

## **PROGRAMIRANJE 1**

Elektronsko izdanje (2019)



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

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

**Beograd  
2019.**

Autori:

*dr Milena Vujošević Janičić*, docent na Matematičkom fakultetu u Beogradu

*dr Jovana Kovačević*, docent na Matematičkom fakultetu u Beogradu

*dr Danijela Simić*, asistent na Matematičkom fakultetu u Beogradu

*Anđelka Zečević*, asistent na Matematičkom fakultetu u Beogradu

*Aleksandra Kocić*, asistent na Matematičkom fakultetu u Beogradu

#### PROGRAMIRANJE 1

Zbirka zadataka sa rešenjima

Izdavač: Matematički fakultet Univerziteta u Beogradu. Studentski trg 16, Beograd.

Za izdavača: *prof. dr Zoran Rakić*, dekan

Recenzenti:

*dr Gordana Pavlović-Lažetić*, redovni profesor na Matematičkom fakultetu u Beogradu

*dr Dragan Urošević*, naučni savetnik na Matematičkom institutu SANU

Obrada teksta, crteži i korice: *autori*.

# Sadržaj

<b>1</b>	<b>Osnovni elementi imperativnog programiranja</b>	<b>1</b>
1.1	Izrazi . . . . .	1
1.2	Grananja . . . . .	10
1.3	Petlje . . . . .	19
1.4	Funkcije . . . . .	44
<b>2</b>	<b>Napredni tipovi podataka</b>	<b>55</b>
2.1	Nizovi . . . . .	55
2.2	Pokazivači . . . . .	72
2.3	Niske . . . . .	75
2.4	Višedimenzioni nizovi . . . . .	85
2.5	Strukture . . . . .	98
<b>3</b>	<b>Ulaz i izlaz programa</b>	<b>109</b>
3.1	Argumenti komandne linije . . . . .	109
3.2	Datoteke . . . . .	111
<b>A</b>	<b>Ispitni rokovi</b>	<b>127</b>
A.1	Modul Matematika . . . . .	127
A.1.1	Praktični deo ispita, januar 2019. . . . .	127
A.1.2	Praktični deo ispita, februar 2019. . . . .	129
A.2	Modul Informatika . . . . .	131
A.2.1	Praktični deo ispita, januar 2019. . . . .	131
A.2.2	Praktični deo ispita, februar 2019. . . . .	133



# Predgovor

U okviru kursa *Programiranje 1*, koji se drži na prvoj godini na svim smevovima na Matematičkom fakultetu, vežbaju se zadaci koji imaju za cilj da studentima pomognu da nauče osnovne algoritme i strukture podataka koji se sreću u imperativnim programskim jezicima. Ova zbirka predstavlja objedinjen skup zadataka sa vežbi i praktikuma ovog kursa, kao i primere zadataka sa održanih ispita. Sva rešenja su data u programskom jeziku C, ali se većina zadataka može koristiti za vežbanje proizvoljnog imperativnog programskog jezika. Elektronska verzija zbirke i prpratna rešenja u elektronskom formatu, dostupna su besplatno u skladu sa navedenom licencom i mogu se naći na adresi <http://www.programiranje1.matf.bg.ac.rs/zbirka>.

Zbirka je podeljena u četiri poglavlja. U prvom poglavlju obrađene su uvodne teme koje obuhvataju osnovne elemente imperativnog programiranja koje se koriste u rešavanju svih ostalih zadataka u zbirci. Uvodne teme uključuju osnovne tipove podataka, elementarnu komunikaciju sa korisnikom, građenje izraza, upotrebu naredbi dodele i naredbi koje regulišu kontrolu toka programa (sekvenca, selekcija i iteracija) uključujući i izdvajanje logičkih celina u funkcije. Drugo poglavlje je posvećeno radu sa naprednijim tipovima podataka: nizovima (uključujući niske i višedimenzione nizove), pokazivačima i strukturama. Treće poglavlje posvećeno je dodatnim tehnikama koje se koriste za komunikaciju sa korisnikom. Obrađen je rad sa argumentima komandne linije, kao i rad sa datotekama. Dodatak sadrži primere dva ispitna roka iz jedne akademske godine. Većina zadataka je rešena, a teži zadaci su obeleženi zvezdicom.

