

PROGRAMIRANJE 1

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

PROGRAMIRANJE 1

Zbirka zadataka

**Beograd
2017.**

Autori:

dr Milena Vujošević Jančić, docent na Matematičkom fakultetu u Beogradu

dr Jovana Kovačević, docent na Matematičkom fakultetu u Beogradu

Danijela Simić, asistent na Matematičkom fakultetu u Beogradu

Anđelka Zečević, asistent na Matematičkom fakultetu u Beogradu

PROGRAMIRANJE 1

Zbirka zadataka

Sadržaj

1	Predstavljanje podataka	vii
1.1	Nizovi	vii
1.2	Rešenja	xxvii

1

Predstavljanje podataka

1.1 Nizovi

Zadatak 1.1.1 Ako su $a = (a_1, \dots, a_n)$ i $b = (b_1, \dots, b_n)$ vektori dimenzije n , njihov skalarni proizvod je $a \cdot b = a_1 \cdot b_1 + \dots + a_n \cdot b_n$. Napisati program koji računa skalarni proizvod dva vektora. Vektori se zadaju kao celobrojni nizovi sa najviše 100 elemenata. Program učitava dimenziju i elemente nizova, a na izlaz ispisuje vrednost skalarnog proizvoda.

Primer 1

```
INTERAKCIJA SA PROGRAMOM:
Unesite dimenziju vektora:
5
Unesite koordinate vektora a:
8 -2 0 2 4
Unesite koordinate vektora b:
35 12 5 -6 -1
Skalarni proizvod vektora a i b:
240
```

Primer 2

```
INTERAKCIJA SA PROGRAMOM:
Unesite dimenziju vektora:
3
Unesite koordinate vektora a:
-1 0 1
Unesite koordinate vektora b:
5 5 5
Skalarni proizvod vektora a i b:
0
```

Primer 3

```
INTERAKCIJA SA PROGRAMOM:
Unesite dimenziju vektora:
120
Greska: Nedozvoljena vrednost!
```

[Rešenje 1.1.1]

Zadatak 1.1.2 Napisati program koji za učitani niz ispisuje:

(a) elemente niza koji se nalaze na parnim pozicijama.

(b) parne elemente niza.

Maksimalan broj elemenata niza je 100.

Primer 1

```
INTERAKCIJA SA PROGRAMOM:
Unesite dimenziju niza:
6
Unesite elemente niza:
1 8 2 -5 -13 75
Elementi niza na parnim pozicijama:
1 2 -13
Parni elementi niza:
8 2
```

Primer 2

```
INTERAKCIJA SA PROGRAMOM:
Unesite dimenziju niza:
3
Unesite elemente niza:
11 81 -63
Elementi niza na parnim pozicijama:
11 -63
Parni elementi niza:
```

Primer 3

```
INTERAKCIJA SA PROGRAMOM:
Unesite dimenziju niza:
-4
Greska: Nedozvoljena vrednost!
```

[Rešenje 1.1.2]

Zadatak 1.1.3 Takmičari na Beogradskom maratonu su označeni rednim brojevima počevši od 0, a vreme za koje su istrčali maraton je dato minutima. Ovi podaci zadati su nizom celih brojeva, pri čemu indeks niza označava redni broj takmičara, a vrednosti u nizu označavaju vreme trčanja. Napisati funkcije za rad sa ovim nizom.

- (a) Napisati funkciju `void ucitaj(int a[], int n)` koja učitava elemente niza `a` dimenzije `n`.
- (b) Napisati funkciju `void stampaj(int a[], int n)` koja štampa elemente niza `a` dimenzije `n`.
- (c) Napisati funkciju `int suma(int a[], int n)` koja računa i vraća ukupno vreme trčanja svih takmičara.
- (d) Napisati funkciju `float prosek(int a[], int n)` koja računa i vraća prosečno vreme (aritmetičku sredinu) trčanja takmičara.
- (e) Napisati funkciju `int maksimum(int a[], int n)` koja izračunava i vraća najduže vreme trčanja takmičara.

- (f) Napisati funkciju `int pozicija_minimum(int a[], int n)` koja vraća redni broj pobjednika Beogradskog maratona, tj. onog takmičara koji je najkraće trčao. U slučaju da ima više takvih takmičara, vratiti onog sa najmanjim indeksom.

Napisati program koji testira rad zadatih funkcija. Maksimalan broj takmičara je 1000.

Primer 1

```
INTERAKCIJA SA PROGRAMOM:
Unesite dimenziju niza:
5
19 47 27 34 16
Vreme trcanja takmicara: 19 47 27 34 16
Ukupno vreme: 143
Prosecno vreme trcanja: 28.6
Maksimalno vreme trcanja: 47
Indeks pobjednika: 4
```

Primer 2

```
INTERAKCIJA SA PROGRAMOM:
Unesite dimenziju niza:
-5
Greska: Nedozvoljena vrednost!
```

[Rešenje [1.1.3](#)]

Zadatak 1.1.4 Napisati funkciju `int min_max(int a[], int n)` koja pronalazi indekse najmanjeg i najvećeg elementa u nizu a dimenzije n koristeći samo jedan prolaz kroz niz. Funkcija kao povratnu vrednost vraća manji od ta dva indeksa. Napisati program koji testira ovu funkciju za učitane nizove celih brojeva maksimalne dužine 100 elemenata.

Primer 1

```
INTERAKCIJA SA PROGRAMOM:
Unesite broj elemenata niza:
7
5 8 -4 11 17 89 1
2
```

Primer 2

```
INTERAKCIJA SA PROGRAMOM:
Unesite broj elemenata niza:
3
Unesite elemente niza:
9 11 6
1
```

Primer 3

```
INTERAKCIJA SA PROGRAMOM:
Unesite broj elemenata niza:
-45
Greska: Nedozvoljena vrednost!
```

Zadatak 1.1.5 Napisati funkciju `int prebrojavanje(int a[], int n)` koja izračunava broj elemenata celobrojnog niza a dužine n koji su manji od poslednjeg elementa niza. Napisati i program koji testira rad funkcije. Maksimalan broj elemenata niza je 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 1.1.5]

Zadatak 1.1.6 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. Maksimalan broj elemenata niza je 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
```

[Rešenje 1.1.6]

Zadatak 1.1.7 Elementi niza celih brojeva su podaci o nadmorskim visinama u nekom području sveta. Na kartografskoj mapi su indeksima označene različite tačke, a u nizu je svakom indeksu pridružen neki ceo broj (odnosno nadmorska visina). Napisati funkcije za rad sa ovim nizom.

- (a) Napisati funkciju koja proverava da li niz sadrži zadatu nadmorsku visinu m . Povratna vrednost funkcije je 1 ako je vrednost sadržana u nizu ili 0 ako nije.

- (b) Napisati funkciju koja vraća vrednost prve pozicije na kojoj se nalazi element koji ima nadmorsku visinu m ili -1 ukoliko element nije u nizu.
- (c) Napisati funkciju koja vraća vrednost poslednje pozicije na kojoj se nalazi element koji ima nadmorsku visinu m ili -1 ukoliko element nije u nizu.

Napisati i program koji testira rad napisanih funkcija za uneti broj m . Maksimalan broj elemenata niza je 100.

Primer 1

```
INTERAKCIJA SA PROGRAMOM:
Unesite dimenziju niza:
7
800 1100 -200 1400 -200 1100 800
Ucitani niz: 800 1100 -200 1400 -200 1100 800
Unesite jedan ceo broj:
1100
Niz sadrzi element cija je vrednost 1100.
Indeks njegovog prvog pojavljivanja u nizu je 1.
Indeks njegovog poslednjeg pojavljivanja u nizu je 5.
```

Primer 2

```
INTERAKCIJA SA PROGRAMOM:
Unesite dimenziju niza:
-5
Greska: Nedozvoljena vrednost!
```

[Rešenje 1.1.7]

Zadatak 1.1.8 Marko skuplja sličice za Svetsko prvenstvo u fudbalu. U celobrojnom nizu a se nalaze brojevi onih sličica koje je već sakupio. Marko je primetio da mu se neke sličice ponavljaju i rešio je da ih razmeni sa drugarima. Napisati program koji od datog niza a formira niz b sličica koje se u nizu a pojavljuju više od dva puta. Maksimalan broj elemenata niza je 100.

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 2 1
```

Primer 3

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

[Rešenje 1.1.8]

Zadatak 1.1.9 Sa standardnog ulaza se učitava dimenzija niza, elementi niza i jedan ceo broj k . Napisati program koji štampa indekse elemenata koji su deljivi sa k . Maksimalan broj elemenata niza je 100.

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 1.1.9]

Zadatak 1.1.10 Autobusi su označeni rednim brojevima (počevši od 1) i u nizu se čuva vreme putovanja svakog autobusa u minutima. Međutim, zbog radova na putu između Požege i Užica, svi autobusi koji saobraćaju na tom potezu (autobusi označeni rednim brojevima od k do t) saobraćaju m minuta duže. Uneti potrebne izmene u niz i ispisati elemente niza. Maksimalan broj autobusa je 200.

Primer 1

```
INTERAKCIJA SA PROGRAMOM:
Unesite dimenziju niza:
8
Unesite vreme putovanja:
24 78 13 124 56 90 205 45
Unesite redne brojeve autobusa koji
putuju na potezu Pozega, Uzice:
3 6
Unesite novo vreme:
23
Vreme putovanja nakon izmena:
24 78 36 147 79 113 205 45
```

Primer 2

```
INTERAKCIJA SA PROGRAMOM:
Unesite dimenziju niza:
8
Unesite vreme putovanja:
24 78 13 124 56 90 205 45
Unesite redne brojeve autobusa koji
putuju na potezu Pozega, Uzice:
3 15
Redni brojevi autobusa nisu
u dozvoljenom opsegu.
```

[Rešenje 1.1.10]

Zadatak 1.1.11 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. Maksimalan broj elemenata niza je 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 [1.1.11](#)]

Zadatak 1.1.12 Napisati program koji transformiše uneti niz tako što kvadrira sve negativne elemente niza. Maksimalan broj elemenata niza je 100.

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 [1.1.12](#)]

Zadatak 1.1.13 Napisati funkciju `void kvadriranje(float a[], int n)` koja kvadrira elemente realnog niza a dužine n koji se nalaze na parnim pozicijama. Napisati program koji transformiše na ovaj način uneti niz. Maksimalan broj elemenata niza je 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 [1.1.13](#)]

Zadatak 1.1.14 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. Maksimalan broj elemenata niza je 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 [1.1.14](#)]

Zadatak 1.1.15 Napisati funkciju `int blizu_3(int a[], int n)` koja pronalazi i vraća indeks elementa niza koji je po vrednosti najbliži aritmetičkoj sredini onih elemenata niza koji su deljivi brojem tri. Napisati program koji testira rad funkcije. Maksimalan broj elemenata niza je 200.

Primer 1

```
INTERAKCIJA SA PROGRAMOM:
Unesite broj elemenata niza: 5
Unesite elemente niza:
1 2 3 4 5
3
```

Primer 2

```
INTERAKCIJA SA PROGRAMOM:
Unesite broj elemenata niza: 5
Unesite elemente niza:
3 6 2 4 7
4
```

[Rešenje [1.1.15](#)]

Zadatak 1.1.16 Napisati program koji za učitani ceo broj, ispisuje broj pojavljivanja svake od cifara u zapisu tog broja. UPUTSTVO: Za evidenciju broja pojavljivanja svake cifre pojedinačno, koristiti niz.

Primer 1

```
INTERAKCIJA SA PROGRAMOM:
Unesite ceo broj:
2355623
U zapisu broja 2355623, cifra 2 se pojaviljuje 2 puta
U zapisu broja 2355623, cifra 3 se pojaviljuje 2 puta
U zapisu broja 2355623, cifra 5 se pojaviljuje 2 puta
U zapisu broja 2355623, cifra 6 se pojaviljuje 1 puta
```

Primer 2

```
INTERAKCIJA SA PROGRAMOM:
Unesite ceo broj:
-39902
U zapisu broja -39902, cifra 0 se pojaviljuje 1 puta
U zapisu broja -39902, cifra 2 se pojaviljuje 1 puta
U zapisu broja -39902, cifra 3 se pojaviljuje 1 puta
U zapisu broja -39902, cifra 9 se pojaviljuje 2 puta
```

[Rešenje [1.1.16](#)]

Zadatak 1.1.17 Napisati funkciju `int cifre(char s[], int n)` koja izračunava broj cifara u nizu karaktera *a* dužine *n*. Napisati program koji za karaktere koji se unose u zasebnim redovima ispisuje broj unetih cifara. Maksimalan broj elemenata niza je 100.

