovoljniju cenu za proizvod koji žele da kupe. Napisati program koji učitava najpre broj različitih prodavnica (ceo broj manji od 50) a zatim i podatke o ceni traženog artikla –

zadaje se naziv prodavnice (niske maksimalne dužine 20 karaktera) i cena u toj prodavnici (realan broj). Korisnik zadaje željenu cenu proizvoda, a program ispisuje imena svih onih prodavnica u kojima je cena proizvoda jednaka ili manja od željene. U slučaju greške ispisati odgovarajuću poruku.

Primer 1

```
INTERAKCIJA SA PROGRAMOM:
Uneti broj prodavnica: 5
idea 58.9
mazi 58.2
roda 55.1
tempo 54.5
intereza 57.99
Uneti zeljenu cenu: 57.0
Povoljne prodavnice su:
roda
tempo
```

Primer 2

```
INTERAKCIJA SA PROGRAMOM:
Uneti broj prodavnica: 4
dm 43.2
lily 45.99
benu_apoteke 43.99
sephora 50.99
Uneti zeljenu cenu: 47.00
Povoljne prodavnice su:
dm
lily
benu_apoteke
```

[Rešenje 0.1.6]

Zadatak 0.1.7 Statistički zavod Srbije istražuje kako rade obdaništa u Srbiji. Za dato obdanište dobija spisak n dece sa kolonama: pol (m ili z), broj godina (od 3 do 6) i ocena koju je dete dalo radu obdaništa (od 1 do 5). Maksimalan broj dece u obdaništu je 200. Napisati program koji za decu datog pola i broja godina ispisuje na tri decimale prosečnu ocenu obdaništa. U slučaju neispravnog unosa ispisati odgovarajuću poruku.

Primer 1

```
INTERAKCIJA SA PROGRAMOM:
Uneti broj dece: 5
Uneti podatke za svako dete, pol,
broj godina i ocenu:
m 3 5
z 3 4
m 4 2
m 5 4
m 3 4
Uneti pol i broj godina: m 3
Prosečna ocena je: 4.500.
```

Primer 2

```
INTERAKCIJA SA PROGRAMOM:
Uneti broj dece: 10
Uneti podatke za svako dete, pol,
broj godina i ocenu:
m 3 5
z 4 4
m 5 4
z 4 3
z 3 2
z 4 5
m 6 5
z 4 4
z 4 5
m 6 3
Uneti pol i broj godina: z 4
Prosečna ocena je: 4.200.
```


Primer 3

```
INTERAKCIJA SA PROGRAMOM:
Uneti broj dece: 15
Uneti podatke za svako dete, pol,
broj godina i ocenu:
m 3 2
z 7 5
Neispravan broj godina.
```

Primer 4

```
INTERAKCIJA SA PROGRAMOM:
Uneti broj dece: 2
Uneti podatke za svako dete, pol,
broj godina i ocenu:
m 3 2
z 3 5
Uneti pol i broj godina: h 5
Neispravan pol.
```

[Rešenje 0.1.7]

Zadatak 0.1.8 Definirati 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. U slučaju greške ispisati odgovarajuću poruku.

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
```

Primer 2

```
INTERAKCIJA SA PROGRAMOM:
Unesite broj lopti: 8
Unesite dalje poluprecnike i boje lopti
(1-plava, 2-zuta, 3-crvena, 4-zelena):
1. lopta: 2 1
2. lopta: 30 3
3. lopta: 7 3
4. lopta: 4 1
5. lopta: 5 2
6. lopta: 6 2
7. lopta: 12 3
8. lopta: 14 2
Ukupna zapremina: 134996.34
Ukupno crvenih lopti: 3
```

Primer 3