Autori velikog broja zadataka ove zbirke su ujedno i autori same zbirke, ali postoje i zadaci za koje se ne može tačno utvrditi ko je originalni autor jer su zadacima davali svoje doprinose različiti asistenti koji su držali vežbe iz ovog kursa tokom prethodnih godina. Zbog toga smatramo da je naš osnovni doprinos što smo objedinili, precizno formulisali, rešili i detaljno iskomentarisali sve najvažnije zadatke koji su potrebni za uspešno savlađivanje konceptata koji se obrađuju u okviru kursa. Takođe, formulacije zadataka smo obogatili primerima koji upotpunjuju razumevanje zahteva zadataka i koji omogućavaju čitaocu

zbirke da proveri sopstvena rešenja. Primeri su dati u obliku testova i interakcija sa programom. Testovi su svedene prirode i obuhvataju samo jednostavne ulaze i izlaze iz programa. Interakcija sa programom obuhvata naizmeničnu interakciju čovek-računar u kojoj su ulazi i izlazi isprepletani. U zadacima koji zahtevaju rad sa argumentima komandne linije, navedeni su i primeri poziva programa, a u zadacima koji demonstriraju rad sa datotekama, i primeri ulaznih ili izlaznih datoteka. Test primeri koji su navedeni uz ispitne zadatke u dodatku su oni koji su korišćeni u okviru testiranja i ocenjivanja studentskih radova na ispitima.

Neizmerno zahvaljujemo recenzentima, Gordani Pavlović Lažetić i Draganu Uroševiću, na veoma pažljivom čitanju rukopisa i na brojnim korisnim sugestijama koje su unapredile kvalitet zbirke. Takođe, zahvaljujemo studentima koji su svojim aktivnim učešćem u nastavi pomogli i doprineli uobličavanju ovog materijala kao i svim kolegama koje su držale vežbe na ovom kursu.

Svi komentari i sugestije na sadržaj zbirke su dobrodošli i osećajte se slobodnim da ih pošaljete elektronskom poštom bilo kom autoru <sup>1</sup>.

*Autori*

---

<sup>1</sup>Adrese autora su: milena, jovana, danijela, andjelkaz, aleksandra\_kocic, sa nastavkom @matf.bg.ac.rs



# 1

## Osnovni elementi imperativnog programiranja

### 1.1 Izrazi

**Zadatak 1.1.1** Napisati program koji na standardni izlaz ispisuje poruku Zdravo svima!.

*Test 1*

```
|| IZLAZ:  
|| Zdravo svima!
```

**Zadatak 1.1.2** Napisati program koji za uneti ceo broj ispisuje njegov kvadrat i njegov kub.

*Primer 1*

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

*Primer 2*

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

**Zadatak 1.1.3** Napisati program koji za uneta dva cela broja  $x$  i  $y$  ispisuje njihov zbir, razliku, proizvod, ceo deo pri deljenju prvog broja drugim brojem i ostatak pri deljenju prvog broja drugim brojem. NAPOMENA: *Pretpostaviti da je unos ispravan.*<sup>1</sup>

---

<sup>1</sup>U zadacima sa ovom napomenom treba pretpostaviti da se na ulazu zadaje očekivani broj vrednosti očekivanog tipa  $u$ , gde je primereno, odgovarajućem formatu.

## 1 Osnovni elementi imperativnog programiranja

---

### Primer 1

```
INTERAKCIJA SA PROGRAMOM:
Unesite vrednost broja x: 7
Unesite vrednost broja y: 2
7 + 2 = 9
7 - 2 = 5
7 * 2 = 14
7 / 2 = 3
7 % 2 = 1
```

### Primer 2

```
INTERAKCIJA SA PROGRAMOM:
Unesite vrednost promenljive x: -3
Unesite vrednost promenljive y: 8
-3 + 8 = 5
-3 - 8 = -11
-3 * 8 = -24
-3 / 8 = 0
-3 % 8 = -3
```

**Zadatak 1.1.4** Napisati program koji pomaže kasirki da izračuna ukupan račun ako su poznate cene dva kupljena artikla. Cene artikala su pozitivni celi brojevi. NAPOMENA: *Pretpostaviti da je unos ispravan.*

### Primer 1

```
INTERAKCIJA SA PROGRAMOM:
Unesite cenu prvog artikla: 173
Unesite cenu drugog artikla: 2024
Ukupna cena: 2197
```

### Primer 2

```
INTERAKCIJA SA PROGRAMOM:
Unesite cenu prvog artikla: 384
Unesite cenu drugog artikla: 555
Ukupna cena: 939
```

**Zadatak 1.1.5** Napisati program koji za unetu količinu jabuka u kilogramima i unetu cenu po kilogramu ispisuje ukupnu cenu date količine jabuka. Obe ulazne vrednosti su pozitivni celi brojevi. NAPOMENA: *Pretpostaviti da je unos ispravan.*

### Primer 1

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

### Primer 1

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

**Zadatak 1.1.6** Napisati program koji pomaže kasirki da obračuna kusur koji treba da vrati kupcu. Za unetu cenu artikla, količinu artikla i iznos koji je kupac dao, program treba da ispiše vrednost kusura. Sve ulazne vrednosti su pozitivni celi brojevi. NAPOMENA: *Pretpostaviti da je unos ispravan.*

### Primer 1

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

### Primer 2

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

**Zadatak 1.1.7** Napisati program koji za uneta vremena poletanja i sletanja

aviona izražena u satima i minutima ispisuje dužinu trajanja leta. NAPOMENA: Pretpostaviti da su poletanje i sletanje u istom danu kao i da su sve vrednosti ispravno unete.