Primer 1

```
INTERAKCIJA SA PROGRAMOM:
Unesite broj elemenata niza: 5
Unesite elemente niza:
4
+
A
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 1.1.17]

Zadatak 1.1.18 Napisati program koji učitava karaktere sa standardnog ulaza sve do kraja ulaza i izračunava koliko se puta u unetom tekstu pojavila svaka od cifara, svako malo slovo i svako veliko slovo. Ispisati broj pojavljivanja samo za karaktere koji su se u unetom tekstu pojavili barem jednom. UPUTSTVO: *Za evidenciju broja pojavljivanja cifara, malih i velikih slova koristiti pojedinačne nizove.*

Primer 1

```
INTERAKCIJA SA PROGRAMOM:
123 abcabcabc 123
Karakter 1 se pojavljuje 2 puta
Karakter 2 se pojavljuje 2 puta
Karakter 3 se pojavljuje 2 puta
Karakter a se pojavljuje 3 puta
Karakter b se pojavljuje 3 puta
Karakter c se pojavljuje 3 puta
```

Primer 2

```
INTERAKCIJA SA PROGRAMOM:
Programiranje 1 je zanimljivo!!
Karakter 1 se pojavljuje 1 puta
Karakter a se pojavljuje 3 puta
Karakter e se pojavljuje 2 puta
Karakter g se pojavljuje 1 puta
Karakter i se pojavljuje 3 puta
Karakter j se pojavljuje 3 puta
Karakter l se pojavljuje 1 puta
Karakter m se pojavljuje 2 puta
Karakter n se pojavljuje 2 puta
Karakter o se pojavljuje 2 puta
Karakter r se pojavljuje 3 puta
Karakter v se pojavljuje 1 puta
Karakter z se pojavljuje 1 puta
Karakter P se pojavljuje 1 puta
```

[Rešenje 1.1.18]

Zadatak 1.1.19 Sa standardnog ulaza se unosi jedna linija teksta. Napisati program koji izračunava i ispisuje koliko puta se pojavilo svako od slova engleskog alfabeta u unetom tekstu. Ne praviti razliku između malih i velikih slova.

Primer 1

```
INTERAKCIJA SA PROGRAMOM:
Tasi, tasi, TaNaNa i SVILENA marama....
a:9 b:0 c:0 d:0 e:1 f:0 g:0 h:0 i:4 j:0 k:0 l:1 m:2
n:3 o:0 p:0 q:0 r:1 s:3 t:3 u:0 v:1 w:0 x:0 y:0 z:0
```

Primer 2

```
INTERAKCIJA SA PROGRAMOM:
Mihailo Petrovic Alas (6 Maj 1868 - 8 Jun 1943)
a:4 b:0 c:1 d:0 e:1 f:0 g:0 h:1 i:3 j:2 k:0 l:2 m:2
n:1 o:2 p:1 q:0 r:1 s:1 t:1 u:1 v:1 w:0 x:0 y:0 z:0
```

Primer 3

```
INTERAKCIJA SA PROGRAMOM:
Alan Matison Tjuring (London, 23. jun 1912 - Cesir, 7. jun 1954)
a:3 b:0 c:1 d:1 e:1 f:0 g:1 h:0 i:3 j:3 k:0 l:2 m:1
n:7 o:3 p:0 q:0 r:2 s:2 t:2 u:3 v:0 w:0 x:0 y:0 z:0
```

[Rešenje [1.1.19](#)]

Zadatak 1.1.20 Napisati program koji učitane karaktere (najviše njih 100, učitavaju se sve do pojave karaktera *) 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
```

Primer 3

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

[Rešenje 1.1.20]

Zadatak 1.1.21 Palindrom je tekst koji se isto čita i sa leve i sa desne strane. Napisati funkciju koja proverava da li elementi niza karaktera čine palindrom (zanemariti velika/mala slova). Maksimalan broj elemenata niza je 200.

Primer 1

```
INTERAKCIJA SA PROGRAMOM:  
Unesite dimenziju niza:  
15  
Unesite elemente niza:  
AnaVolimilovana  
Jeste palindrom!
```

Primer 2

```
INTERAKCIJA SA PROGRAMOM:  
Unesite dimenziju niza:  
26  
Unesite elemente niza:  
Zanimljivo je programirati!  
Nije palindrom.
```

[Rešenje 1.1.21]

Zadatak 1.1.22 Napisati program koji učitava dimenziju i elemente niza i štampa niz u kojem su najveći i najmanji element niza razmenili mesta. Ukoliko se najmanji ili najveći element više puta pojavljuju u nizu, uzeti u obzir njihova prva pojavljivanja. Maksimalan broj elemenata niza je 100.

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: Nedozvoljena vrednost!
```

[Rešenje 1.1.22]

Zadatak 1.1.23 Korišćenjem nizova moguće je predstaviti skupove podataka. Napisati program koji demonstrira osnovne operacije nad skupovima — unija, presek i razlika. Pomoću dva niza predstaviti dva skupa celih brojeva. Maksimalan broj elemenata niza je 500.

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 1.1.23]

Zadatak 1.1.24 Napisati program koji za dva učitana niza a i b dimenzije n formira i na izlaz ispisuje niz c koji se dobija naizmeničnim raspoređivanjem elemenata nizova a i b , tj. $c = [a_0, b_0, a_1, b_1, \dots, a_{n-1}, b_{n-1}]$. Maksimalan broj elemenata niza je 100.

Primer 1

```
INTERAKCIJA SA PROGRAMOM:
Unesite dimenziju nizova:
5
Unesite elemente niza a:
2 -5 11 4 8
Unesite elemente niza b:
3 3 9 -1 17
Rezultujući niz:
2 3 -5 3 11 9 4 -1 8 17
```

Primer 2

```
INTERAKCIJA SA PROGRAMOM:
Unesite dimenziju nizova:
105
Greska: Nedozvoljena vrednost!
```

[Rešenje 1.1.24]

Zadatak 1.1.25 Napisati program koji za dva učitana niza a i b dimenzije n formira i na izlaz ispisuje niz c čija prva polovina odgovara elementima niza b , a druga polovina elementima niza a , tj. $c = [b_0, b_1, \dots, b_{n-1}, a_0, a_1, \dots, a_{n-1}]$. Maksimalan broj elemenata niza je 100.

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: Nedozvoljena vrednost!
```

[Rešenje 1.1.25]

* **Zadatak 1.1.26** Sa standardnog ulaza se učitava ceo broj n manji od 100 i elementi dvaju celobrojnih, sortiranih neopadajuće nizova a i b dimenzije n . Napisati program koji formira i ispisuje niz c koji se dobija spajanjem nizova a i b u treći, takođe sortiran neopadajuće, niz.

Primer 1

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Uneti broj elemenata niza: 5  
|| Uneti elemente sortiranog niza:  
|| 2 11 28 40 63  
|| Uneti elemente sortiranog niza:  
|| -19 -5 5 11 52  
|| Niz c:  
|| -19 -5 2 5 11 11 28 40 52 63
```

Primer 2

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Uneti broj elemenata niza: 3  
|| Uneti elemente sortiranog niza:  
|| -2 4 8  
|| Uneti elemente sortiranog niza:  
|| 6 15 19  
|| Niz c:  
|| -2 4 6 8 15 19
```

Primer 3

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Uneti broj elemenata niza: 145  
|| Greska: Nedozvoljena vrednost!
```

[Rešenje 1.1.26]

Zadatak 1.1.27 Napisati program koji sa standardnog ulaza učitava 10 celih brojeva i razdvaja ih na parne i neparne tako što parne brojeve upisuje na početak niza, a neparne brojeve na kraj niza. Ispisati niz dobijen na ovaj način. NAPOMENA: *Nije dozvoljeno koristiti pomoćne nizove.*

Primer 1

```
INTERAKCIJA SA PROGRAMOM:
Unesite 10 brojeva:
-2 8 11 53 59 20 17 -8 3 14
Rezultujući niz:
14 142 -6 -278 28 34 33 -69 -9 9
```

Primer 2

```
INTERAKCIJA SA PROGRAMOM:
Unesite 10 brojeva:
9 142 -9 -278 -69 33 34 28 -6 14
Rezultujući niz:
-2 8 14 -8 20 59 17 53 3 11
```

[Rešenje 1.1.27]

Zadatak 1.1.28 Napisati funkcije za rad sa nizovima celih brojeva.

- (a) Napisati funkciju koja obrće elemente niza.
- (b) Napisati funkciju koja rotira niz ciklično za jedno mesto u levo.
- (c) Napisati funkciju koja rotira niz ciklično za k mesta u levo.

Napisati i program koji testira rad napisanih funkcija za uneti broj m . Maksimalan broj elemenata niza je 100.

Primer 1

```
INTERAKCIJA SA PROGRAMOM:
Unesite dimenziju niza:
6
7 -3 11 783 26 -19
Elementi niza nakon obrtanja:
-17 28 785 13 -1 9
Elementi niza nakon rotiranja za 1 mesto ulevo:
28 785 13 -1 9 -17
Unesite jedan pozitivan ceo broj:
3
Elementi niza nakon rotiranja za 3 mesto ulevo:
-1 9 -17 28 785 13
```

Primer 2

```
INTERAKCIJA SA PROGRAMOM:
Unesite dimenziju niza:
252
Greska: Nedozvoljena vrednost!
```

[Rešenje 1.1.28]

Zadatak 1.1.29 Prilikom ulaska u banku klijent dobija redni broj, a u nizu se čuva redosled opsluživanja klijenata. Tako, prvi klijent u nizu će biti prvi uslužen, a klijent koji je poslednji dosao se nalazi na kraju niza. Redni brojevi se izdaju počevši od 1 svakog radnog dana, ali se niz za redosled stalno menja. Dodatno, postoje specijalni klijenti (npr. oni koji plaćaju platni promet ili oni koji podižu stambeni kredit) koji mogu dobiti i negativan redni broj da bi se razlikovali od uobičajenih usluga koje banka omogućava. Pomozite radniku obezbeđenja da lako prati redosled opsluživanja klijenata.

- (a) Napisati funkciju koja ubacuje datog klijenta sa rednim brojem x na kraj niza.
- (b) Napisati funkciju koja ubacuje datog klijenta sa rednim brojem x na početak niza (lica sa posebnim potrebama, trudnice, stara lica i ostale ugrožene kategorije).
- (c) Napisati funkciju koja ubacuje datog klijenta sa rednim brojem x na datu poziciju k (manje prioritetna lica, recimo službena lica ili roditelji sa decom, poziciju k bira radnik obezbeđenja).
- (d) Napisati funkciju koja izbacuje prvi element niza (usluženi klijent).
- (e) Napisati funkciju koja izbacuje poslednji element niza (klijent je odustao jer je shvatio da ima mnogo klijenata ispred njega).
- (f) Napisati funkciju koja izbacuje element sa date pozicije k (klijent je odustao jer je dugo čekao).

Napisati program koji testira rad funkcija. Maksimalan broj klijenata u jednom danu je 2000.

Primer 1

```

INTERAKCIJA SA PROGRAMOM:
Unesite trenutni broj klijenata:
8
Unesite niz sa rednim brojevima klijenata:
2 5 -2 16 33 19 8 11
Unesite klijenta kojeg treba ubaciti u niz:
35
Niz nakon ubacivanja klijenta: 2 5 -2 16 33 19 8 11 35
Unesite prioritetnog klijenta kojeg treba ubaciti u niz:
36
Niz nakon ubacivanja klijenta: 36 2 5 -2 16 33 19 8 11 35
Unesite prioritetnog klijenta kojeg treba ubaciti u niz i njegovu poziciju:
-6 2
Niz nakon ubacivanja klijenta: 36 2 -6 5 -2 16 33 19 8 11 35
Niz nakon odlaska klijenta: 2 -6 5 -2 16 33 19 8 11 35
Niz nakon odlaska poslednjeg klijenta: 2 -6 5 -2 16 33 19 8 11
Unesite redni broj klijenta koji je napustio red:
-2
Niz nakon odlaska klijenta: 2 -6 5 16 33 19 8 11

```

[Rešenje 1.1.29]

Zadatak 1.1.30 Napisati program koji za učitani niz formira i ispisuje niz koji se dobija izbacivanjem svih neparnih elemenata niza. Zadatak rešiti na dva načina: korišćenjem pomoćnog niza i transformacijom polaznog niza. Maksimalan broj elemenata niza je 100.

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 1.1.30]

Zadatak 1.1.31 Napisati program koji učitava dimenziju n celobrojnog niza a i njegove elemente, i iz niza a izbacuje sve elemente koji nisu deljivi svojom poslednjom cifrom. Izuzetak su elementi čija je poslednja cifra 0 koje treba zadržati. Program treba da ispiše izmenjeni niz na standardni izlaz. Maksimalan broj elemenata niza je 100.

Primer 1

```
INTERAKCIJA SA PROGRAMOM:
Unesite dimenziju niza:
9
Unesite elemente niza a:
173 -25 23 7 17 25 34 61 -4612
Niz a nakon izmene:
-25 7 25 61 -4612
```

Primer 2

```
INTERAKCIJA SA PROGRAMOM:
Unesite dimenziju niza:
0
Greska: Nedozvoljena vrednost!
```

[Rešenje 1.1.31]

Zadatak 1.1.32 Napisati program koji u nizu dužine n čiji se elementi učitavaju sa ulaza eliminiše sve brojeve koji nisu deljivi svojim indeksom. Niz reorganizovati tako da nema *rupa* koje su nastale eliminacijom elemenata i ispisati na standardni izlaz. Maksimalan broj elemenata niza je 700. Ne razmatrati da li je u novom nizu, nakon brisanja i pomeranja, element deljiv svojim indeksom. NAPOMENA: *Nulti element niza treba zadržati jer nije dozvoljeno deljenje nulom.*

Primer 1

```
INTERAKCIJA SA PROGRAMOM:
Unesite broj elemenata niza: 10
Unesite elemente niza:
4 2 1 6 7 8 10 2 16 3
Novi niz:
4 2 6 16
```

Primer 2

```
INTERAKCIJA SA PROGRAMOM:
Unesite broj elemenata niza: 10
Unesite elemente niza:
-8 5 10 6 7 10 8 2 16 27
Novi niz:
-8 5 10 6 10 16 27
```

[Rešenje 1.1.32]

Zadatak 1.1.33 Napisati program koji za učitani niz ispisuje niz koji se dobija izbacivanjem svih elemenata koji su prosti brojevi. Zadatak rešiti na dva načina: korišćenjem pomoćnog niza i transformacijom polaznog niza. Maksimalan broj elemenata niza je 100. NAPOMENA: *Broj 1 nije prost.*

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 1.1.33]

Zadatak 1.1.34 Napisati funkciju koja proverava da li su elementi celebnog niza uređeni neopadajuće. Maksimalan broj elemenata niza je 300.

Primer 1

```
INTERAKCIJA SA PROGRAMOM:
Unesite dimenziju niza:
8
Unesite elemente niza:
-40 -8 -8 2 30 30 46 200
Jeste uredjen neopadajuće!
```

Primer 2

```
INTERAKCIJA SA PROGRAMOM:
Unesite dimenziju niza:
4
Unesite elemente niza:
4 23 15 30
Nije uredjen neopadajuće.
```


[Rešenje 1.1.34]

Zadatak 1.1.35 Saki indeks niza označava jedan dan u mesecu, a elementi celobrojnog niza predstavljaju broj artikala koji se prodao tog dana. Naći koliko najduže je iz dana u dan broj prodatih artikala rastao.

Primer 1

```
INTERAKCIJA SA PROGRAMOM:
Unesite dimenziju niza: 30
Unesite broj prodatih artikala:
89 171 112 67 119 36 181 157
49 96 73 116 21 172
140 0 23 71 157 135 11 166 21
56 56 87 103 183 148 174
Duzina najduzeg neopadajućeg
prodavanja je 6.
```

Primer 2

```
INTERAKCIJA SA PROGRAMOM:
Unesite dimenziju niza: 31
Unesite broj prodatih artikala:
215 223 262 95 18 116 334 97
146 146 19 314 270 115 21 40
253 27 210 68 96 175 41 242
98 163 8 218 107 102
Duzina najduzeg neopadajućeg
prodavanja je 3.
```

[Rešenje 1.1.35]

Zadatak 1.1.36 Napisati funkciju koja određuje dužinu najduže serije jednakih uzastopnih elemenata u datom nizu brojeva. Maksimalan broj elemenata niza je 100.

Primer 1