```
INTERAKCIJA SA PROGRAMOM:
Unesite broj lopti: 8
Unesite dalje poluprecnike i boje lopti
(1-plava, 2-zuta, 3-crvena, 4-zelena):
1. lopta: 1 2
2. lopta: 2 10
Nekorektan unos.
```

[Rešenje 0.1.8]

* **Zadatak 0.1.9** Sefinirati strukturu **IZRAZ** kojom se opisuje numerički izraz nad celim brojevima koji se sastoji od dva celobrojna operanda, numeričke

operacije (sabiranje, oduzimanje, množenje ili celobrojno deljenje) nad celim brojevima.

- (a) Napisati funkciju koja ispituje da li je dati izraz korektno zadat i vraća 1 ako jeste a 0 u suprotnom. Podrazumevamo da je izraz korektno zadat ako operacija odgovara $+$, $-$, $*$ ili $/$ i u slučaju deljenja drugi operand je različit od 0.
- (b) Napisati funkciju koja za dati izraz određuje vrednost izraza.
- (c) Napisati funkciju koja učitava dati izraz. Funkcija treba da učitava sa standardnog ulaza izlaz koji je zadat prefiksno — prvo operacija, a potom dva operanda. Funkcija vraća 1 ako je učitavanje bilo uspešno, tj. ako je izraz bio korektno zadat ili 0 u suprotnom.
- (d) Napisati funkciju koja štampa dati izraz infiksno, u obliku "*operand₁ operacija operand₂ = vrednost*".

Napisati glavni program koji učitava prirodan broj n , ($n < 1000$) a zatim n izraza u prefiksnoj notaciji. Program treba da ispiše maksimalnu vrednost unetih izraza i sve izraze čija vrednost je manja od polovine maksimalne vrednosti.

Primer 1

```
INTERAKCIJA SA PROGRAMOM:
Uneti broj izraza: 4
Uneti izraze u prefiksnoj notaciji:
+ 10 4
- 9 2
* 11 2
/ 7 3
Maksimalna vrednost izraza: 22
Izrazi čija je vrednost manja
od polovine maksimalne vrednosti:
9 - 2 = 7
7 / 3 = 2
```

Primer 2

```
INTERAKCIJA SA PROGRAMOM:
Uneti broj izraza: 10
Uneti izraze u prefiksnoj notaciji:
+ 10 2
- -678 34
* 77 2
+ 1000 -23
+ 102 4
- 200 23
/ 67 12
/ 1000 2
* 44 6
/ 13 1
Maksimalna vrednost izraza: 977
Izrazi čija je vrednost manja
od polovine maksimalne vrednosti:
10 + 2 = 12
-678 - 34 = -712
77 * 2 = 154
102 + 4 = 106
200 - 23 = 177
67 / 12 = 5
44 * 6 = 264
13 / 1 = 13
```

Primer 3

```
INTERAKCIJA SA PROGRAMOM:  
Uneti broj izraza: 3  
Uneti izraze u prefiksnoj notaciji:  
* 1 2  
/ 3 0  
Deljenje nulom!  
Nekorektan unos
```

[Rešenje 0.1.9]

0.2 Rešenja

Rešenje 0.1.1

```
1  #include <stdio.h>  
  
3  /* Struktura koja opisuje kompleksni broj obuhvata polje za realni  
   * i polje za imaginarni deo broja.  
5  */  
   typedef struct Complex {  
7  
       float re;  
9       float im;  
   } Complex;  
  
11  /* Funkcija kojom se izracunava zbir kompleksnih brojeva. */  
13  Complex saberi(Complex *a, Complex *b) {  
  
15     Complex c;  
     c.re = a->re + b->re;  
17     c.im = a->im + b->im;  
     return c;  
19 }  
  
21  /* Funkcija kojom se izracunava razlika kompleksnih brojeva. */  
   Complex oduzmi(Complex *a, Complex *b) {  
23  
       Complex c;  
25     c.re = a->re - b->re;  
     c.im = a->im - b->im;  
27     return c;  
   }  
29  
   /* Funkcija kojom se izracunava proizvod kompleksnih brojeva. */  
31  Complex pomnozi(Complex *a, Complex *b) {
```

```

33     Complex c;
34     c.re = a->re * b->re - a->im * b->im;
35     c.im = b->re * a->im + a->re * b->im;
36     return c;
37 }