## Primer 1

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

## Primer 2

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

**Zadatak 1.1.8** Napisati program koji razmenjuje vrednosti dveju promenljivih  $x$  i  $y$ . Njihove vrednosti, kao dva cela broja, zadaje korisnik.

## Primer 1

```
INTERAKCIJA SA PROGRAMOM:
Unesite vrednosti x i y: 5 7
Pre zamene: x = 5, y = 7
Posle zamene: x = 7, y = 5
```

## Primer 2

```
INTERAKCIJA SA PROGRAMOM:
Unesite vrednosti x i y: 237 -592
Pre zamene: x = 237, y = -592
Posle zamene: x = -592, y = 237
```

**Zadatak 1.1.9** Date su dve celobrojne promenljive  $a$  i  $b$ . Napisati program koji promenljivoj  $a$  dodeljuje njihovu sumu, a promenljivoj  $b$  njihovu razliku. NAPOMENA: Ne koristiti pomoćne promenljive.

## Primer 1

```
INTERAKCIJA SA PROGRAMOM:
Unesite vrednosti a i b: 5 7
Nove vrednosti su: a = 12, b = -2
```

## Primer 2

```
INTERAKCIJA SA PROGRAMOM:
Unesite vrednosti a i b: 237 -592
Nove vrednosti su: a = -355, b = 829
```

**Zadatak 1.1.10** Napisati program koji za uneti pozitivan trocifreni broj ispisuje njegove cifre jedinica, desetica i stotina. NAPOMENA: Pretpostaviti da je unos ispravan.

## Primer 1

```
INTERAKCIJA SA PROGRAMOM:
Unesite trocifreni broj: 697
Cifra jedinica: 7
Cifra desetica: 9
Cifra stotina: 6
```

## Primer 2

```
INTERAKCIJA SA PROGRAMOM:
Unesite trocifreni broj: 504
Cifra jedinica: 4
Cifra desetica: 0
Cifra stotina: 5
```

**Zadatak 1.1.11** Napisati program koji za unetu cenu proizvoda ispisuje najmanji broj novčanica koje je potrebno izdvojiti prilikom plaćanja proizvoda. Na raspolaganju su novčanice od 5000, 2000, 1000, 500, 200, 100, 50, 20, 10 i

## 1 Osnovni elementi imperativnog programiranja

1 dinar. Cena proizvoda je pozitivan ceo broj. NAPOMENA: *Pretpostaviti da je unos ispravan.*

Primer 1  $X \rightarrow \text{11212}$

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

Primer 2

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

**Zadatak 1.1.12** Napisati program koji učitava pozitivan trocifreni broj i ispisuje broj dobijen obrtanjem njegovih cifara. NAPOMENA: *Pretpostaviti da je unos ispravan.*

Primer 1

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

Primer 2

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

**Zadatak 1.1.13** Napisati program koji za uneti pozitivan četvorocifreni broj:

- (a) izračunava proizvod cifara
- (b) izračunava razliku sume krajnjih i srednjih cifara
- (c) izračunava sumu kvadrata cifara
- (d) izračunava broj koji se dobija zapisom cifara u obrnutom poretku
- (e) izračunava broj koji se dobija zamenom cifre jedinice i cifre stotine

NAPOMENA: *Pretpostaviti da je unos ispravan.*

Primer 1

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

Primer 2

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

**Zadatak 1.1.14** Napisati program koji ispisuje broj koji se dobija izbacivanjem cifre desetica u unetom pozitivnom celom broju. NAPOMENA: *Pretpostaviti da je unos ispravan.*

*Primer 1*

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

*Primer 2*

```
INTERAKCIJA SA PROGRAMOM:
Unesite broj: 825
Rezultat: 85
```

**Zadatak 1.1.15** Napisati program koji učitava pozitivan ceo broj  $n$  i pozitivan dvocifreni broj  $m$  i ispisuje broj dobijen umetanjem broja  $m$  između cifre stotina i cifre hiljada broja  $n$ . NAPOMENA: *Za neke ulazne podatke može se dobiti neočekivan rezultat zbog prekoračenja, što ilustruje test primer broj 2.*

*Primer 1*

```
INTERAKCIJA SA PROGRAMOM:
Unesite pozitivan ceo broj: 12345
Unesite pozitivan dvocifreni broj: 67
Rezultat: 1267345
```

*Primer 2*

```
INTERAKCIJA SA PROGRAMOM:
Unesite pozitivan ceo broj: 50000000
Unesite pozitivan dvocifreni broj: 12
Rezultat: 705044704
```