```
INTERAKCIJA SA PROGRAMOM:
Unesite dimenziju niza: 8
Unesite elemente niza:
9 -1 2 2 2 2 80 -200
Duzina najduze serije je 4.
```

Primer 2

```
INTERAKCIJA SA PROGRAMOM:
Unesite dimenziju niza: 8
Unesite elemente niza:
9 9 0 -3 -3 -3 -3 72
Duzina najduze serije je 4.
```

[Rešenje 1.1.36]

Zadatak 1.1.37 Napisati funkciju koja određuje da li se jedan niz javlja kao podniz uzastopnih elemenata drugog niza. Zadatak rešiti na dva načina:

- (a) Elementi moraju da budu uzastopni.
- (b) Elementi ne moraju da budu uzastopni, ali je redosled pojavljivanja isti.

Maksimalan broj elemenata niza je 100.

Primer 1

```
INTERAKCIJA SA PROGRAMOM:
Unesite dimenziju niza: 8
Unesite elemente niza:
-4 2 7 90 -22 15 14 7
Unesite dimenziju niza: 4
Unesite elemente niza: 90 -22 15 14
Elementi drugog niza cine
uzastopni podniz prvog niza.
Elementi drugog niza cine
podniz prvog niza.
```

Primer 2

```
INTERAKCIJA SA PROGRAMOM:
Unesite dimenziju niza: 8
Unesite elemente niza:
-4 2 7 90 -22 15 14 7
Unesite dimenziju niza: 4
Unesite elemente niza: 2 7 15 7
Elementi drugog niza ne cine
uzastopni podniz prvog niza.
Elementi drugog niza cine
podniz prvog niza.
```

Primer 3

```
INTERAKCIJA SA PROGRAMOM:
Unesite dimenziju niza: 8
Unesite elemente niza:
-4 2 7 90 -22 15 14 7
Unesite dimenziju niza: 4
Unesite elemente niza: 90 -22 200 1
Elementi drugog niza ne cine
uzastopni podniz prvog niza.
Elementi drugog niza ne
cine podniz prvog niza.
```

Primer 4

```
INTERAKCIJA SA PROGRAMOM:
Unesite dimenziju niza: 8
Unesite elemente niza:
-4 2 7 90 -22 15 14 7
Unesite dimenziju niza: 1
Unesite elemente niza: 90
Elementi drugog niza cine
uzastopni podniz prvog niza.
Elementi drugog niza cine
podniz prvog niza.
```

[Rešenje 1.1.37]

Zadatak 1.1.38 Za celobrojni niz a dimenzije n kažemo da je *permutacija* ako sadrži sve brojeve od 1 do n .

- (a) Napisati funkciju `void brojanje(int a[], int b[], int n)` koja na osnovu celobrojnog niza a dimenzije n formira niz b tako što i -ti element niza b odgovara broju pojavljivanja vrednosti i u nizu a .
- (b) Napisati funkciju `int permutacija(int a[], int n)` koja proverava da li je zadati niz permutacija. Funkcija vraća vrednost 1 ako je svojstvo ispunjeno, odnosno 0 ako nije. UPUTSTVO: *Koristiti funkciju brojanje iz tačke (a).*

Napisati program koji sa standardnog ulaza učitava dimenziju niza i elemente niza i ispisuje da li je uneti niz permutacija ili ne. Maksimalan broj elemenata niza je 100.

Primer 1

```
INTERAKCIJA SA PROGRAMOM:
Unesite broj elemenata niza: 5
Unesite elemente niza:
1 5 4 3 2
Uneti niz je permutacija.
```

Primer 2

```
INTERAKCIJA SA PROGRAMOM:
Unesite broj elemenata niza: 6
Unesite elemente niza:
2 3 3 1 1 5
Uneti niz nije permutacija.
```

[Rešenje 1.1.38]

Zadatak 1.1.39 Napisati program koji za dva cela broja x i y koja se učitavaju sa standardnog ulaza proverava da li se zapisuju pomoću istih cifara.
UPUTSTVO: *Rešiti korišćenjem nizova. Pogledati zadatak ??.*

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 1.1.39]

1.2 Rešenja

Rešenje 1.1.1

```
2  #include <stdio.h>  
3  
4  /* Predprocesorska direktiva kojom se definise maksimalni broj  
   * elemenata niza. */  
5  #define MAX 100  
6  
7  int main()  
8  {  
9      int a[MAX];  
10     int b[MAX];  
11     int n;  
12     int i;  
13     int skalarni_proizvod;  
14  
15     /* Ucitava se dimenzija vektora i proverava njena ispravnost. */  
16     printf("Unesite dimenziju vektora: ");  
17     scanf("%d", &n);  
18     if (n<1 || n>100)  
19     {  
20         printf("Nedozvoljena vrednost!\n");  
21         return -1;  
22     }
```

```

22     }

24     /* Ucitavaju se koordinate vektora. */
    printf("Unesite koordinate vektora a: ");
26     for (i=0; i<n; i++)
    {
28         scanf("%d", &a[i]);
    }

30

    printf("Unesite koordinate vektora b: ");
32     for (i=0; i<n; i++)
    {
34         scanf("%d", &b[i]);
    }

36

    /* Izracunava se skalarni proizvod po zadataj formuli. */
38     skalarni_proizvod=0;

40     for (i=0; i<n; i++)
        skalarni_proizvod = skalarni_proizvod + a[i]*b[i];

42

    /* I ispisuje se njegova vrednost. */
44     printf("Skalarni proizvod vektora a i b: %d\n",skalarni_proizvod);

46     return 0;
}

```

Rešenje 1.1.2

```

#include <stdio.h>

2
#define MAX 100

4
int main()
6 {
    int a[MAX];
    int n;
    int i;

10

    /* Ucitava se dimenzija niza i proverava se njena ispravnost. */
12     printf("Unesi dimenziju niza:\n");
    scanf("%d", &n);
14     if (n<1 || n>MAX)
    {
16         printf("Nedozvoljena vrednost!\n");
        return -1;
18     }

20

    /* Ucitavaju se elementi niza. */
    printf("Unesi elemente niza:\n");
22     for (i=0; i<n; i++)

```

```

24     {
        scanf("%d", &a[i]);
    }

26

    /* Ispisuju se elementi niza na parnim pozicijama. */
28    printf("Elementi niza na parnim pozicijama:\n");
    for (i=0; i<n; i+=2)
30    {
        printf("%d ", a[i]);
32    }
    printf("\n");

34

    /* Ispisuju se parni elementi niza. */
36    printf("Parni elementi niza:\n");
    for (i=0; i<n; i++){
38        if (a[i]%2==0){
            printf("%d ", a[i]);
40        }
    }
42    printf("\n");

44

    return 0;
46 }

```

Rešenje 1.1.3

```

#include <stdio.h>

2

#define MAX 1000

4

/* Funkcija koja učitava elemente niza. */
6 void ucitaj(int a[], int n)
{
8     int i;
    for(i=0; i<n; i++)
10     {
        scanf("%d", &a[i]);
12     }
}

14

/* Funkcija koja stampa elemente niza. */
16 void stampaaj(int a[], int n)
{
18     int i;
    for(i=0; i<n; i++)
20         printf("%d ", a[i]);
    printf("\n");
22 }

24 /* Funkciju racuna sumu elemenata niza. */

```

```

26 int suma(int a[], int n)
{
    int i;
28     int s=0;
    for(i=0;i<n;i++)
30         s+=a[i];
    return s;
32 }

34
/* Funkciju koja racuna prosecnu vrednost elemenata niza. */
36 float prosek(int a[], int n)
{
    int i;
38     int s = suma(a,n);
    return (float) s/n;
40 }
42

44 /* Funkciju koja izracunava maksimum elemenata niza.*/
int maksimum (int a[],int n)
46 {
    int m;
48     int i;

    /* Maksimum se inicijalizuje prvim elementom niza (a[0]), a zatim
       se prolazi kroz ostatak niza. U svakom koraku se poredi vrednost
       maksimuma sa tekucim elementom niza. */
50     m = a[0];
    for(i=1;i<n;i++)
52         if (a[i] > m)
54             m = a[i];

    /* Vraca se izracunata vrednost maksimuma. */
56     return m;
58 }

60
/* Funkciju koja izracunava poziciju maksimalnog elementa u nizu. */
62 int pozicija_maksimuma (int a[],int n)
{
    int m;
64     int m_pozicija;
    int i;
66

    /* Minimum se inicijalizuje prvim elementom niza (a[0]) i pamti se
       njegova pozicija (0), a zatim se prolazi kroz ostatak niza. U
       svakom koraku se poredi vrednost minimuma sa tekucim elementom
       niza i ukoliko je potrebno menjaju se vrednosti minimuma i
       njegove pozicije. */
68

70     m = a[0];

```

```

72     m_pozicija=0;
73     for(i=1;i<n;i++)
74         if (a[i] < m)
75         {
76             m = a[i];
77             m_pozicija=i;
78         }
79
80     /* Vraca se izracunata pozicija. */
81     return m_pozicija;
82 }
83
84 int main()
85 {
86     int a[MAX];
87     int n;
88
89     /* Ucitava se dimenzija niza i proverava njena ispravnost. */
90     printf("Unesite dimenziju niza:");
91     scanf("%d",&n);
92     if (n<1 || n>MAX)
93     {
94         printf("Nedozvoljena vrednost!\n");
95         return -1;
96     }
97
98
99     /* Testira se funkcija kojom se ucitavaju elementi niza. */
100    ucitaj(a,n);
101
102    /* Testira se funkcija kojom se ispisuju elementi niza. */
103    printf("Vreme trcanja takmicara: ");
104    stampa(a,n);
105
106    /* Testira se funkcija kojom se izracunava suma elemenata niza. */
107    printf("Ukupno vreme: %d\n", suma(a,n));
108
109    /* Testira se funkcija kojom se racuna prosek elemenata niza. */
110    printf("Prosecno vreme trcanja: %.2f\n", prosek(a,n));
111
112    /* Testira se funkcija kojom se izracunava minimum niza. */
113    printf("Maksimalno vreme trcanja: %d\n", minimum(a,n));
114
115    /* Testira se funkcija kojom se izracunava pozicija maksimalnog
116       elementa. */
117    printf("Indeks pobednika: %d\n", pozicija_maksimuma(a,n));
118
119    return 0;
120 }

```

Rešenje 1.1.4

Rešenje 1.1.5

```
1  #include <stdio.h>
2
3  #define MAX 100
4
5  /* Funkcija prebrojavanje vraca broj elemenata niza koji su manji od
6     poslednjeg elementa. */
7  int prebrojavanje(int a[], int n)
8  {
9      int i;
10
11     /* Brojac elemenata koji su manji od poslednjeg. */
12     int broj_manjih=0;
13
14     /* Obilazi se element po element niza, */
15     for(i=0;i<n-1;i++){
16         /* Ako je tekuci element manji od poslednjeg (on se nalazi na
17            poziciji n-1) uvecava se brojac. */
18         if(a[i]<a[n-1])
19             broj_manjih++;
20     }
21
22     /* Vraca se izracunata vrednost. */
23     return broj_manjih;
24 }
25
26 int main()
27 {
28     int a[MAX];
29     int n;
30     int i;
31
32     /* Ucitava se broj elemenata niza i proverava se njegova ispravnost
33        . */
34     printf("Unesite broj elemenata niza:");
35     scanf("%d", &n);
36     if(n<=0 || n>MAX)
37     {
38         printf("Greska: Nedozvoljena vrednost!\n");
39         return 0;
40     }
41
42     /* Ucitavaju se elementi niza. */
43     printf("Unesite elemente niza:");
44     for(i=0;i<n;i++)
45         scanf("%d",&a[i]);
46
47     /* Ispisuje se rezultat poziva funkcije. */
```



```

46     printf("%d\n", prebrojavanje(a,n));
48     return 0;
}

```

Rešenje 1.1.6

```

#include <stdio.h>
2
#define MAX 100
4
/* Funkcija vraca broj parnih elemenata niza koji prethode
   maksimalnom elementu niza. */
6 int prebrojavanje(int a[], int n)
{
8     int i;

10     int maksimum;
    int pozicija_maksimuma;

12     /* Brojac elemenata koji su parni i prethode maksimalnom. */
14     int broj_parnih;

16     /* Pronalazi se maksimalni element niza i njegova pozicija. */
    maksimum = a[0];
    pozicija_maksimuma = 0;

20     for(i=1;i<n-1;i++)
        if(a[i]> maksimum)
22     {
            maksimum = a[i];
            pozicija_maksimuma = i;
24     }

26     /* Prebrojavaju se parni elementi koji prethode maksimalnom. */
    broj_parnih = 0;
    for(i=0; i < pozicija_maksimuma; i++){
30         if(a[i]%2==0){
            broj_parnih++;
32         }
    }

34     /* Vraca se izracunata vrednost. */
36     return broj_parnih;
}

38 int main()
40 {
    int a[MAX];
42     int n;
    int i;

```

```

44      /* Ucitava se broj elemenata niza i proverava se njegova ispravnost
      . */
46      printf("Unesite broj elemenata niza:");
      scanf("%d", &n);
48      if(n<=0 || n>MAX)
      {
49          printf("Greska: Nedozvoljena vrednost!\n");
          return 0;
52      }

54      /* Ucitavaju se elementi niza. */
      printf("Unesite elemente niza:");
56      for(i=0;i<n;i++){
          scanf("%d",&a[i]);
58      }

60      /* Ispisuje se rezultat poziva funkcije. */
      printf("%d\n", prebrojavanje(a,n));
62      return 0;
64  }

```

Rešenje 1.1.7

```

#include <stdio.h>
2
#define MAX 100
4
/* Funkcija koja ucitava elemente niza. */
6 void ucitaj(int a[], int n)
{
8     int i;
    for(i=0;i<n;i++)
10     {
        scanf("%d",&a[i]);
12     }
}

14
/* Funkcija koja ispisuje elemente niza. */
16 void stampaj(int a[], int n)
{
18     int i;
    for(i=0;i<n;i++)
20     printf("%d ",a[i]);
    printf("\n");
22 }

24
/* Funkcija koja proverava da li niz sadrzi zadatu vrednost m. */
26 int sadrzi(int a[], int n, int m)

```