38
39 /* Funkcija kojom se izracunava kolicnik kompleksnih brojeva. */
40 Complex podeli(Complex *a, Complex *b) {
41
42     Complex c;
43     c.re = (a->re * b->re + a->im * b->im) / (b->re*b->re + b->im*b->im
44 );
45     c.im = (b->re * a->im - a->re * b->im) / (b->re*b->re + b->im*b->im
46 );
47     return c;
48 }
49
50 int main() {
51
52     Complex a, b;
53     Complex c;
54
55     /* Ucitavamo kompleksne brojeve. */
56     printf("Unesite realni i imaginarni deo prvog broja: ");
57     scanf("%f%f", &a.re, &a.im);
58
59     printf("Unesite realni i imaginarni deo drugog broja: ");
60     scanf("%f%f", &b.re, &b.im);
61
62     c = saberi(&a, &b);
63     /* Ukoliko je imaginarni deo negativan,
64      * njegov zapis vec ukljucuje znak,
65      * te to treba proveriti.
66      * Inace, broj je oblika a+b*i.
67      */
68     printf("Zbir: %.2f%c%.2f*i\n", c.re, c.im > 0 ? '+' : ' ', c.im);
69
70     c = oduzmi(&a, &b);
71     printf("Razlika: %.2f%c%.2f*i\n", c.re, c.im > 0 ? '+' : ' ', c.im);
72
73     c = pomnozi(&a, &b);
74     printf("Proizvod: %.2f%c%.2f*i\n", c.re, c.im > 0 ? '+' : ' ', c.im);
75
76     if(b.re != 0 || b.im != 0) {
77         c = podeli(&a, &b);
78         printf("Kolicnik: %.2f%c%.2f*i\n", c.re, c.im > 0 ? '+' : ' ', c.im);
79     }
80     /* U polju kompleksnih brojeva

```

```

79     * nije dozvoljeno deljenje nulom.
    */
81     else
        printf("Kolicnik ne postoji.\n");
83
85     return 0;
}

```

Rešenje 0.1.2

```

1  #include <stdio.h>
3  typedef struct
4  {
5      int brojilac;
6      int imenilac;
7  }razlomak;
9  int nzd(int a, int b)
10 {
11     int pom;
13     if (a < b)
14     {
15         pom = a;
16         a = b;
17         b = pom;
18     }
19     while(b != 0)
20     {
21         pom = a % b;
22         a = b;
23         b = pom;
24     }
25
26     return a;
27 }
29 razlomak zbir(razlomak a, razlomak b)
30 {
31     razlomak c;
32     int nzd_razlomka;
34
35     c.brojilac = a.brojilac * b.imenilac + b.brojilac*a.imenilac;
36     c.imenilac = a.imenilac*b.imenilac;
37
38     /* Brojilac i imenilac dobijenog zbira se dele najvećim zajedničkim
39     * deliocom.
40     */

```

```

41     nzd_razlomka = nzd(c.brojilac, c.imenilac);

43     c.brojilac = c.brojilac/nzd_razlomka;
    c.imenilac = c.imenilac/nzd_razlomka;

45     return c;
47 }

49 razlomak proizvod(razlomak a, razlomak b)
{
51     razlomak c;
    int nzd_razlomka;

53     c.brojilac = a.brojilac*b.brojilac;
55     c.imenilac = a.imenilac*b.imenilac;

57     /* Brojilac i imenilac dobijenog zbira se dele najvećim zajedničkim
    * deliocom.
59     */
    nzd_razlomka = nzd(c.brojilac, c.imenilac);

61     c.brojilac = c.brojilac/nzd_razlomka;
63     c.imenilac = c.imenilac/nzd_razlomka;

65     return c;
67 }

69 int main()
{
71     int n, i;

    razlomak suma, proizvod_svih, r;

73     printf("Unesi broj razlomaka: ");
75     scanf("%d", &n);

77     suma.brojilac = 0;
79     suma.imenilac = 1;