**Zadatak 1.1.16** Napisati program koji učitava dužinu dijagonale monitora izraženu u inčima, konvertuje je u centimetre i ispisuje zaokruženu na dve decimale. UPUTSTVO: *Jedan inč ima 2,54 centimetra.*

*Primer 1*

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

*Primer 2*

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

**Zadatak 1.1.17** Napisati program koji učitava dužinu pređenog puta izraženu u miljama, konvertuje je u kilometre i ispisuje zaokruženu na dve decimale. UPUTSTVO: *Jedna milja ima 1,609344 kilometara.*

*Primer 1*

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

*Primer 2*

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

**Zadatak 1.1.18** Napisati program koji učitava težinu avionskog tereta izraženu u funtama, konvertuje je u kilograme i ispisuje zaokruženu na dve decimale. UPUTSTVO: *Jedna funta ima 0,45359237 kilograma.*

## 1 Osnovni elementi imperativnog programiranja

---

### Primer 1

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

### Primer 2

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

**Zadatak 1.1.19** Napisati program koji učitava temperaturu izraženu u farenhajtima, konvertuje tu vrednost u celzijuse i ispisuje je zaokruženu na dve decimale. NAPOMENA: *Pretpostaviti da je unos ispravan.* UPUTSTVO: *Veza između farenhajta i celzijusa je zadata narednom formulom  $F = \frac{9 \cdot C}{5} + 32$ .*

### Primer 1

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

### Primer 2

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

**Zadatak 1.1.20** Napisati program koji za unete realne vrednosti  $a_{11}$ ,  $a_{12}$ ,  $a_{21}$ ,  $a_{22}$  ispisuje vrednost determinante matrice:

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

Pri ispisu vrednost zaokružiti na četiri decimale.

### Primer 1

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

### Primer 2

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

### Primer 3

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

**Zadatak 1.1.21** Napisati program koji za unete realne vrednosti dužina stranica pravougaonika ispisuje njegov obim i površinu. Ispisati tražene vrednosti zaokružene na dve decimale. NAPOMENA: *Pretpostaviti da je unos ispravan.*

### Primer 1

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

### Primer 2

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

**Zadatak 1.1.22** Napisati program koji za unetu realnu vrednost dužine poluprečnika kruga ispisuje njegov obim i površinu zaokružene na dve decimale. NAPOMENA: *Pretpostaviti da je unos ispravan.*

## Primer 1

```

INTERAKCIJA SA PROGRAMOM:
Unesite poluprecnik: 4.2
Obim: 26.39
Povrsina: 55.42

```

## Primer 2

```

INTERAKCIJA SA PROGRAMOM:
Unesite poluprecnik: 14.932
Obim: 93.82
Povrsina: 700.46

```

**Zadatak 1.1.23** Napisati program koji za unetu realnu vrednost dužine stranice jednakostraničnog trougla ispisuje njegov obim i površinu zaokružene na dve decimale. NAPOMENA: *Pretpostaviti da je unos ispravan.* UPUTSTVO: *Za računanje površine jednakostraničnog trougla može se iskoristiti obrazac  $P = \frac{a^2 \cdot \sqrt{3}}{4}$  pri čemu je  $a$  dužina stranice. Za računanje korena broja koristiti funkciju `sqrt` čija se deklaracija nalazi u zaglavlju `math.h`.*

## Primer 1

```

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

```

## Primer 2

```

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

```

**Zadatak 1.1.24** Napisati program koji za unete realne vrednosti dužina stranica trougla ispisuje njegov obim i površinu zaokružene na dve decimale. NAPOMENA: *Pretpostaviti da je unos ispravan.* UPUTSTVO: *Za računanje površine trougla može se koristiti Heronov obrazac  $P = \sqrt{S \cdot (S - a) \cdot (S - b) \cdot (S - c)}$ , pri čemu su  $a$ ,  $b$  i  $c$  dužine stranica, a  $S$  je poluobim.*

## Primer 1

```

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

```

## Primer 2

```

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

```

**Zadatak 1.1.25** Pravougaonik čije su stranice paralelne koordinatnim osama zadat je svojim realnim koordinatama naspramnih temena (gornje levo i donje desno teme). Napisati program koji ispisuje njegov obim i površinu zaokružene na dve decimale. NAPOMENA: *Pretpostaviti da je unos ispravan.*

## Primer 1

```

INTERAKCIJA SA PROGRAMOM:
Unesite koordinate gornjeg levog temena: 4.3 5.8
Unesite koordinate donjeg desnog temena: 6.7 2.3
Obim: 11.80
Povrsina: 8.40

```

### Primer 2

```
INTERAKCIJA SA PROGRAMOM:
Unesite koordinate gornjeg levog temena: -3.7 8.23
Unesite koordinate donjeg desnog temena: -0.56 2
Obim: 18.74
Povrsina: 19.56
```

**Zadatak 1.1.26** Napisati program koji za tri uneta cela broja ispisuje njihovu aritmetičku sredinu zaokruženu na dve decimale.