```

{
28     int i;

30     /* Poredi se element po element niza a sa zadatim brojem m. */

32     for(i=0;i<n;i++){
        /* Ukoliko je tekuci element niza jednak traženom broju funkcija
           vraca vrednost 1. */
34         if (a[i]==m)
            return 1;
36     }

38     /* Ako se stigne do kraja niza i ne naidje na vrednost koja je
        jednaka broju m, to znaci da se broj ne nalazi u nizu i da
        funkcija treba da vrati 0. */
        return 0;
40 }

42 /* Funkcija koja vraca vrednost prve pozicije na kojoj se nalazi
    element koji ima vrednost m, ili -1 ukoliko element nije u nizu.
    */
    int prvo_pojavljivanje(int a[], int n, int m)
44 {
        int i;

46         /* Poredi se element po element niza a sa zadatim brojem m. */

48         for(i=0;i<n;i++){
            /* Ukoliko je tekuci element niza jednak traženom broju vraca se
               njegov indeks. */
50             if (a[i]==m)
                return i;
52         }

54         /* Ako se stigne do kraja niza i ne naidje na vrednost koja je
            jednaka broju m, to znaci da se broj ne nalazi u nizu i da
            funkcija treba da vrati -1. */
56         return -1;
58     }

    /* Funkcija koja vraca vrednost poslednje pozicije na kojoj se nalazi
       element koji ima vrednost m, ili -1 ukoliko element nije u nizu.
       */
60     int poslednje_pojavljivanje(int a[], int n, int m)
    {
62         int i;

64         /* Polazi se od kraja niza i poredi se element po element sa
            zadatim brojem m. */

66         for(i=n-1;i>=0;i--){
            /* Ukoliko je tekuci element niza jednak traženom broju. */

```

```

68         if (a[i]==m){
69             /* Vraca se njegov indeks. */
70             return i;
71         }
72     }
73
74     /* Ako se stigne do pocetka niza i ne naidje na vrednost koja je
75     jednaka broju m, to znaci da se broj ne nalazi u nizu i da
76     funkcija treba da vrati -1. */
77     return -1;
78 }
79
80 int main()
81 {
82     int a[MAX];
83     int n;
84     int m;
85     int i;
86
87     /* Ucitava se dimenzija niza i proverava se njena ispravnost. */
88     printf("Unesite dimenziju niza:");
89     scanf("%d",&n);
90     if (n<1 || n>MAX)
91     {
92         printf("Nedozvoljena vrednost!\n");
93         return -1;
94     }
95
96     /* Ucitavaju se i ispisuju elementi niza. */
97     ucitaj(a,n);
98     printf("Ucitani niz:");
99     stampaj(a,n);
100
101     /* Ucitava se vrednost za pretragu. */
102     printf("Unesi jedan ceo broj:");
103     scanf("%d",&m);
104
105     /* Proverava se rad napisanih funkcija. */
106     if(sadrzi(a,n,m))
107         printf("Niz sadrzi element cija je vrednost %d\n", m);
108     else
109         printf("Niz ne sadrzi element cija je vrednost %d\n", m);
110
111     i = prvo_pojavljivanje(a,n,m);
112     if(i!=-1)
113         printf("Indeks njegovog prvog pojavljivanja u nizu je %d\n", m,
114             i);
115
116     i = poslednje_pojavljivanje(a,n,m);
117     if(i!=-1)

```

```

118     printf("Indeks njegovog poslednjeg pojavljivanja u nizu je %d\n", m,i);
120     return 0;
}

```

Rešenje 1.1.8

```

1  #include <stdio.h>
3  #define MAX 100
5  /* Funkcija koja vraca broj pojavljivanja broja x u nizu */
   int broj_pojavljanja(int niz[], int n, int x)
7  {
       int i;
9
       /* Broj pojavljivanja broja x */
11    int brojac = 0;
13
       /* Obilazi se element po element niza */
       for(i=0;i<n;i++){
15         /* Ukoliko je tekuci element jednak trazenom broju */
         if(niz[i] == x){
17             /* Uvecava se broj pojavljivanja */
             brojac++;
19         }
   }
21
   /* Vraca se izracunata vrednost */
23   return rezultat;
   }
25
   int main()
27   {
       /* Niz elemenata koje zadaje korisnik */
29     int a[MAX];
31
       /* Niz elemenata koji se pojavljuju tri puta */
       int b[MAX];
33
       int i, j, n, n_b;
35
       /* Ucitava se broj elemenata korisnickog niza i proverava se
       njegova ispravnost. */
37     printf("Unesite broj n: ");
       scanf("%d", &n);
39     if(n<1 || n>MAX)
       {
41         printf("Greska: Nedozvoljena vrednost!\n");
         return -1;
       }
   }

```

```

43 }

45 /* Ucitavaju se elementi korisnickog niza. */
printf("Unesite elemente niza a: ");
47 for(i=0;i<n;i++)
    scanf("%d", &a[i]);

49

51 /* Parametar j je brojac elemenata rezultujuceg niza b. */
j = 0;

53 /* Obilazi se element po element niza a. */
for(i=0;i<n;i++)
55 {
    /* Ukoliko se tekuci element pojavljuje vise od dva puta u nizu a
       i nije upisan u niz b koji trenutno ima j elemenata, dodaje se u
       niz b na poziciju j i uvecava se broj elemenata niza b. */
57     if(broj_pojavljivanja(a, n, a[i])>=3 && broj_pojavljivanja(b, j,
        a[i])==0)
        {
59             b[j] = a[i];
            j++;
61         }
    }

63

65 /* Ispisuje se rezultujuci niz b, broj elemenata u nizu b je j. */
n_b = j;
for(i=0;i<n_b;i++)
67     printf("%d ", b[i]);
printf("\n");

69

71 return 0;
}

```

Rešenje 1.1.9

```

1  #include <stdio.h>

3  #define MAX 100

5  int main()
{
7      int brojevi[MAX];
      int n, i, k, indikator;

9

11     /* Ucitava se dimenzija niza i proverava se njena ispravnost. */
    printf("Unesite dimenziju niza: ");
    scanf("%d", &n);
13     if(n<1 || n>MAX)
    {
15         printf("Greska: Nedozvoljena vrednost!\n");
        return -1;
    }
}

```

```

17     }

19     /* Ucitavaju se elementi niza. */
    printf("Unesite elemente niza: ");
21     for(i=0;i<n;i++)
        scanf("%d", &brojevi[i]);

23

25     /* Ucitava se broj k i proverava se njegova ispravnost. */
    printf("Unesite broj k: ");
27     scanf("%d", &k);
    if(k == 0)
29     {
        printf("Greska: Pogresan unos!\n");
31         return -1;
    }

33     /* Promenljiva koja cuva informaciju o tome da li je u nizu
        postojao element koji je deljiv brojem k. Inicijalna vrednost je
        0. */

35     indikator = 0;

37     /* Ukoliko je element niza deljiv brojem k, indikator se postavlja
        na 1 i ispisuje se indeks tog elementa. */

39     for(i=0;i<n;i++){
41         if(brojevi[i]%k == 0)
            {
43             indikator = 1;
            printf("%d ",i);
45         }
    }

47     /* Ukoliko je indikator jednak nuli to znaci da ne postoji element
        u nizu koji je deljiv brojem k. */

49     if(indikator == 0){
51         printf("U nizu nema elemenata koji su deljivi brojem %d!\n",k);
    }

53

55     return 0;
}

```

Rešenje 1.1.10

```

#include <stdio.h>

2 /* Indeksiranje pocinje od 1, pa zato maksimalna dimenzija niza mora
    biti 201, a ne 200. */

```

```

4  #define DIM 201

6  int main()
{
8      int n, niz[DIM], i;
      int k, t;
10     int m;

12     /* Ucitavanje dimenzije i elemenata niza. */
    printf("Unesite dimenziju niza: ");
14     scanf("%d", &n);

16     if (n<=0 || n>DIM)
    {
18         printf("Nedozvoljena dimenzija niza.\n");
        return 0;
20     }

22     printf("Unesite vreme putovanja:\n");
    for(i=1; i<=n; i++)
24         scanf("%d", &niz[i]);

26     /* Unos rednih brojeva autobusa. */
    printf("Unesite redne brojeve autobusa koji putuju na potezu Pozega
      , Uzice:\n");
28     scanf("%d%d", &k, &t);

30     if (k<=0 || k>n || t<=0 || t>n)
    {
32         printf("Redni brojevi autobusa nisu u dozvoljenom opsegu.\n");
        return -1;
34     }

36     /* Unos vremena. */
    printf("Unesite novo vreme:\n");
38     scanf("%d", &m);

40     if (m < 0)
    {
42         printf("Vreme ne moze biti negativno.\n");
        return -1;
44     }

46     /* Unos izmena u niz. */
    for(i=k; i<=t; i++)
48         niz[i] += m;

50     /* Ispis niza. */
    printf("Vreme putovanja nakon izmena:");
52     for(i=1; i<=n; i++)
        printf("%d ", niz[i]);
54     printf("\n");

```



```
56     return 0;
    }
```

Rešenje 1.1.11

```
1  #include<stdio.h>
3  #define MAX 100
5  /* Funkcija racuna zbir elemenata niza od pozicije i do pozicije j.
   */
6  int zbir(int a[], int n, int i, int j){
7      int k;
9      /* Zbir elemenata niza iz zadatog opsega. */
10     int z = 0;
11
12     /* Obilaze se elementi niza. */
13     for(k=i; k<=j; k++){
14         z+=a[k];
15     }
16
17     /* Vraca se izracunata vrednost. */
18     return z;
19 }
21 int main(){
22
23     int n, i, j;
24     int a[MAX];
25
26     /* Ucitava se broj elemenata niza i proverava se njegova ispravnost
       . */
27     printf("Unesite broj elemenata niza: ");
28     scanf("%d", &n);
29     if(n <= 0 || n > MAX)
30     {
31         printf("Greska: Nedozvoljena vrednost!\n");
32         return 0;
33     }
34
35     /* Ucitavaju se elementi niza. */
36     printf("Unesite elemente niza:");
37     for(i=0; i<n; i++)
38         scanf("%d", &a[i]);
39
40     /* Ucitavaju se vrednosti granica. */
41     printf("Unesite vrednosti za i i j: ");
42     scanf("%d%d", &i, &j);
43 }
```

```

45  /* Proverava se korektnost zadatog intervala. */
    if(i < 0 || j < 0 || i > n-1 || j > n-1 || i > j){
47      printf("Greska: Nekorektne vrednosti granica!\n");
        return 0;
    }

49

    /* Ispisuje se rezultat poziva funkcije. */
51    printf("Zbir je: %d", zbir(a,n,i,j));

53    return 0;
}

```

Rešenje 1.1.12

```

1  #include <stdio.h>

3  #define MAX 100

5  int main()
{
7      float brojevi[MAX];
        int n, i;

9

    /* Ucitava se dimenzija niza i proverava se njena ispravnost. */
11    printf("Unesite broj elemenata niza: ");
        scanf("%d", &n);
13    if(n<1 || n>MAX)
        {
15        printf("Nedozvoljena vrednost!\n");
            return -1;
17    }

19    /* Ucitavaju se elementi niza. */
        printf("Unesite elemente niza:\n");
21    for(i=0;i<n;i++){
            scanf("%f", &brojevi[i]);
23    }

25    /* Ukoliko je i-ti element niza brojevi[i] negativan broj, kvadrira
        se tako sto se pomnozi sa samim sobom. */
        for(i=0;i<n;i++){
27            if(brojevi[i]<0)
                brojevi[i] *= brojevi[i];
29        }

31    /* Ispisuje se novodobijeni niz. */
        for(i=0;i<n;i++){
33        printf("%g ", brojevi[i]);
        }
35    printf("\n");

```

```
37     return 0;
    }
```

Rešenje 1.1.13

```
1  #include<stdio.h>
3  #define MAX 100
5  /* Funkcija koja kvadrira elemente niza koji se nalaze na parnim
   pozicijama. */
void kvadriranje(float a[], int n){
7
   int i;
9
   /* Obilaze se elementi na parnim pozicijama i kvadriraju se: a[i] =
      a[i]*a[i]. */
11  for(i=0; i<n; i+=2)
      a[i]*=a[i];
13 }
15 int main(){
17     int n, i, j;
    float a[MAX];
19
    /* Ucitava se broj elemenata niza i proverava se njegova ispravnost
       . */
21  printf("Unesite broj elemenata niza: ");
    scanf("%d", &n);
23  if(n <=0 || n>MAX)
    {
25  printf("Greska: Nedozvoljena vrednost!\n");
    return 0;
27  }
29
    /* Ucitavaju se elementi niza. */
    printf("Unesite elemente niza:");
31  for(i=0; i<n; i++)
    scanf("%f", &a[i]);
33
    /* Poziva se funkcija. */
    kvadriranje(a,n);
35
    /* Ispisuju se elementi novodobijenog niza. */
    for(i=0; i<n; i++)
37  printf("%g ", a[i]);
    printf("\n");
41
    return 0;
43 }
```

Rešenje 1.1.14

```
1  #include<stdio.h>
2
3  #define MAX 100
4
5  /* Funkcija racuna zbir prvih k pozitivnih elemenata niza. */
6  float zbir_pozitivnih(float a[], int n, int k){
7
8      int i;
9
10     /* Zbir pozitivnih elemenata. */
11     float zbir=0;
12
13     /* Obilazi se element po element niza. Postupak se završava ukoliko
14        se dodje do kraja niza ili ukoliko se sabere k pozitivnih
15        elemenata. */
16     for(i=0; i<n && k>0; i++){
17         /* Ako je tekuci element pozitivan broj. */
18         if(a[i] >= 0){
19             /* Dodaje se zbiru. */
20             zbir+=a[i];
21             /* Umanjuje se brojac pozitivnih elemenata. */
22             k--;
23         }
24     }
25
26     /* Vraca se izracunata vrednost. */
27     return zbir;
28 }
29
30 int main(){
31     int n, i, k;
32     float a[MAX];
33
34     /* Ucitava se broj elemenata niza i proverava se njegova ispravnost
35        . */
36     printf("Unesite broj elemenata niza: ");
37     scanf("%d", &n);
38     if(n<=0 || n> MAX){
39         printf("Greska: Nedozvoljena vrednost!\n");
40         return 0;
41     }
42
43     /* Ucitavaju se elementi niza. */
44     printf("Unesite elemente niza: ");
45     for(i=0; i<n; i++)
46         scanf("%f", &a[i]);
```