81     proizvod_svih.brojilac = 1;
    proizvod_svih.imenilac = 1;

83     printf("Uneti razlomke:\n");
85     for(i=0; i<n; i++)
    {
87         scanf("%d%d", &r.brojilac, &r.imenilac);

89         suma = zbir(suma, r);
        proizvod_svih = proizvod(proizvod_svih, r);

91     }

```

```

93     printf("Suma svih razlomaka je %d/%d.\n", suma.brojilac, suma.
        imenilac);
    printf("Proizvod svih razlomaka je %d/%d.\n", proizvod_svih.
        brojilac, proizvod_svih.imenilac);
95
    return 0;
97 }

```

Rešenje 0.1.3

```

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

4  #define MAX_DUZINA 21
    #define MAX_BR_VOCKI 50

6
    typedef struct vocka
8  {
        char ime[MAX_DUZINA];
10     float vitamin;
    } VOCKA;
12

14 int main()
    {
16     VOCKA vocke[MAX_BR_VOCKI];
        int i = 0, n, max_vocka;
18     char ime[MAX_DUZINA];

20     /*
        Program ucitava podatke o vockama i smesta ih u niz
22     sve dok se ne unese rec KRAJ ili ucita MAX_BR_VOCKI vocki.
    */
24     do
    {
26         printf("Unesite ime vocke i njenu kolicinu vitamina C: ");
            scanf("%s",ime);
28         /*
            Kada se unese rec KRAJ prekida se petlja.
        */
30         if(strcmp(ime, "KRAJ") == 0)
            break;
32

34         /*
            Inace ucitava se kolicina vitamina
36         i ta vrednost se smesta u vocku na poziciji "i".
        */
38         strcpy(vocke[i].ime,ime);
            scanf("%f",&vocke[i].vitamin);
40         i++;
    }
}

```

```

42  while(i<MAX_BR_VOCKI);

44  n = i;

46  /*
48  Pretpostavka je da prva vocka ima najvise vitamina.
   Petljom se prolazi niz vocki i ukoliko se naide na vocku koja
   ima vise vitamina
   od one koja trenutno ima najvise, azurira se vrednosti maksimalne
   vocke.

50  Sve vreme se cuva indeks vocke sa najvise vitamina C.
52  */

54  max_vocka = 0;
   for(i=1;i<n;i++)
56     if(vocke[i].vitamin > vocke[max_vocka].vitamin)
       {
58         max_vocka = i;
       }

60  printf("Voce sa najvise C vitamina je: %s\n", vocke[max_vocka].ime)
   ;

62  return 0;
64  }

```

Rešenje 0.1.4

```

1  #include <stdio.h>
   #define MAX_DUZINA 20
3  #define MAX_BR_GRADOVA 50

5  typedef struct Grad{
   char ime_grada[MAX_DUZINA+1];
7  float temperatura;
   }Grad;
9

11 int main(){
   int n, i;
13  Grad grad[MAX_BR_GRADOVA];

15  printf("Unesite broj n: ");
   scanf("%d", &n);
17  if(n<0 || n>MAX_BR_GRADOVA){
       printf("Greska: pogresan unos!\n");
19  return 0;
   }

21  for(i=0; i<n; i++){

```



```

23     printf("Unesite grad i temperaturu: ");
    scanf("%s %f", grad[i].ime_grada, &grad[i].temperatura);
25 }

27 printf("Gradovi sa idealnom temperaturom za klizanje u decembru:\n"
    );
    for(i=0; i<n; i++){
29         if(grad[i].temperatura>=3 && grad[i].temperatura<=8){
            printf("%s\n", grad[i].ime_grada);
31         }
    }
33
    return 0;
35 }

```

Rešenje 0.1.5

```

#include <stdio.h>
2 #include <string.h>
#define MAX_DUZINA 21
4 #define MAX_BR_REC 100

6 typedef struct ParReci{
    char sr[MAX_DUZINA+1];
    char en[MAX_DUZINA+1];
8 }ParReci;