### Primer 1

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

### Primer 2

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

**Zadatak 1.1.27** Napisati program koji pomaže moleru da izračuna površinu zidova prostorije koju treba da okreči. Za unete celobrojne vrednosti dimenzije sobe u metrima (dužinu, širinu i visinu), program treba da ispiše površinu zidova za krečenje pod pretpostavkom da na vrata i prozore otpada oko 20%. Omogućiti i da na osnovu unete celobrojne cene usluge po kvadratnom metru program izračuna ukupnu cenu krečenja. Sve realne vrednosti ispisati zaokružene na dve decimale. NAPOMENA: *Pretpostaviti da je unos ispravan.*

### Primer 1

```
INTERAKCIJA SA PROGRAMOM:
Unesite dimenzije sobe: 4 4 3
Unesite cenu po m2: 500
Moler treba da okreći 51.20 m2
Cena krecenja: 25600.00
```

### Primer 2

```
INTERAKCIJA SA PROGRAMOM:
Unesite dimenzije sobe: 13 17 3
Unesite cenu po m2: 475
Moler treba da okreći 320.80 m2
Cena krecenja: 152380.00
```

**Zadatak 1.1.28** Napisati program koji za unete pozitivne cele brojeve  $x$ ,  $p$  i  $c$  ispisuje broj koji se dobija ubacivanjem cifre  $c$  u broj  $x$  na poziciju  $p$ . Pretpostaviti da numeracija cifara počinje od nule, odnosno da se cifra najmanje težine nalazi se na nultoj poziciji. NAPOMENA: *Pretpostaviti da je unos ispravan.* UPUTSTVO: *Koristiti funkciju  $\text{pow}$  čija se deklaracija nalazi u zaglavlju  $\text{math.h}$ .*

### Primer 1

```
INTERAKCIJA SA PROGRAMOM:
Unesite redom x, p i c: 140 1 2
Rezultat: 1420
```

### Primer 2

```
INTERAKCIJA SA PROGRAMOM:
Unesite redom x, p i c: 12345 2 9
Rezultat: 123945
```



**Zadatak 1.1.29** Napisati program koji za uneta dva cela broja  $a$  i  $b$  dodeljuje promenljivoj *rezultat* vrednost 1 ako važi uslov:

- (a)  $a$  i  $b$  su različiti brojevi
- (b)  $a$  i  $b$  su parni brojevi
- (c)  $a$  i  $b$  su pozitivni brojevi, ne veći od 100

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

*Primer 1*

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

*Primer 2*

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

**Zadatak 1.1.30** Napisati program koji za uneta dva cela broja ispisuje njihov maksimum.

*Primer 1*

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

*Primer 2*

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

**Zadatak 1.1.31** Napisati program koji za uneta dva cela broja ispisuje njihov minimum.

*Primer 1*

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

*Primer 2*

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

**Zadatak 1.1.32** Napisati program koji za zadate realne vrednosti  $x$  i  $y$  ispisuje vrednost sledećeg izraza

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

zaokruženu na dve decimale.

*Primer 1*

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

*Primer 2*

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

### 1.2 Grananja

**Zadatak 1.2.1** Napisati program koji ispisuje najmanji od tri uneta cela broja.

*Primer 1*

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesite tri cela broja:  
|| 5 18 -1  
|| Najmanji: -1
```

*Primer 2*

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesite tri cela broja:  
|| 0 43 16  
|| Najmanji: 0
```

*Primer 3*

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesite tri cela broja:  
|| -5 -5 -5  
|| Najmanji: -5
```

**Zadatak 1.2.2** Napisati program koji za uneti realan broj ispisuje njegovu apsolutnu vrednost zaokruženu na dve decimale.

*Primer 1*

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesite jedan realan broj:  
|| 7.42  
|| Apsolutna vrednost: 7.42
```

*Primer 2*

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesite jedan realan broj:  
|| -562.428  
|| Apsolutna vrednost: 562.43
```

*Primer 3*

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesite jedan realan broj:  
|| 0  
|| Apsolutna vrednost: 0.00
```

**Zadatak 1.2.3** Napisati program koji za uneti ceo broj ispisuje njegovu recipročnu vrednost zaokruženu na četiri decimale. U slučaju neispravnog unosa, ispisati odgovarajuću poruku o grešci.

*Primer 1*

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

*Primer 2*

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

*Primer 3*

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesite jedan ceo broj: 0  
|| Greska: nedozvoljeno je deljenje nulom.
```

*Primer 4*

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

**Zadatak 1.2.4** Napisati program koji učitava tri cela broja i ispisuje zbir pozitivnih.

*Primer 1*

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

*Primer 2*

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

*Primer 3*

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesite tri cela broja:  
|| 16 2 576  
|| Zbir pozitivnih: 594
```

**Zadatak 1.2.5** U prodavnici je organizovana akcija da svaki kupac dobije najjeftiniji od tri artikla za jedan dinar. Napisati program koji za unete cene tri artikla izračunava ukupnu cenu, kao i koliko dinara se uštedi zahvaljujući popustu. Cene artikala su pozitivni celi brojevi. U slučaju neispravnog unosa, ispisati odgovarajuću poruku o grešci.