```

46  /* Ucitava se broj k i proverava se njegova ispravnost. */
printf("Unesite vrednost za k: ");
scanf("%d", &k);
48  if(k<0 || k>n){
printf("Greska: Nedozvoljena vrednost!");
50  return 0;
}

52  /* Ispisuje se rezultat poziva funkcije. */
54  printf("Zbir je: %.2f\n", zbir_pozitivnih(a,n,k));

56  return 0;
}

```

Rešenje 1.1.15

```

1  #include <stdio.h>
#include <math.h>
3
#define DIM 200
5
/* Funkcija koja racuna aritmeticku sredinu onih elemenata niza koji
   su deljivi sa 3. */
7  float artmeticka_sredina_3(int niz[], int n)
{
9  /* Brojac u petlji. */
int i;
11 /* Parametar koji predstavlja zbir svih elemenata deljivih sa 3. */
int suma = 0;
13 /* Parametar koji predstavlja brojih elemenata niza koji su deljivi
   sa 3. */
int br_3 = 0;
15
/* U petlji se proverava svaki element niza i za one koji su
   deljivi sa 3 uvecava se suma i odgovarajuci brojac. */
17 for(i=0; i<n; i++)
if (niz[i]%3 == 0)
19 {
suma += niz[i];
21 br_3++;
}

23 return (float)suma/br_3;
25 }

27
int blizu_3(int a[], int n)
29 {
/* Parametar koji predstavlja aritmeticku sredinu onih elemenata
   niza koji su deljivi sa 3. */
31 float art = artmeticka_sredina_3(a, n);

```

```

/* Pretpostavka je da je prvi element niza najblizi izracunatoj
   aritmetickoj sredini. */
33 int element = a[0];
/* Radi brzine izracunavanja pamti se razdaljina izmedju trenutno
   najblizeg elementa i aritmeticke sredine. Izracunava se apsolutna
   vrednost razdaljine jer ona moze biti i negativna. Moze se
   koristiti ugradjena funkcija fabs za racunanje apsolutne
   vrednosti realnih brojeva. */
35 float razdaljina = fabs(art - element);
/* Pomocni parametar koji sluzi da se u petlji pamti tekuca
   razdaljina. */
37 float nova_razdaljina;
/* Brojac u petlji. */
39 int i;

41 /* U petlji se ispituje svaki element niza i proverava se da li
   postoji neki koji je po vrednosti blizi aritmetickoj sredini. */
for(i=1; i<n; i++)
43 {
    nova_razdaljina = fabs(a[i] - art);
45 /* Izracunava se razdaljina izmedju tekeceg elementa niza i
   poredi sa trenutnom razdaljinom. Ako je novodobijena vrednost
   manja vrsi se zamena zapamcenih parametara. */
    if (nova_razdaljina < razdaljina)
47     {
        razdaljina = nova_razdaljina;
49         element = a[i];
    }
51 }

53 return element;
}

55 int main()
57 {
    int niz[DIM], n, i;

59 /* Ucitavanje dimenzije i elemenata niza. */
    printf("Unesite broj elemenata niza: ");
    scanf("%d", &n);

63 if (n<=0 || n>DIM)
65 {
    printf("Nedozvoljena dimenzija niza.\n");
67     return 0;
}

69 printf("Unesite elemente niza: ");
71 for(i=0; i<n; i++)
    scanf("%d", &niz[i]);

73 /* Ispisivanje rezultata. */

```

```
75     printf("Trazeni broj je %d.\n", blizu_3(niz, n));
77     return 0;
}
```

Rešenje 1.1.16

```
1  #include<stdio.h>
   #include<stdlib.h>
3
   int main()
5  {
       int x;
       int brojaci[10];
       char cifra;
       int original;
       int i;
11
       /* Ucitava se ceo broj sa standardnog ulaza. */
13     printf("Unesite ceo broj:\n");
       scanf("%d",&x);
15
       /* Cuva se njegova originalna vrednost zbog finalnog ispisa. */
17     original = x;
19
       /* Nadalje posmatra apsolutna vrednost. */
       x = abs(x);
21
       /* Svaki element niza brojaci predstavljace brojac za jednu od
          cifara:
23         brojac[0] predstavljace broj nula u zapisu broja x
          brojac[1] predstavljace broj jedinica u zapisu broja x
25         ...
          brojac[9] predstavljace broj devetki u zapisu broja x. */
27
29     /* Brojaci se na pocetku inicijalizuju nulama, */
       for(i=0;i<10;i++){
31         brojaci[i]=0;
       }
33
       /* Sve dok ima cifara u zapisu broja x */
35     do
       {
37         /* Izdvaja se krajnja desna cifra. */
           cifra = x%10;
39
           /* Uvecava se njen broj pojavljivanja. */
           brojaci[cifra]++;
41
43         /* I prelazi se na analiziranje sledece cifre. */

```

```

        x/=10;
45     } while(x);
47
    /* Ispisuju se informacije o ciframa koje se nalaze u zapisu broja
       x. */
49     for(i=0; i<10; i++){
        if(brojaci[i]){
51             printf("U zapisu broja %d, cifra %d se pojaviljuje %d puta\n",
                    original, i, brojaci[i]);
                }
53     }
55
    return 0;
57 }

```

Rešenje 1.1.17

```

1  #include <stdio.h>
   #include <ctype.h>
3
   #define MAX 100
5
   /* Funkcija prebrojava cifre u datom nizu karaktera. */
7  int cifre(char a[], int n)
   {
9      int i;
11
   /* Brojac cifara. */
   int broj_cifara = 0;
13
   /* Obilazi se element po element niza. */
15   for(i=0; i<n; i++){
       /* Ako je tekuci element cifra uvecava se broj cifara. */
17       if(isdigit(a[i]))
           broj_cifara++;
19   }
21
   /* Vraca se izracunata vrednost. */
   return broj_cifara;
23 }
25
int main()
{
27     char a[MAX];
    int n;
29     int i;
31
    /* Ucitava se broj elemenata niza i proverava se njegova ispravnost
       . */

```



```

33 printf("Unesite broj elemenata niza:");
scanf("%d", &n);
if(n<=0 || n>MAX)
35 {
    printf("Greska: Nedozvoljena vrednost!\n");
37     return 0;
}

39 /* Ucitavaju se elementi niza. */
41 printf("Unesite elemente niza:");
for(i=0;i<n;i++) {
43     /* Preskace se prethodno uneti znak za novi red. */
    getchar();

45     /* A zatim se ucitava sam karakter i smesta u niz. */
47     scanf("%c",&a[i]);
}

49 /* Ispisuje se rezultat poziva funkcije. */
51 printf("Broj cifara je: %d\n", cifre(a,n));

53 return 0;
}

```

Rešenje 1.1.18

```

#include <stdio.h>

2 /* Funkcija za ispis elemenata niza. */
4 void ispis(int niz[], int n, char c)
{
6     int i;

8     for(i = 0; i < n; i++){
        if (niz[i]!=0)
10         printf("Karakter %c se pojavljuje %d puta\n", c + i, niz[i]);
    }
12 }

14 int main()
{
16     /* Niz u kojem ce se cuvati informacije o broju pojavljivanja
        cifara. */
        int cifre[10];

18     /* Niz u kojem ce se cuvati informacije o broju pojavljivanja malih
        slova. */
20     int mala_slova[26];

22     /* Niz u kojem ce se cuvati informacije o broju pojavljivanja
        velikih slova. */

```

```

24     int velika_slova[26];

26     int c, i;

28     /* Brojaci karaktera se na pocetku inicijalizuju nulama. */
29     for(i=0;i<10;i++){
30         cifre[i]=0;
31     }

32     for(i=0;i<26;i++)
33     {
34         mala_slova[i]=0;
35         velika_slova[i]=0;
36     }

38     /* Ucitavaju se karakteri sve do kraja ulaza. */
39     while((c = getchar()) != EOF)
40     {
41         /* Ako je procitani karakter veliko slovo uvecava se broj
42            pojavljivanja odgovarajuceg velikog slova. */
43         if (c>='A' && c<='Z'){
44             velika_slova[c-'A']++;
45         }
46         else{
47             /* Ako je procitani karakter malo slovo uvecava se broj
48                pojavljivanja odgovarajuceg malog slova. */
49             if (c>='a' && c<='z'){
50                 mala_slova[c-'a']++;
51             }
52             else{
53                 /* Ako je procitani karakter cifra uvecava se broj
54                    pojavljivanja odgovarajuce cifre. */
55                 if(c >='0' && c <= '9'){
56                     cifre[c-'0']++;
57                 }
58             }
59         }

60         /* Ispisuju se trazene informacije. */
61         ispis(cifre, 10, '0');
62         ispis(mala_slova, 26, 'a');
63         ispis(velika_slova, 26, 'A');

64     return 0;
65 }

```

Rešenje 1.1.19

```

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

```

```

4  /* Funkcija za ispis elemenata niza. */
void ispis(int niz[], int n)
6  {
    int i;
8
    for(i = 0; i < n; i++)
10     printf("%c:%d ", 'a' + i, niz[i]);
    putchar('\n');
12 }

14 int main()
{
16     /* Niz u kojem ce se cuvati informacije o broju pojavljivanja malih
        slova. */
    int mala_slova[26];
18     int c, i;

20     for(i=0;i<26;i++)
        mala_slova[i]=0;

22     /* Ucitavaju se karakteri sve do kraja ulaza. */
24     while((c = getchar()) != EOF)
    {
26         /* Ako je procitani karakter slovo broj pojavljivanja slova se
            uvecava. Kako se zanemaruje velicina slova, svako slovo se
            pretvori u malo i potom se element na odgovarajucoj poziciji u
            nizu uveca. */
            if (isalpha(c))
28                 mala_slova[tolower(c)-'a']++;
    }

30     /* Ispisuju se trazene informacije. */
32     ispis(mala_slova, 26);

34     return 0;
}

```

Rešenje 1.1.20

```

1  #include <stdio.h>

3  #define MAX 100

5  int main()
{
7     /* Niz karaktera. */
    char karakteri[MAX];
9     char c;
    int i, n;
11

```

```

13  for(i=0;i<MAX;i++)
14  {
15      /* Ucitava se karakter po karakter sa standardnog ulaza sve dok
16      se ne unese * ili se ne prekorači maksimalni broj karaktera.s */
17      printf("Unesite karakter: ");
18      scanf("%c", &c);

19      /* Cita se znak za novi red nakon unesenog karaktera. */
20      getchar();

21      /* Ukoliko je unet karakter * prekida se dalje citanje i izlazi
22      se iz petlje. */
23      if(c == '*')
24          break;

25      /* Inace, procitani karakter se smesta u niz. */
26      karakteri[i] = c;
27  }

28  /* Broj unetih karaktera je nakon izlaska iz petlje i-1. */
29  n = i-1;

30  /* Ispisuju se karakteri u obrnutom redosledu. */
31  for(i=n;i>=0;i--)
32  {
33      printf("%c ", karakteri[i]);
34  }
35  printf("\n");

36  return 0;
37  }

```

Rešenje 1.1.21

```

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

3  #define DIM 200

4  int main()
5  {
6      int n, i;
7      char niz[DIM], blanko;

8      /* Ucitavanje dimenzije i elemenata niza. */
9      printf("Unesite trenutni broj klijenata: ");
10     scanf("%d", &n);

11     if (n<=0 || n>DIM)

```

```

16 {
17     printf("Nedozvoljena dimenzija niza.\n");
18     return 0;
19 }
20
21 /* Nakon unosa dimenzije, korisnik unosi blanko ili znak za novi
22    red. Ovaj karakter je potrebno učitati pre unosa elemenata niza,
23    inače će ovaj karakter biti prvi element niza, što nije željeno
24    ponašanje. */
25 scanf("%c", &blanko);
26
27 printf("Unesite elemente niza: ");
28 for(i=0; i<n; i++)
29     scanf("%c", &niz[i]);
30
31 /* Provera da li je niz palindrom. Upoređuju se prvi i poslednji,
32    drugi i preposlednji... odnosno i-ti i ((n-1)-i)-ti element niza
33    . Ako nisu jednaki, niz nije palindrom. Kako se istovremeno
34    posmatraju dva elementa niza dovoljno je da brojac u petlji ide
35    do polovine dimenzije niza.*/
36 for(i=0; i<n/2; i++)
37     /* Kako se zanemaruje veličina slova, svaki element niza se
38        pretvara u malo slovo i potom se vrši upoređivanje. Ako je
39        element niza veliko slovo, on će biti pretvoren u malo; ako nije
40        veliko slovo, element neće biti promenjen. Moglo je se i
41        drugacije resiti, recimo pretvaranjem svih slova u velika, sa
42        funkcijom toupper. */
43     if (tolower(niz[i]) != tolower(niz[n-1-i]))
44     {
45         printf("Niz nije palindrom.\n");
46         /* Ukoliko niz nije palindrom, ispitivanje se može prekinuti i
47            izaci iz programa, dalje ispitivanje nije potrebno. */
48         return 0;
49     }
50
51 /* U slučaju kada je petlja završena, a nije izaslo iz programa,
52    niz jeste palindrom jer uslov nikada nije bio ispunjen. */
53 printf("Niz jeste palindrom!\n");
54
55 return 0;
56 }

```

Rešenje 1.1.22

```

1 #include <stdio.h>
2
3 #define MAX 100
4
5 int main()
6 {
7     int brojevi[MAX];

```

```

9      int n, i, poz_max, poz_min, max, min, tmp;

10
11     /* Ucitava se dimenzija niza i proverava se njena ispravnost. */
12     printf("Unesite dimenziju niza: ");
13     scanf("%d", &n);
14     if(n<1 || n>MAX)
15     {
16         printf("Greska: Nedozvoljena vrednost!\n");
17         return -1;
18     }

19     /* Ucitavaju se elementi niza. */
20     printf("Unesite elemente niza:\n");
21     for(i=0;i<n;i++)
22         scanf("%d", &brojevi[i]);

23
24     /* Maksimalnim tj. minimalnim elementom niza proglašava se nulti
25        element niza. Pozicije maksimalnog tj. minimalnog elementa se
26        postavljaju na 0. */
27     max = brojevi[0];
28     min = brojevi[0];
29     poz_max = 0;
30     poz_min = 0;