10

12 /*
    Funkcija koja u rečniku koji sadrži n reči traži prevod reči rec i
    upisuje ga u prevod.
14    Ukoliko se rec ne nalazi u rečniku, prevod se sastoji od zvezdica
    pri čemu broj zvezdica odgovara
    dužini nepoznate reči.
16 */

18 void pronadji_prevod(ParReci recnik[], int n, char rec[], char prevod
    []){
    int i;

20
    /* Pretražuje se rečnik i traži se zadata rec. */
22    for(i=0; i<n; i++){
        {
24            if(strcmp(recnik[i].sr, rec)==0)
            {
26                strcpy(prevod, recnik[i].en);
                return;
28            }
        }
    }
30 }

```

```

32  /* Ukoliko rec nije pronadjena, formira se prevod reci koji se
    sastoji od zvezdica. */
33  for(i=0; rec[i]; i++){
34      prevod[i]='*';
35  }
36  prevod[i]='\0';
37  }
38
39  int main(){
40      ParReci recnik[MAX_BR_RECII];
41      int n;
42      char sr[MAX_DUZINA+1];
43      char en[MAX_DUZINA+1];
44      int i, j, k;
45      char rec[MAX_DUZINA+1];
46      char prevod[MAX_DUZINA+1];
47      char c;
48
49      /* Ucitavaju se parovi reci sa standardnog ulaza sve do kraja ulaza
    . */
50      i=0;
51      while(scanf("%s %s", sr, en)!=EOF){
52          if(i==MAX_BR_RECII)
53              break;
54
55          strcpy(recnik[i].sr, sr);
56          strcpy(recnik[i].en, en);
57
58          i++;
59      }
60      /* Broj parova reci se cuva u promenljivoj n. */
61      n=i;
62
63      printf("Unesite recenicu za prevod: \n");
64      do
65      {
66          /* Ucitava se rec po rec date recenice i pronalazi se njen prevod
    . */
67          scanf("%s", rec);
68
69          pronadji_prevod(recnik, n, rec, prevod);
70          printf("%s ", prevod);
71
72          /* Ukoliko je karakter iza reci znak za novi red, onda se prekida
    sa unosom, a ako nije
    * ucitava se sledeca recenica.
    */
73          c = getchar();
74
75      }while(c != '\n');
76
77
78

```

```
    putchar('\n');
80     return 0;
82 }
```

Rešenje 0.1.6

```
1  #include <stdio.h>
3  #define MAX_PRODAVNICA 50
4  #define DUZINA_RECIP 21
5
6  typedef struct
7  {
8      char prodavnica[DUZINA_RECIP];
9      double cena;
10 }podatak;
11
12 int main()
13 {
14     podatak niz[MAX_PRODAVNICA];
15     double zeljena;
16     int n, i;
17
18     printf("Uneti broj prodavnica: ");
19     scanf("%d", &n);
20
21     if (n <= 0 || n > MAX_PRODAVNICA)
22     {
23         printf("Neispravan broj prodavnica.\n");
24         return -1;
25     }
26
27     for(i=0; i<n; i++)
28     {
29         scanf("%s%lf", niz[i].prodavnica, &niz[i].cena);
30
31         if (niz[i].cena <= 0)
32         {
33             printf("Neispravna cena.\n");
34             return -1;
35         }
36     }
37
38     printf("Uneti zeljenu cenu: ");
39     scanf("%lf", &zeljena);
40
41     printf("Povoljne prodavnice su:\n");
42     for(i=0; i<n; i++)
43     {
44         if (niz[i].cena <= zeljena)
45             printf("%s\n", niz[i].prodavnica);
46     }
```

```
45     return 0;
47 }
```

Rešenje 0.1.7

```
#include <stdio.h>

2  #define MAX_DECE 200

4  typedef struct
6  {
8      char pol;
9      int broj_godina;
10     int ocena;
11 }dete;

12 int main()
13 {
14     int n, i, broj_godina;
15     dete niz[MAX_DECE];
16     char blanko, pol;
17     int suma, broj_dece;