*Primer 1*

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

*Primer 2*

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

*Primer 3*

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

*Primer 4*

```
INTERAKCIJA SA PROGRAMOM:
Unesite tri cene: 247 -133 126
Greska: neispravan unos cene.
```

**Zadatak 1.2.6** Napisati program koji za uneto vreme u formatu *sat:minut* ispisuje koliko je sati i minuta ostalo do ponoći. Broj sati treba da bude iz intervala  $[0, 24)$ , a broj minuta iz intervala  $[0, 60)$ . U slučaju neispravnog unosa, ispisati odgovarajuću poruku o grešci.

*Primer 1*

```
INTERAKCIJA SA PROGRAMOM:
Unesite vreme: 18:19
Do ponoci: 5 sati i 41 minuta
```

*Primer 2*

```
INTERAKCIJA SA PROGRAMOM:
Unesite vreme: 23:7
Do ponoci: 0 sati i 53 minuta
```

*Primer 3*

```
INTERAKCIJA SA PROGRAMOM:
Unesite vreme: 24:20
Greska: neispravan unos vremena.
```

*Primer 4*

```
INTERAKCIJA SA PROGRAMOM:
Unesite vreme: 14:0
Do ponoci: 10 sati i 0 minuta
```

**Zadatak 1.2.7** Napisati program koji za unetu godinu ispisuje da li je prestupna. Godina je neoznačen ceo broj.

*Primer 1*

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

*Primer 2*

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

*Primer 3*

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

**Zadatak 1.2.8** Napisati program koji za učitani karakter ispisuje uneti karakter i njegov ASCII kod. Ukoliko je uneti karakter malo (veliko) slovo, ispisati i odgovarajuće veliko (malo) slovo i njegov ASCII kod.

### Primer 1

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

### Primer 2

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

### Primer 3

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

### Primer 4

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

**Zadatak 1.2.9** Napisati program koji učitava tri karaktera. Ispitati da li među unetim karakterima ima cifara i ako je tako odrediti proizvod tih cifara. Ukoliko među unetim karakterima nema cifara, program treba da ispiše odgovarajuću poruku. NAPOMENA: *Karakter koji se unose su razdvojeni blanko znacima.*

### Primer 1

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesite karaktere: A 5 3  
|| Proizvod cifara: 15
```

### Primer 2

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesite karaktere: k ! m  
|| Medju unetim karakterima nema cifara.
```

### Primer 3

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesite karaktere: 9 9 9  
|| Proizvod cifara: 729
```

### Primer 4

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesite karaktere: a 8 0  
|| Proizvod cifara: 0
```

**Zadatak 1.2.10** Kasirka unosi šifru artikla koja se zadaje u formi tri spojena karaktera koji mogu biti mala slova, velika slova ili cifre. U kasi su sve šifre zapisane malim slovima i ciframa. Napisati program koji kasirkin unos konvertuje u unos koji je odgovarajući za kasu, tj. koji sva velika slova pretvara u odgovarajuća mala, a ostale karaktere ne menja. U slučaju neispravnog unosa šifre, ispisati odgovarajuću poruku o grešci.

### Primer 1

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesite šifru: aBc  
|| Rezultat: abc
```

### Primer 2

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesite šifru: a?!  
|| Greška: ? je neispravan karakter.
```

*Primer 3*

```

|| INTERAKCIJA SA PROGRAMOM:
|| Unesite karaktere: 545
|| Rezultat: 5a5

```

*Primer 4*

```

|| INTERAKCIJA SA PROGRAMOM:
|| Unesite karaktere: 123
|| Rezultat: 123

```

**Zadatak 1.2.11** Napisati program koji za uneti četvorocifreni broj ispisuje njegovu najveću cifru. U slučaju neispravnog unosa, ispisati odgovarajuću poruku o grešci.

*Primer 1*

```

|| INTERAKCIJA SA PROGRAMOM:
|| Unesite četvorocifreni broj: 6835
|| Najveća cifra je: 8

```

*Primer 2*

```

|| INTERAKCIJA SA PROGRAMOM:
|| Unesite četvorocifreni broj: 7777
|| Najveća cifra je: 7

```

*Primer 3*

```

|| INTERAKCIJA SA PROGRAMOM:
|| Unesite četvorocifreni broj: 238
|| Greska: niste uneli četvorocifreni broj.