31     /* U prolazu kroz niz trazi se maksimalni i minimalni element i
32        pamte se njihove pozicije. */
33     for(i=1;i<n;i++)
34     {
35         if(brojevi[i] > max)
36         {
37             max = brojevi[i];
38             poz_max = i;
39         }

40         if(brojevi[i] < min)
41         {
42             min = brojevi[i];
43             poz_min = i;
44         }
45     }

46
47     /* Zamenjuju se elementi na pozicijama poz_min i poz_max. */
48     tmp = max;
49     brojevi[poz_max] = min;
50     brojevi[poz_min] = tmp;

51     /* Ispisuje se rezultujući niz. */
52     for(i=0;i<n;i++)
53         printf("%d ", brojevi[i]);
54     printf("\n");
55

```

```
57     return 0;
    }
```

Rešenje 1.1.23

```
1  #include <stdio.h>
2
3  #define MAX 500
4
5  /* Ucitavanje dimenzije i elemenata niza. */
6  int ucitavanje(int niz[])
7  {
8      int i, n;
9
10     printf("Unesite dimenziju niza: ");
11     scanf("%d", &n);
12
13     if (n<=0 || n>DIM)
14     {
15         printf("Nedozvoljena dimenzija niza.\n");
16         exit(EXIT_FAILURE);
17     }
18
19     printf("Unesite elemente niza: ");
20     for(i=0; i<n; i++)
21         scanf("%d", &niz[i]);
22
23     return n;
24 }
25
26 /* Funkcija koja vraca 1 ukoliko broj x postoji u nizu, 0 inace. */
27 int postoji(int niz[], int n, int x)
28 {
29     int i;
30
31     for(i=0; i<n; i++)
32         if(niz[i] == x)
33             return 1;
34
35     return 0;
36 }
37
38 int main()
39 {
40     int a[MAX], b[MAX], unija[2*MAX], presek[MAX], razlika[MAX];
41     int i, j, n_a, n_b, n_u, n_p, n_r, indikator;
42
43     /* Unose se dva niza. */
44     n_a = ucitavanje(a);
45     n_b = ucitavanje(b);
46 }
```

```

48  /* Brojaci elemenata u nizovima unija, presek i razlika. */
50  n_u = 0;
50  n_p = 0;
50  n_r = 0;

52  for(i=0;i<n_a;i++)
53  {
54      /* Ukoliko se element a[i] ne nalazi u uniji, dodaje se u uniju
55       i povecava se brojac elemenata u nizu unija. */
56      if(postoji(unija,n_u,a[i]) == 0)
57      {
58          unija[n_u] = a[i];
59          n_u++;
60      }

61      /* Ukoliko se element a[i] nalazi u nizu b i ne postoji u nizu
62       presek, dodaje se presek i povecava se brojac elemenata u nizu
63       presek. */
64      if(postoji(b, n_b, a[i])==1 && postoji(presek, n_p, a[i])==0)
65      {
66          presek[n_p] = a[i];
67          n_p++;
68      }

69      /* Ukoliko element a[i] ne postoji u nizu b i ne postoji u nizu
70       razlika, dodaje se u razliku i povecava se brojac elemenata u
71       nizu razlika. */
72      if(postoji(b, n_b, a[i])==0 && postoji(razlika, n_r, a[i])==0)
73      {
74          razlika[n_r] = a[i];
75          n_r++;
76      }
77  }

78  /* Elemente niza b koji nisu uneti u uniju dodaju se u uniju. */
79  for(i=0;i<n_b;i++)
80      if(postoji(unija, n_u, b[i]))
81      {
82          unija[n_u] = b[i];
83          n_u++;
84      }

85  printf("Unija: ");
86  for(i=0;i<n_u;i++)
87      printf("%d ", unija[i]);

88  printf("\nPresek: ");
89  for(i=0;i<n_p;i++)
90      printf("%d ", presek[i]);

91  printf("\nRazlika: ");
92  for(i=0;i<n_r;i++)

```



```
94     printf("%d ", razlika[i]);
96     return 0;
}
```

Rešenje 1.1.24

```
#include <stdio.h>
2
#define MAX 100
4
int main()
6 {
    int a[MAX];
    int b[MAX];
8
    /* Rezultujući niz ima najviše 2*MAX elemenata. */
    int c[2*MAX];
10
    int n;
    int i,j;
12
    /* Učitava se dimenzija nizova i proverava njena ispravnost. */
    printf("Unesite dimenziju nizova:\n");
    scanf("%d", &n);
    if (n<1 || n>MAX)
14     {
        printf("Nedozvoljena vrednost!\n");
        return -1;
22     }
24
    /* Učitavaju se elementi prvog niza. */
    printf("Unesite elemente niza a:\n");
    for(i=0;i<n;i++)
26     {
        scanf("%d", &a[i]);
    }
28
    /* Učitavaju se elementi drugog niza. */
    printf("Unesite elemente niza b:\n");
    for(i=0;i<n;i++)
34     {
        scanf("%d", &b[i]);
    }
36
38
40     /* Formira se treci niz.
       Koriste se dva indeksa:
42
        - indeks i pomocu kojeg se pristupa elementima nizova a i b i
        koji treba uvecati za 1 nakon svake iteracije
```

```

44         - indeks j pomocu kojeg se pristupa elementima rezultujućeg
           niza c; s obzirom da se u svakoj iteraciji u niz c smestaju dva
           elementa, jedan iz niza a i jedan iz niza b, indeks j se uvecava
           za 2 nakon svake iteracije. */
46     for(i=0,j=0;i<n;i++,j+=2)
47     {
48         c[j]=a[i];
           c[j+1]=b[i];
50     }

52     /* Ispisuju se elementi rezultujućeg niza. */
    printf("Rezultujući niz:\n");
54     for(i=0;i<2*n;i++)
        printf("%d ",c[i]);
56     printf("\n");

58     return 0;
}

```

Rešenje 1.1.25

```

1  #include <stdio.h>

3  #define MAX 100

5  int main()
6  {
7      int a[MAX], b[MAX], c[2*MAX];
8      int i, n;

9      /* Ucitava se broj elemenata nizova i proverava se njegova
           ispravnost. */
11     printf("Unesite broj n: ");
        scanf("%d", &n);
13     if(n<1 || n>MAX)
        {
15         printf("Greska: Nedoovoljena vrednost!\n");
            return -1;
17     }

19     /* Ucitavaju se elementi nizova */
    printf("Unesite elemente niza a: ");
21     for(i=0;i<n;i++)
        scanf("%d", &a[i]);

23     printf("Unesite elemente niza b: ");
25     for(i=0;i<n;i++)
        scanf("%d", &b[i]);

27

```

```

/* Niz c ima 2*n elemenata: prvih n elemenata su elementi niza b, a
   narednih n elemenata elementi niza a. Elementi niza b se nalaze
   na pozicijama 0,1,2,...n-1, a elementi niza a na pozicijama n,n
   +1,...2*n-1. Jednim prolaskom kroz petlju na poziciju i u nizu c
   se postavlja element b[i] niza b, a na poziciju n+i element a[i]
   niza a. */
29 for(i=0;i<n;i++)
   {
31     c[i] = b[i];
     c[n+i] = a[i];
33 }

35 /* Ispisuju se elementi niza c. */
   for(i=0;i<2*n;i++)
37     printf("%d ", c[i]);
   printf("\n");
39
   return 0;
41 }

```

Rešenje 1.1.26

```

1  #include <stdio.h>

3  #define DIMENZIJA 100

5  /* Funkcija za učitavanje niza. */
   void unos(int niz[], int n)
7  {
     int i;

9     printf("Uneti elemente sortiranog niza:\n");
     for(i=0; i<n; i++)
11         scanf("%d", &niz[i]);
13 }

15 /* Funkcija za ispis niza. */
   void ispis(int niz[], int n)
17 {
     int i;

19     for(i=0; i<n; i++)
21         printf("%d ", niz[i]);

23     printf("\n");
   }

25 int main()
27 {
     int a[DIMENZIJA], b[DIMENZIJA];

```

```

29  /* Niz c moze biti duplo veci od nizova a i b, pa za njega treba
    odvojiti dupli prostor. */
    int c[2*DIMENZIJA];
31  int m, n;
    /* Brojac u petlji za elemente niza a. */
33  int i = 0;
    /* Brojac u petlji za elemente niza b. */
35  int j = 0;
    /* Brojac u petlji za elemente niza c. */
37  int k = 0;

39  printf("Uneti broj elemenata niza: ");
    scanf("%d", &n);

41  unos(a, n);
43  unos(b, n);

45  while(i<n && j<n)
    {
47      /* Porede se elementi nizova a i b i u niz c upisuje se samo onaj
        koji je manji. */
        if (a[i] < b[j])
49      {
            c[k] = a[i];
51      /* Kako element niza a je upisan u niz c, uvecava se brojac
        niza a. Element niza b nije upisan u niz c, te brojac za niz b ne
        treba uvecavati. */
            i++;
53      }
        else
55      {
            c[k] = b[j];
57      j++;
        }

59      /* U nizu c na poziciju k je upisan ili a[i] ili b[j]. Brojac k
        se uvecava. */
61      k++;
    }

63  /* Ukoliko je ostalo elemenata u nizu a, upisuju se u niz c. */
65  while(i < n)
    {
67      c[k] = a[i];
        k++;
69      i++;
    }

71  /* Ukoliko je ostalo elemenata u nizu b, upisuju se u niz c. */
73  while(j < n)
    {
75      c[k] = b[j];

```

```

77     k++;
78     j++;
79 }
80
81 /* Ispis elemenata niza c cija dimenzija je zbir dimenzija nizova a
82    i b. */
83 ispis(c, 2*n);
84
85 return 0;
86 }

```

Rešenje 1.1.27

```

1  #include <stdio.h>
2
3  #define DIMENZIJA 10
4
5  /* Funkcija za učitavanje niza. */
6  void unos(int niz[])
7  {
8      int i;
9
10     printf("Unesite %d brojeva:\n", DIMENZIJA);
11     for(i=0; i<DIMENZIJA; i++)
12         scanf("%d", &niz[i]);
13 }
14
15 /* Funkcija za ispis niza. */
16 void ispis(int niz[], int n)
17 {
18     int i;
19
20     for(i=0; i<DIMENZIJA; i++)
21         printf("%d ", niz[i]);
22
23     printf("\n");
24 }
25
26 int main()
27 {
28     int niz[DIMENZIJA];
29     /* Brojaci i i j. Brojac i kreće od početka niza, a brojac j od
30        kraja. */
31     int i=0, j = DIMENZIJA-1;
32     /* Pomocna promenljiva za razmenu elemenata niza. */
33     int pom;
34
35     unos(niz);
36
37     /* Ideja u resenju je da se krene sa dva kraja niza -- sa početka
38        niza i sa kraja i svaki put kada se naidje na elemente koji po

```

```

    parnosti ne odgovaraju delu niza u kome treba da budu, ti
    elementi se zamene. */
37 while(i < j && i < DIMENZIJA && j >= 0)
    {
39     /* Ukoliko elementi na pozicijama i i j su razlicite parnosti,
        vrsi se razmena tih elemenata niza. */
        if (niz[i] % 2 != 0 && niz[j] % 2 == 0)
41     {
            pom = niz[i];
43         niz[i] = niz[j];
            niz[j] = pom;
45     }

47     /* Ukoliko je element na poziciji i paran, prelazi se na sledeci
        element niza, brojac i se uvecava. */
        if (niz[i] % 2 == 0)
49         i++;

51     /* Ukoliko je element na poziciji j neparan, prelazi se na
        sledeci element niza, brojac j se smanjuje. */
        if (niz[j] % 2 != 0)
53         j--;
    }

55     printf("Rezultujuci niz:\n");
57     ispis(niz, DIMENZIJA);

59     return 0;
}

```

Rešenje 1.1.28

```

#include<stdio.h>
2
#define MAX 100
4
/* Funkcija kojom se ucitavaju elementi niza a dimenzije n. */
6 void ucitaj(int a[], int n)
    {
8         int i;
            for(i=0;i<n;i++)
10         {
                scanf("%d",&a[i]);
12         }
    }

14
/* Funkcija kojom se ispisuju elementi niza a dimenzije n. */
16 void stampaaj(int a[], int n)
    {
18         int i;
            for(i=0;i<n;i++)

```

```

20         printf("%d ",a[i]);
21     printf("\n");
22 }
23
24
25
26 /* Funkcija koja obrce elemente niza. */
27 void obrni(int a[], int n)
28 {
29
30     int t;
31     int i,j;
32
33
34     /* Za niz a[0], a[1], ..., a[n-2], a[n-1] obrnuti niz je a[n-1],
35        a[n-2], ..., a[1], a[0]. Zato je potrebno razmeniti vrednosti
36        elemenata a[0] i a[n-1], a[1] i a[n-2], itd. i zaustaviti se kada
37        je vrednost indeksa prvog elementa veca od vrednosti drugog
38        elementa. */
39
40     for(i=0,j=n-1;i<j;i++, j--)
41     {
42         t = a[i];
43         a[i] = a[j];
44         a[j] = t;
45     }
46 }
47
48 /* Funkcija koja rotira niz ciklicno za jedno mesto u levo. */
49 void rotiraj1(int a[], int n)
50 {
51     int i;
52     int tmp;
53
54     /* Izdvaja se prvi element niza. */
55     tmp=a[0];
56
57     /* Pomeraju se preostali elementi niza. */
58     for(i=0;i<n-1;i++){
59         a[i]=a[i+1];
60     }
61
62     /* Poslednjem elementu se dodeljuje sacuvana vrednost prvog
63        elementa. */
64     a[n-1] = tmp;
65 }
66
67 /* Funkcija koja rotira niz ciklicno za k mesta u levo. */
68 void rotirajk(int a[], int n, int k)
69 {
70     int i;

```

```

68      /* Odredjuje se vrednost broja k koja je u opsegu od 0 do n-1 kako
        bi se izbegla suvisna pomeranja. */
        k=k%n;

70
71      /* Niz se rotira za jednu poziciju ulevo k puta. */
72      for(i=0;i<k;i++)
            rotiraj1(a,n);
74  }

76  int main()
77  {
78      int a[MAX];
79      int n;
80      int i;
81      int k;
82      int m;

83
84      /* Ucitava se dimenzija niza i proverava se njena ispravnost. */
85      printf("Unesite dimenziju niza:");
86      scanf("%d",&n);
87      if (n<1 || n>MAX)
88      {
89          printf("Nedozvoljena vrednost!\n");
90          return -1;
91      }
92
93      /* Ucitavaju se elementi niza. */
94      ucitaj(a,n);