18     printf("Uneti broj dece: ");
19     scanf("%d", &n);

20     if (n <= 0 || n > MAX_DECE)
21     {
22         printf("Neispravan broj dece.\n");
23         return -1;
24     }

25     printf("Uneti podatke za svako dete, pol, broj godina i ocenu:\n");
26     for(i=0; i<n; i++)
27     {
28         scanf("%c%c%d%d", &blanko, &niz[i].pol, &niz[i].broj_godina, &niz[i].ocena);

29         /* Ispitivanje pogresnog unosa. */
30         if (niz[i].pol != 'm' && niz[i].pol != 'z')
31         {
32             printf("Neispravan pol.\n");
33             return -1;
34         }
35         if (niz[i].broj_godina > 6 || niz[i].broj_godina < 3)
36         {
37             printf("Neispravan broj godina.\n");
38             return -1;
39         }
40         if (niz[i].ocena < 1 || niz[i].ocena > 5)
```

```

46     {
47         printf("Neispravna ocena.\n");
48         return -1;
49     }
50 }
51
52 printf("Uneti pol i broj godina: ");
53 scanf("%c%c%d", &blanko, &pol, &broj_godina);
54
55 /* Ispitivanje ispravnosti unetih podataka. */
56 if (pol != 'm' && pol != 'z')
57 {
58     printf("Neispravan pol.\n");
59     return -1;
60 }
61 if (broj_godina > 6 || broj_godina < 3)
62 {
63     printf("Neispravan broj godina.\n");
64     return -1;
65 }
66
67 suma = 0;
68 broj_dece = 0;
69
70 for(i=0; i<n; i++)
71     if (niz[i].pol == pol && niz[i].broj_godina == broj_godina)
72     {
73         suma += niz[i].ocena;
74         broj_dece++;
75     }
76
77 if (broj_dece == 0)
78     printf("Ne postoje deca sa takvim karakteristikama.\n");
79 else
80     printf("Prosečna ocena je: %.3lf.\n", (double)suma/broj_dece);
81
82 return 0;
83 }

```

Rešenje 0.1.8

```

1  #include <stdio.h>
2  #include <math.h>
3
4  #define MAX 50
5
6  typedef struct lopta {
7      int poluprecnik;
8      enum {plava, zuta, crvena, zelena} boja;
9  } LOPTA;

```

```

11 float zapremina(LOPTA l) {
12     return pow(l.poluprecnik, 3)*4/3*M_PI;
13 }

15
16
17 float ukupna_zapremina(LOPTA lopte[], int n) {
18
19     int i;
20     float z = 0;
21
22     for(i = 0; i < n; i++)
23         z += zapremina(lopte[i]);
24
25     return z;
26 }
27
28 /*
29 Funkcija je opstija od trazene i broji sve lopte odredjene boje u
30 nizu lopti.
31 U zavisnosti od prosledjene boje funkciji, funkcija vraca
32 odgovarajuci broj.
33 */
34 int broj_lopti_u_boji(LOPTA lopte[], int n, int boja) {
35
36     int br = 0;
37     int i;
38     for(i = 0; i < n; i++)
39         if(lopte[i].boja == boja)
40             br++;
41     return br;
42 }
43
44 int main() {
45
46     LOPTA lopte[MAX];
47     int n;
48     int i;
49     int boja;
50
51     printf("Unesite broj lopti: ");
52     scanf("%d", &n);
53
54     if(n < 1 || n > MAX) {
55
56         printf("Nekorektan unos.\n");
57         return 0;
58     }
59
60     printf("Unesite dalje poluprecnike i boje lopti (1-plava, 2-zuta,
61         3-crvena, 4-zelena):\n");
62     for(i = 0; i < n; i++) {

```

```

61     printf("%d. lopta: ", i+1);
    scanf("%d%d", &lopte[i].poluprecnik, &boja);