```

*Primer 4*

```

|| INTERAKCIJA SA PROGRAMOM:
|| Unesite četvorocifreni broj: -2002
|| Najveća cifra je: 2

```

**Zadatak 1.2.12** Trocifreni broj je Armstrongov ako je jednak zbiru kubova svojih cifara. Napisati program koji za dati pozitivan trocifreni broj proverava da li je Armstrongov. U slučaju neispravnog unosa, ispisati odgovarajuću poruku o grešci.

*Primer 1*

```

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

```

*Primer 2*

```

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

```

*Primer 3*

```

|| INTERAKCIJA SA PROGRAMOM:
|| Unesite pozitivan trocifreni broj:
|| 84
|| Greska: niste uneli pozitivan trocifreni broj.

```

*Primer 4*

```

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

```

**Zadatak 1.2.13** Napisati program koji ispisuje proizvod parnih cifara unetog četvorocifrenog broja. U slučaju neispravnog unosa, ispisati odgovarajuću poruku o grešci.

## 1 Osnovni elementi imperativnog programiranja

---

### Primer 1

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

### Primer 2

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

### Primer 3

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesite cetvorocifreni broj: 288  
|| Greska: niste uneli cetvorocifreni broj.
```

### Primer 4

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

**Zadatak 1.2.14** Napisati program koji učitava četvorocifreni broj i ispisuje broj koji se dobija kada se unetom broju razmene najmanja i najveća cifra. U slučaju da se najmanja ili najveća cifra pojavljuju na više pozicija, uzeti prvo pojavljivanje, gledajući sa desna na levo. U slučaju neispravnog unosa, ispisati odgovarajuću poruku o grešci.

### Primer 1

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

### Primer 2

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesite cetvorocifreni broj: 1192  
|| Rezultat: 1912
```

### Primer 3

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesite cetvorocifreni broj: 247  
|| Greska: niste uneli cetvorocifreni broj.
```

### Primer 4

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

**Zadatak 1.2.15** Napisati program koji za uneti četvorocifreni broj proverava da li su njegove cifre uređene neopadajuće, nerastuće ili nisu uređene i štampa odgovarajuću poruku. U slučaju neispravnog unosa, ispisati odgovarajuću poruku o grešci.

### Primer 1

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

### Primer 2

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

### Primer 3

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesite cetvorocifreni broj: 88  
|| Greska: niste uneli cetvorocifreni broj.
```

### Primer 4

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

**Zadatak 1.2.16** Napisati program koji ispituje da li se tačke  $A(x_1, y_1)$  i

$B(x_2, y_2)$  nalaze u istom kvadrantu. Koordinate tačaka su realni brojevi jednostruke tačnosti.

*Primer 1*

```
INTERAKCIJA SA PROGRAMOM:
Unesite koordinate tacke A: 1.5 6
Unesite koordinate tacke B: 2.33 9.8
Tacke se nalaze u istom kvadrantu.
```

*Primer 2*

```
INTERAKCIJA SA PROGRAMOM:
Unesite koordinate tacke A: -3 6
Unesite koordinate tacke B: 0.33 -5
Tacke se ne nalaze u istom kvadrantu.
```

*Primer 3*

```
INTERAKCIJA SA PROGRAMOM:
Unesite koordinate tacke A: 0 -6
Unesite koordinate tacke B: -1 -99.66
Tacke se nalaze u istom kvadrantu.
```

*Primer 4*

```
INTERAKCIJA SA PROGRAMOM:
Unesite koordinate tacke A: 3 -6
Unesite koordinate tacke B: -0.33 0
Tacke se ne nalaze u istom kvadrantu.
```

**Zadatak 1.2.17** Napisati program koji ispituje da li se tačke  $A(x_1, y_1)$ ,  $B(x_2, y_2)$  i  $C(x_3, y_3)$  nalaze na istoj pravoj.

*Primer 1*

```
INTERAKCIJA SA PROGRAMOM:
Unesite koordinate tacke A: 1.5 6
Unesite koordinate tacke B: -2.5 -10
Unesite koordinate tacke C: 3 12
Tacke se nalaze na istoj pravoj.
```

*Primer 2*

```
INTERAKCIJA SA PROGRAMOM:
Unesite koordinate tacke A: -1.5 3
Unesite koordinate tacke B: -0.4 9.8
Unesite koordinate tacke C: 2 3
Tacke se ne nalaze na istoj pravoj.
```

*Primer 3*

```
INTERAKCIJA SA PROGRAMOM:
Unesite koordinate tacke A: 1.55 6
Unesite koordinate tacke B: -8.4 9.8
Unesite koordinate tacke C: 5 4.682412
Tacke se nalaze na istoj pravoj.
```

*Primer 4*

```
INTERAKCIJA SA PROGRAMOM:
Unesite koordinate tacke A: 5.5 3.5
Unesite koordinate tacke B: 5.5 3.5
Unesite koordinate tacke C: 5.5 3.5
Tacke se nalaze na istoj pravoj.
```