95
96      /* Testira se rad napisanih funkcija */

97
98      /* Obrtanje niza. */
99      printf("Elementi niza nakon obrtanja:\n");
100     obrni(a,n);
101     stampaj(a,n);

102
103     /* Rotiranje za jedno mesto u levo. */
104     printf("Elementi niza nakon rotiranja za 1 mesto ulevo:\n");
105     rotiraj1(a,n);
106     stampaj(a,n);

107
108     /* Rotiranje za k mesta u levo. */
109     printf("Unesite jedan pozitivan ceo broj:");
110     scanf("%d",&k);
111     if (k<=0)
112     {
113         printf("Nekorektan unos\n");
114         return -1;
115     }
116     rotirajk(a,n,k);

```



```

118     printf("Elementi niza nakon rotiranja za %d mesto ulevo:\n",k);
120     stampaj(a,n);

122     return 0;
}

```

Rešenje 1.1.29

```

1  #include <stdio.h>

3  #define DIM 2000

5  /* Funkcija za ispis elemenata niza. */
void ispis(int niz[], int n)
7  {
    int i;

9      for(i=0; i<n; i++)
11         printf("%d ", niz[i]);
    printf("\n");
13 }

15 /* Funkcija za ubacivanje na kraj niza. Vraca novu dimenziju niza. */
int ubaci_na_kraj(int niz[], int n, int x)
17 {
    niz[n] = x;
19     return n+1;
}

21 /* Funkcija za ubacivanje na pocetak niza. Vraca novu dimenziju niza.
    */
23 int ubaci_na_pocetak(int niz[], int n, int x)
{
25     int i;

27     /* Prvo se svi elementi niza pomere za jednu poziciju u desno da bi
        se oslobodio prostor za prvi element niza. Poslednji element
        niza se pomera sa pozicije (n-1) na poziciju (n). Slicno se
        pomeraju i ostali elementi. */
    for(i=n; i>0; i--)
29         niz[i] = niz[i-1];

31     /* Na prvu poziciju se upisuje novi element. Bitan je redosled
        naredbi: ako bi prvo bio upisan novi element, a tek onda izvršeno
        pomeranje, element na poziciji niz[0] bi bio obrisan i ne bi
        mogao biti upisan na poziciju niz[1]. */
    niz[0] = x;

33     return n+1;

35 }

```

```

37  /* Funkcija za ubacivanje elementa na neku poziciju u nizu. Vraca
    novu dimenziju niza. */
39  int ubaci_na_poziciju(int niz[], int n, int x, int pozicija)
    {
41      int i;

43      /* Prvo se svi elementi niza od pozicije do kraja pomere za jedno
        mesto u desno da bi se oslobodio prostor za novi element niza.
        */
        for(i=n; i>pozicija; i--)
45          niz[i] = niz[i-1];

47      /* Na poziciju se upisuje novi element. */
        niz[pozicija] = x;

49      return n+1;
51  }

53  /* Funkcija za brisanje prvog elementa niza. Vraca novu dimenziju
    niza. */
    int brisi_prvog(int niz[], int n)
55  {
        int i;

57      /* Svi elementi niza pomeraju se za jedno mesto u levo. */
        for(i=0; i<n-1; i++)
59          niz[i] = niz[i+1];

61      return n-1;
63  }

65  /* Funkcija za brisanje poslednjeg elementa niza. Vraca novu
    dimenziju niza. */
    int brisi_poslednjeg(int niz[], int n)
67  {
        /* Dovoljno je smanjiti dimenziju niza, elemente niza nije potrebno
            brisati. */
69      return n-1;
    }

71  /* Funkcija za brisanje elementa niza. Pretpostavlja se da element
    ima samo jedno pojavljivanje (za vezbu napisati funkciju koja
    brise sva pojavljivanja, ako ih ima vise). Vraca novu dimenziju
    niza. */
73  int brisi_element(int niz[], int n, int x)
    {
75      int i, j;

77      /* Prvo treba pronaci poziciju elementa u nizu. */
        for(i=0; i<n; i++)

```

```

79     if (niz[i] == x)
80         break;
81
82     /* Provera da li element postoji u nizu. Ako je brojac stigao do
83        kraja niza, onda element ne postoji u nizu. */
84     if (i == n)
85     {
86         printf("Klijent sa rednim brojem %d ne postoji u nizu.\n", x);
87         return n;
88     }
89
90     /* Ukoliko element postoji u nizu, svi elementi niza nakon njega se
91        pomeraju za jedno mesto u levo. */
92     for(j = i; j<n-1; j++)
93         niz[j] = niz[j+1];
94
95     return n-1;
96 }
97
98 int main()
99 {
100     int n, niz[DIM], i, klijent, pozicija;
101
102     /* Ucitavanje dimenzije i elemenata niza. */
103     printf("Unesite trenutni broj klijenata: ");
104     scanf("%d", &n);
105
106     if (n<=0 || n>DIM)
107     {
108         printf("Nedozvoljena dimenzija niza.\n");
109         return 0;
110     }
111
112     printf("Unesite niz sa rednim brojevima klijenata: ");
113     for(i=0; i<n; i++)
114         scanf("%d", &niz[i]);
115
116     /* Ubacivanje klijenta na kraj. */
117     printf("Unesite klijenta kojeg treba ubaciti u niz: ");
118     scanf("%d", &klijent);
119     n = ubaci_na_kraj(niz, n, klijent);
120     printf("Niz nakon ubacivanja klijenta:\n");
121     ispis(niz, n);
122
123     /* Ubacivanje klijenta na pocetak. */
124     printf("Unesite prioritetnog klijenta kojeg treba ubaciti u niz: ");
125     scanf("%d", &klijent);
126     n = ubaci_na_pocetak(niz, n, klijent);
127     printf("Niz nakon ubacivanja klijenta:\n");
128     ispis(niz, n);

```

```

129  /* Ubacivanje klijenta na zadatu poziciju. */
    printf("Unesite prioritetnog klijenta kojeg treba ubaciti u niz i
        njegovu poziciju:");
131  scanf("%d%d", &klijent, &pozicija);
    if (pozicija < 0 || pozicija > n)
133  {
        printf("Neispravna pozicija.\n");
135  }
    else
137  {
        n = ubaci_na_poziciju(niz, n, klijent, pozicija);
139        printf("Niz nakon ubacivanja klijenta:\n");
        ispis(niz, n);
141    }

143  /* Brisanje prvog klijenta. */
    n = brisi_prvog(niz, n);
145    printf("Niz nakon odlaska klijenta:\n");
    ispis(niz, n);
147
    /* Brisanje poslednjeg klijenta. */
149    n = brisi_poslednjeg(niz, n);
    printf("Niz nakon odlaska klijenta:\n");
151    ispis(niz, n);

153  /* Brisanje klijenta sa datim rednim brojem. */
    printf("Unesite redni broj klijenta koji je napustio red: ");
155    scanf("%d", &klijent);
    n = brisi_element(niz, n, klijent);
157    printf("Niz nakon odlaska klijenta:\n");
    ispis(niz, n);
159
    return 0;
161 }

```

Rešenje 1.1.30

```

#include <stdio.h>

2
#define MAX 100

4
int main()
{
6    int a[MAX];
    int i, j, n_a;

8

10    /* Ucitava se broj elemenata niza i proverava se njegova ispravnost
        . */
    printf("Unesite broj elemenata niza: ");
12    scanf("%d", &n_a);

```

```

14     if(n_a<1 || n_a>100)
15     {
16         printf("Greska: Nedozvoljena vrednost!\n");
17         return -1;
18     }
19
20     /* Ucitavaju se elementi niza. */
21     printf("Unesite elemente niza: ");
22     for(i=0;i<n_a;i++)
23         scanf("%d", &a[i]);
24
25     /* Parametar j predstavlja brojac prve slobodne pozicije na koju se
26        moze upisati element niza koji treba da ostane u nizu. Kada se
27        naidje na element koji je paran, on se kopira na mesto a[j] i
28        poveca se vrednost brojaca j. Ukoliko se naidje na element koji
29        je neparan, njega treba preskociti. */
30     for(i=0, j=0;i<n_a;i++)
31     {
32         /* Ako je tekuci element niza a paran. */
33         if(a[i]%2 == 0)
34         {
35             /* Premesta se na poziciju j. */
36             a[j] = a[i];
37
38             /* Vrednost brojaca j se priprema za narednu iteraciju. */
39             j++;
40         }
41
42         /* Ako je tekuci element niza a neparan, sa njim nista ne treba
43            raditi. */
44     }
45
46     /* U nizu a se sada na pozicijama od 0,...,j-1 nalaze elementi koji
47        su parni, te je njegova nova dimenzija j. */
48     n_a=j;
49
50     /* Ispisuju se elementi modifikovanog niza a. */
51     for(i=0;i<n_a;i++){
52         printf("%d ", a[i]);
53     }
54     printf("\n");
55     return 0;
56 }

```

Rešenje 1.1.31

```

1 #include <stdio.h>
2
3 #define MAX 100
4

```

```

int main()
{
    int a[MAX];
    int n;
    int i,j;
    char poslednja_cifra;
    int novo_n;

    /* Ucitava se dimenzija niza i proverava njena ispravnost. */
    printf("Unesite dimenziju niza:\n");
    scanf("%d", &n);
    if (n<1 || n>MAX)
    {
        printf("Nedozvoljena vrednost!\n");
        return -1;
    }

    /* Ucitavaju se elementi niza a. */
    printf("Unesite elemente niza a:\n");
    for(i=0;i<n;i++)
        scanf("%d", &a[i]);

    /* Obilaze se svi elementi niza a. */
    for(i=0, j=0; i<n; i++)
    {
        /* Izdvaja se poslednja cifra tekućeg elementa. */
        poslednja_cifra = a[i]%10;

        /* Ako je poslednja cifra 0 ili je element deljiv svojom
        poslednjom cifrom, taj element se zadržava i smesta na poziciju j. */
        if (poslednja_cifra==0 || a[i]%poslednja_cifra==0)
        {
            a[j]=a[i];
            j++;
        }
    }

    /* Dimenzija novog niza odgovara posledjoj vrednosti broja j. */
    novo_n=j;

    /* Ispisuje se rezultujući niz. */
    printf("Niz a nakon izmena:\n");
    for(i=0; i<novo_n;i++)
        printf("%d ", a[i]);
    printf("\n");

    return 0;
}

```

Rešenje 1.1.32

```

2  #include <stdio.h>
4
6  #define DIM 700
8
10 /* Funkcija pomera za jedno mesto u levo elemente niza a pocevsi od
    pozicije j. Element na poziciji j se brise i na njegovo mesto se
    upisuje element na poziciji j+1, a u skladu sa tim svi ostali
    elementi posle njega u nizu se pomeraju. */
12 void pomeri_za_jedno_mesto(int a[], int n, int j)
14 {
16     int i;
18     for(i=j; i<n; i++)
19         a[i] = a[i+1];
20 }
22
24 int main()
26 {
28     int n, niz[DIM], i;
30
32     /* Ucitavanje dimenzije i elemenata niza. */
34     printf("Unesite broj elemenata niza: ");
36     scanf("%d", &n);
38
40     if (n<=0 || n>DIM)
42     {
44         printf("Nedozvoljena dimenzija niza.\n");
46         return 0;
48     }
50
52     printf("Unesite elemente niza: ");
54     for(i=0; i<n; i++)
56         scanf("%d", &niz[i]);
58
60     /* Potrebno je krenuti od poslednjeg elementa niza i petljom ici ka
        pocetku niza (element na poziciji 0 se ne razmatra). Proverava
        se da li je element potrebno obrisati i ako jeste vrsi se
        pomeranje elemenata niza za jedno mesto u levo. Prednost ovog
        resenja u odnosu na resenje kada se krene od pocetka niza je u
        tome sto element koji se ispituje sigurno nije promenio svoju
        poziciju usled pomeranja zbog brisanja. Problem se moze resiti i
        koriscenjem pomocnog niza (uraditi za vezbu). To resenje je
        efikasnije, ali trosi vise resursa. */
62     for(i=n-1; i>0; i--)
64     {
66         if (niz[i]%i != 0)
68         {
70             pomeri_za_jedno_mesto(niz, n, i);
72             /* Nakon brisanja elementa, smanjuje se i dimenzija niza. */
74             n--;
76         }
78     }
80 }

```

```

42      /* Stapanje novog niza. */
44      printf("Novi niz:\n");
46      for(i=0; i<n; i++)
47          printf("%d ", niz[i]);
48      printf("\n");
49      return 0;
50  }

```

Rešenje 1.1.33

```

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

4  #define MAX 100

6  /* Funkcija koja proverava da li je zadati broj prost broj. Povratna
   vrednost funkcije je 1 ukoliko broj jeste prost, inace je 0. */
7  int prost(int x)
8  {
9      int i;
10
11      /* Posmatra se apsolutna vrednost broja kako bi se pokrio i slucaj
12      negativnih brojeva. */
13      x=abs(x);
14
15      /* Brojevi 2 i 3 su prosti. */
16      if(x == 2 || x == 3)
17          return 1;
18
19      /* Ako je broj paran nije prost. */
20      if(x%2 == 0)
21          return 0;
22
23      /* Ako broj ima delioce u skupu [3, koren_broja(x)] takodje nije
24      prost. */
25      for(i=3; i<=sqrt(x); i+=2){
26          if(x%i == 0)
27              return 0;
28      }
29
30      /* Ako su svi uslovi ispunjeni, broj je prost. */
31      return 1;
32  }
33
34  int main()
35  {
36      int a[MAX];
37      int i, j, n_a, n_b;

```



```

36  /* Rezultujući niz. */
    int b[MAX];

38

40  /* Učitava se broj elemenata niza i proverava se njegova ispravnost
    . */
    printf("Unesite broj elemenata niza: ");
42  scanf("%d", &n_a);
    if(n_a<1 || n_a>MAX)
44  {
        printf("Greska: Nedoovoljena vrednost!\n");
46        return -1;
    }

48

    /* Učitavaju se elementi niza a. */
50  printf("Unesite elemente niza: ");
    for(i=0;i<n_a;i++)
52        scanf("%d", &a[i]);

54  /* Kada se u nizu a nađje na prost element, on se upisuje u niz b
    i uvecava se brojac za niz b. */
    for(i=0, j=0;i<n_a;i++){
56        if(prost(a[i]) == 0)
            {
58                b[j] = a[i];
                    j++;
60            }
    }