63     /* U zavisnosti od unetog celog broja,
        bira se boja lopte.
65     */
    switch(boja) {
67
        case 1: lopte[i].boja = plava; break;
69         case 2: lopte[i].boja = zuta; break;
        case 3: lopte[i].boja = crvena; break;
71         case 4: lopte[i].boja = zelena; break;
        default:
73             printf("Nekorektan unos.\n");
            return 0;
75     }
    }

77     printf("Ukupna zapremina: %.2f\n", ukupna_zapremina(lopte, n));

79     printf("Ukupno crvenih lopti: %d\n", broj_lopti_u_boji(lopte, n,
        crvena));

81     return 0;
83 }

```

Rešenje 0.1.9

```

1  #include <stdio.h>
   #define MAX 1000

3
   typedef struct
5  {
       char o;
7      int x;
       int y;
9  } IZRAZ;

11
   int korektan_izraz(const IZRAZ izraz)
13  {
       if(izraz.o != '+' && izraz.o != '-' && izraz.o != '*' && izraz.o
           != '/')
15       {
           printf("Nedozvoljena operacija!\n");
17           return 0;
       }
       if(izraz.o == '/' && izraz.y == 0)
19       {
21           printf("Deljenje nulom!\n");
           return 0;

```

```

23     }
24     return 1;
25 }

27 /* Racunanje vrednosti izraza. */
28 int vrednost(const IZRAB izraz)
29 {
30     int v;
31
32     switch (izraz.o)
33     {
34         case '+':
35             return izraz.x + izraz.y;
36         case '-':
37             return izraz.x - izraz.y;
38         case '*':
39             return izraz.x * izraz.y;
40         case '/':
41             return izraz.x / izraz.y;
42     }
43 }

45
46 /*
47  Promenljiva izraz ce se promeniti u funkciji
48  ucitaj_izraz tako sto ce njenim neinicijalizovanim
49  poljima o,x,y biti dodeljene vrednosti ucitane
50  sa ulaza. Zbog toga ovu promenljivu
51  funkciji prosledjujemo po adresi, preko pokazivaca.

52
53  S obzirom da ucitavanje karaktera nije prvo
54  ucitavanje koje se obavlja u programu, funkcijom
55  getchar() se ucita karakter kojim se razdvaja
56  unos karaktera od prethodnog unosa (najcesce blanko
57  znak ili znak za novi red).

58
59 */

60
61 int ucitaj_izraz(IZRAB* izraz)
62 {
63     getchar();
64     scanf("%c%d%d",&izraz->o, &izraz->x, &izraz->y);
65     if (!korektan_izraz(*izraz))
66         return 0;
67     return 1;
68 }

69
70
71 void stampaj_izraz(const IZRAB izraz)
72 {
73     printf("%d %c %d = %d\n", izraz.x, izraz.o, izraz.y, vrednost(
74         izraz));

```



```

}
75
int max_vr(IZRAZ izrazi[], int n)
77
{
    int i;
    int max;

81    max=vrednost(izrazi[0]);

83    for(i=1; i<n; i++)
        if(vrednost(izrazi[i])>max)
85            max=vrednost(izrazi[i]);

87    return max;
}
89
int main()
91
{
    int n;
    IZRAZ izrazi[MAX];
    int max;
    int i;

97    printf("Uneti broj izraza: ");
    scanf("%d", &n);
    if(n<0 || n>MAX)
    {
101        printf("Nekorektna vrednost broja n!\n");
        return -1;
103    }

105
    printf("Uneti izraze u prefiksnoj notaciji:\n");
    for(i=0; i<n; i++)
        if(ucitaj_izraz(&izrazi[i])==0)
109            {
                printf("Nekorektan unos\n");
111                return -1;
            }

113

115    max = max_vr(izrazi, n);
    printf("Maksimalna vrednost izraza:%d\n", max);
117

    printf("Izrazi cija je vrednost manja od polovine maksimalne
        vrednosti:\n");

119
    for(i=0; i<n; i++)
        if(vrednost(izrazi[i])<max/2)
121            stampaj_izraz(izrazi[i]);

123
    return 0;

```