62

    /* Broj elemenata novodobijenog niza b je j. */
64  n_b = j;

66  /* Ispisuju se elementi niza b. */
    for(i=0;i<n_b;i++)
68        printf("%d ", b[i]);
    printf("\n");

70
    return 0;
72 }

```

Rešenje 1.1.34

```

#include <stdio.h>

2

#define DIM 300

4

int main()
6 {
    int n, niz[DIM], i;

8

    /* Učitavanje dimenzije i elemenata niza. */

```

```

10 printf("Unesite dimenziju niza: ");
   scanf("%d", &n);

12
   if (n<=0 || n>DIM)
14   {
       printf("Nedozvoljena dimenzija niza.\n");
16       return 0;
   }

18
   printf("Unesite elemente niza: ");
20   for(i=0; i<n; i++)
       scanf("%d", &niz[i]);
22

24   /* Provera da li su elementi niza uredjeni neopadajuće. */
   for(i=0; i<n-1; i++)
26   /* Porede se svaka dva uzastopna elementa niza. Ukoliko za bilo
       koja takva dva elementa uslov nije ispunjen, prekida se dalje
       ispitivanje i vraća se poruka da niz nije uredjen neopadajuće. U
       suprotnom se nakon izlaska iz petlje ispisuje poruka da je niz
       uredjen neopadajuće. */
       if (niz[i] > niz[i+1])
28   {
       printf("Nije uredjen neopadajuće.\n");
30       return 0;
   }

32   printf("Jeste uredjen neopadajuće!\n");

34   return 0;
36 }

```

Rešenje 1.1.35

```

1 #include <stdio.h>
   #include <stdlib.h>
3
   /* Maksimalan broj dana u mesecu je 31, ali dani pocinju od 1, pa je
       potrebno odvojiti 32 mesta u nizu jer se nulti ne koristi. */
5 #define DIM 32

7 /* Ucitavanje dimenzije i elemenata niza. */
   int ucitavanje(int niz[])
9 {
       int i, n;

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

15       if (n<=0 || n>DIM)
       {

```

```

17     printf("Nedozvoljena dimenzija niza.\n");
18     exit(EXIT_FAILURE);
19 }

21 printf("Unesite broj prodatih artikala: ");
22 for(i=0; i<n; i++)
23     scanf("%d", &niz[i]);

25     return n;
26 }

27 int najduzi_neopadajuci(int a[], int n)
28 {
29     int i;
30     /* Parametar u kome se cuva duzina serije. Na pocetku je duzina
31     serije jednaka 1, odnosno, samo jedan element je u seriji.
32     Kasnije, u petlji se ovaj broj moze uvecati. */
33     int ts = 1;
34     /* Parametar ns predstavlja duzinu najduze serije. Na pocetku se
35     postavlja na 1, odnosno pretpostavlja se da je duzina najduze
36     serije 1, a u petlji se ova vrednost moze izmeniti. */
37     int ns = 1;

38     for (i = 1; i < n; i++)
39     {
40         /* Proverava se da li uzastopni elementi ispunjavaju neopadajuci
41         uslov. Ako je to slucaj uvecava se duzina serije. */
42         if (a[i] >= a[i-1])
43             ts++;
44         else
45             /* Ako uzastopni elementi nisu jednaki serija je prekinuta i
46             parametar za duzinu serije se postavlja ponovo na 1 da bi mogla da
47             se racuna duzina sledece serije. */
48             ts = 1;

49         /* Ukoliko je trenutna duzina serije veca od duzine do sada
50         najduze serije, parametar za duzinu najduze serije se postavlja
51         na novu, vecu vrednost. */
52         if (ts > ns)
53             ns = ts;
54     }

55     return ns;
56 }

57 int main()
58 {
59     int n, a[DIM];
60     int i;

61     n = ucitavanje(a);

```

```

        printf("Duzina najduzeg neopadajućeg prodavanja je %d.\n",
               najduzi_neopadajuci(a, n));
61
        return 0;
63 }

```

Rešenje 1.1.36

```

1  #include <stdio.h>
   #include <stdlib.h>
3
   #define DIM 100
5
   /* Ucitavanje dimenzije i elemenata niza. */
7  int ucitavanje(int niz[])
   {
9      int i, n;

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

13
       if (n<=0 || n>DIM)
15     {
           printf("Nedozvoljena dimenzija niza.\n");
17         exit(EXIT_FAILURE);
       }

19     printf("Unesite elemente niza: ");
21     for(i=0; i<n; i++)
           scanf("%d", &niz[i]);

23     return n;
25 }

27 int najduza_serija(int a[], int n)
   {
29     int i;
       /* Parametar u kome se cuva duzina serije. Na pocetku je duzina
          serije jednaka 1, odnosno, samo jedan element je u seriji.
          Kasnije, u petlji se ovaj broj moze uvecati. */
31     int ts = 1;
       /* Parametar ns predstavlja duzinu najduze serije. Na pocetku se
          postavlja na 1, odnosno pretpostavlja se da je duzina najduze
          serije 1, a u petlji se ova vrednost moze izmeniti. */
33     int ns = 1;

35     for (i = 1; i < n; i++)
       {
37         /* Proverava se da li su uzastopni elementi jednaki. Ako je to
            slucaj uvecava se duzina serije. */
           if (a[i] == a[i-1])

```

```

39     ts++;
    else
41     /* Ako uzastopni elementi nisu jednaki serija je prekinuta i
        paramtar za duzinu serije se postavlja ponovo na 1 da bi mogla da
        se racuna duzina sledece serije. */
        ts = 1;
43
    /* Ukoliko je trenutna duzina serije veca od duzine do sada
        najduze serije, parametar za duzinu najduze serije se postavlja
        na novu, vecu vrednost. */
45     if (ts > ns)
        ns = ts;
47 }

49 return ns;
}

51 int main()
52 {
53     int n, a[DIM];
54     int i;

55     n = učitavanje(a);

56     printf("Duzina najduze serije je %d.\n", najduza_serija(a, n));

57     return 0;
58 }

```

Rešenje 1.1.37

```

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

#define DIM 100

/* Učitavanje dimenzije i elemenata niza. */
int učitavanje(int niz[])
{
    int i, n;

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

    if (n <= 0 || n > DIM)
    {
        printf("Nedozvoljena dimenzija niza.\n");
        exit(EXIT_FAILURE);
    }

    printf("Unesite elemente niza: ");

```

```

22     for(i=0; i<n; i++)
        scanf("%d", &niz[i]);

24     return n;
}

26
/* Resenje pod a. */
28 int podniz_uzastopnih(int a[], int n, int b[], int m)
{
30     int i, j;

32     /* Prolaze se elementi prvog niza. Svaki element prvog niza moze
        biti pocetak podniza, odnosno pocetak drugog niza. */
    for (i = 0; i + m - 1 < n; i++)
    {
34         /* Prolaze se elementi drugog niza. Za svaki element niza b
            proverava se da li je jednak odgovarajucem elementu niza a.
            Za niz a razmatra se da li podniz pocinje od pozicije i.
            Tako 0-ti element niza b je na poziciji i, 1-vi element je na
            poziciji i+1, 2-gi na poziciji i+2, ..., j-ti na poziciji i+j.
            Ako uslov nije ispunjen, petlja se prekida i proverava se da li
            na sledecoj poziciji u nizu a pocinje podniz. */
            for (j = 0; j < m; j++)
38                 if (a[i + j] != b[j])
                    break;
40         /* Ako petlja nije prekinuta nakon ispitivanja, brojac za niz b
            je jedanak dimenziji niza b, odnosno svi elementi niza b se
            uzastopno nalaze u nizu a. */
            if (j == m)
42                 return 1;
    }

44     /* Ukoliko niz b jeste uzastopni podniz uslov u petlji ce u nekom
        trenutku biti ispunjen i iz petlje i funkcije ce se izaci sa
        return naredbom. Ipak, ako se to nije desilo i dalje se izvršava
        funkcija, onda niz b nije uzastopni podniz. */
46     return 0;
}

48
/* Resenje pod b. */
50 int podniz(int a[], int n, int b[], int m)
{
52     int i, j;

54     /* Petljom se prolaze elementi niza a. */
    for (i = 0, j = 0; i < n && j < m; i++)
56         /* Svaki put kada se naidje na element niza b, brojac za niz b se
            uvecava i proverava se da li se sledeci element niza b nalazi u
            nizu a. */
            if (a[i] == b[j])
58                 j++;

```

```

60  /* Ukoliko se pronadju svi elementi niza b u nizu a, onda je brojac
    za niz b jednak dimenziji niza b. U tom slucaju se vraca
    vrednost 1, odnosno da niz jeste podniz. */
    return j == m;
62 }

64 int main()
{
66     int n, a[DIM];
    int m, b[DIM];
68     int i;

70     n = ucitavanje(a);
    m = ucitavanje(b);

72     if (podniz_uzastopnih(a, n, b, m))
64         printf("Elementi drugog niza cine uzastopni podniz prvog niza.\n"
74             );
    else
76         printf("Elementi drugog niza ne cine uzastopni podniz prvog niza
        .\n");

78

    if (podniz(a, n, b, m))
80         printf("Elementi drugog niza cine podniz prvog niza.\n");
    else
82         printf("Elementi drugog niza ne cine podniz prvog niza.\n");

84     return 0;
}

```

Rešenje 1.1.38

```

1  #include <stdio.h>

3  #define DIM 100

5  void brojanje(int a[], int b[], int n)
{
7     int i;

9     /* Niz b se inicijalizuje nulama jer se za svaki element postavi da
    se pojavljuje 0 puta u nizu a. */
    for(i=1; i<=n; i++)
11         b[i] = 0;

13     /* Peljom se prolazi kroz niz a i za svaki element a[i] uvecava se
    broj njegovog pojavljivanja u nizu b.
    Na primer, ako je a[3] = 7, onda treba uvecati broj
    pojavljivanja broja 7, a to je b[7]++, sto se
15     krace moze zapisati kao b[a[3]]++.

```

```

        Pretpostavlja se da je niz a dobro zadat, odnosno da su sve
        njegove vrednosti u intervalu od 1 do n.
17     */
19     for(i=0; i<n; i++)
        b[a[i]]++;
21 }
23 int main()
24 {
25     int a[DIM];
    /* Niz b moze imati index DIM (jer niz b se posmatra od 1 do DIM),
        pa zato njegova dimenzija mora biti za jedan veca. */
27     int b[DIM+1];
    int i, n;

29     /* Unos dimenzije i elemenata niza. */
    printf("Unesite broj elemenata niza: ");
31     scanf("%d", &n);

33     if (n <=0 || n > DIM)
    {
35         printf("Neispravna dimenzija niza.\n");
        return -1;
37     }

39     printf("Unesite elemente niza:\n");
    for(i=0; i<n; i++)
41     {
        scanf("%d", &a[i]);
43         /* Niz a moze sadrzati elemente koji nisu u opsegu od 1 do n. U
            tom slucaju taj niz nije permutacija. */
        if (a[i] <=0 || a[i] > n)
45        {
            printf("Uneti niz nije permutacija.\n");
47            return 0;
        }
49    }

51    brojanje(a, b, n);

53    /* Ukoliko se svaki element niza a javlja tacno jednom u nizu a,
        onda niz a jeste permutacija. Ovo svojstvo se proverava
        koriscenjem dobijenog niza b. */
    for(i=1; i<=n; i++)
55        if (b[i] != 1)
        {
57            printf("Uneti niz nije permutacija.\n");
            return 0;
59        }

61    /* Ukoliko prethodna petlja nije ranije zavrшена (i nije se izaslo
        iz programa), onda svaki element niza b je jednak 1, odnosno

```



```

        svaki element se pojavljuje tacno jednom, pa niz a jeste
        permutacija. */
printf("Uneti niz je permutacija.\n");
63
    return 0;
65 }

```

Rešenje 1.1.39

```

1  #include <stdio.h>
3  #define BROJ_CIFARA 10
5  /* Analiziraju se cifre broja i smestaju u odgovarajuci niz. */
void analiza_cifara(int broj, int niz[])
7  {
    int c;
9
    /* Niz predstavlja brojace za cifre broja.
    Na pocetku se ovi nizovi inicijalizuju nulama. */
    for(i=0; i<BROJ_CIFARA; i++)
13         niz[i] = 0;

15     do
    {
17         c = broj%10;
        niz[c]++;
19         broj /= 10;
    }
21     while(broj);
    }
23
int main()
25 {
    char c;
27     /* Niz cifrex predstavlja brojace za cifre broja x.
        Niz cifrey predstavlja brojace za cifre broja y. */
    int cifrex[BROJ_CIFARA], cifrey[BROJ_CIFARA];
    int x, y, i, indikator;
31
    /* Ucitavaju se brojevi x i y. */
33     printf("Unesite dva broja: ");
    scanf("%d%d", &x, &y);
35

    /* Za slucaj da su unete vrednosti negativne, posmatra se njihova
    apsolutna vrednost. Ovo je opravdano iz razloga sto se brojevi x
    i -x zapisuju istim ciframa. */
37     x=abs(x);
    y=abs(y);
39

    analiza_cifara(x, cifrex);

```

```

41 analiza_cifara(y, cifrey);

43

44 /* Promenljiva indikator služi za praćenje da li su oba broja
45    sastavljena od istih cifara. */
46 indikator = 1;

47 for(i=0;i<BROJ_CIFARA;i++){
48     /* Ako se broj pojavljivanja cifre i u zapisu broja x razlikuje
49        od broja pojavljivanja cifre i u zapisu broja y, brojevi se ne
50        zapisuju istim ciframa. Zato se vrednost indikatora može
51        postaviti na 0 i prekinuti dalje upoređivanje broja
52        pojavljivanja. */
53     if(cifrey[i] != cifrex[i])
54     {
55         indikator = 0;
56         break;
57     }
58 }

59 /* Ako je vrednost promenljive indikator ostala 1, to znači da u
60    petlji nije pronađena cifra koja se ne pojavljuje isti broj puta
61    u zapisima brojeva x i y. Zato se može zaključiti da se brojevi
62    zapisuju istim ciframa. */
63 if(indikator)
64     printf("Brojevi se zapisuju istim ciframa!\n");
65 else
66     printf("Brojevi se ne zapisuju istim ciframa!\n");

67 return 0;
68 }

```