*Primer 5*

```
INTERAKCIJA SA PROGRAMOM:
Unesite koordinate tacke A: 1 2
Unesite koordinate tacke B: 1 2
Unesite koordinate tacke C: -56 1.3
Tacke se nalaze na istoj pravoj.
```

*Primer 6*

```
INTERAKCIJA SA PROGRAMOM:
Unesite koordinate tacke A: 3.4 3.5
Unesite koordinate tacke B: -10 -1
Unesite koordinate tacke C: -10 -1
Tacke se nalaze na istoj pravoj.
```

**Zadatak 1.2.18** Napisati program za rad sa intervalima. Za dva celobrojna intervala  $[a_1, b_1]$  i  $[a_2, b_2]$  program treba da odredi:

- dužinu preseka datih intervala
- presečni interval datih intervala

## 1 Osnovni elementi imperativnog programiranja

---

- (c) dužinu dela prave koju pokrivaju dati intervali
- (d) najmanji interval koji sadrži date intervale.

### Primer 1

```
INTERAKCIJA SA PROGRAMOM:  
Unesite a1, b1, a2 i b2: 2 9 4 11  
Duzina preseka: 5  
Presecni interval: [4,9]  
Duzina koju pokrivaju: 9  
Najmanji interval: [2, 11]
```

### Primer 2

```
INTERAKCIJA SA PROGRAMOM:  
Unesite a1, b1, a2 i b2: 1 2 10 13  
Duzina preseka: 0  
Presecni interval: prazan  
Duzina koju pokrivaju: 4  
Najmanji interval: [1, 13]
```

**Zadatak 1.2.19** Napisati program koji za unete koeficijente kvadratne jednačine ispisuje koliko realnih rešenja jednačina ima i ako ih ima, ispisuje ih zaokružene na dve decimale.

### Primer 1

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

### Primer 2

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

**Zadatak 1.2.20** U nizu 12345678910111213...9899 ispisani su redom brojevi od 1 do 99. Napisati program koji za uneti ceo broj  $k$  ( $1 \leq k \leq 189$ ) ispisuje cifru koja se nalazi na  $k$ -toj poziciji datog niza. U slučaju neispravnog unosa, ispisati odgovarajuću poruku o grešci.

### Primer 1

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

### Primer 2

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

### Primer 3

```
INTERAKCIJA SA PROGRAMOM:  
Unesite k: 200  
Greska: neispravan unos pozicije.
```

### Primer 4

```
INTERAKCIJA SA PROGRAMOM:  
Unesite k: 10  
Na 10-toj poziciji je broj 1.
```

**Zadatak 1.2.21** Data je funkcija  $f(x) = 2 \cdot \cos(x) - x^3$ . Napisati program koji za učitane vrednosti realne promenljive  $x$  i vrednosti celobrojne promenljive  $k$  koja može biti 1, 2 ili 3 izračunava vrednost funkcije  $F(x, k)$  koja se dobija tako što se funkcija  $f$  primeni  $k$ -puta ( $F(x, 1) = f(x)$ ,  $F(x, 2) = f(f(x))$ ,  $F(x, 3) = f(f(f(x)))$ ). Dobijenu vrednosti ispisati zaokruženu na dve decimale. U slučaju neispravnog unosa, ispisati odgovarajuću poruku o grešci.



*Primer 1*

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

*Primer 2*

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

*Primer 3*

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

**Zadatak 1.2.22** Napisati program koji za uneti redni broj dana u nedelji ispisuje ime odgovarajućeg dana. U slučaju neispravnog unosa, ispisati odgovarajuću poruku o grešci.

*Primer 1*

```
INTERAKCIJA SA PROGRAMOM:
Unesite broj: 4
U pitanju je: četvrtak
```

*Primer 2*

```
INTERAKCIJA SA PROGRAMOM:
Unesite broj: 8
Greska: neispravan unos
dana.
```

*Primer 3*

```
INTERAKCIJA SA PROGRAMOM:
Unesite broj: 7
U pitanju je: nedelja
```

**Zadatak 1.2.23** Napisati program koji za uneti karakter ispituje da li je samoglasnik ili ne.

*Primer 1*

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

*Primer 2*

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

*Primer 3*

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

*Primer 4*

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

**Zadatak 1.2.24** Napisati program koji učitava dva cela broja i jedan od karaktera +, -, \*, / ili % i ispisuje vrednost izraza dobijenog primenom date operacije nad učitanim vrednostima. U slučaju neispravnog unosa, ispisati odgovarajuću poruku o grešci.

*Primer 1*

```
INTERAKCIJA SA PROGRAMOM:
Unesite izraz: 8 - 11
Rezultat je: -3
```

*Primer 2*

```
INTERAKCIJA SA PROGRAMOM:
Unesite izraz: 14 / 0
Greska: deljenje nulom.
```

## 1 Osnovni elementi imperativnog programiranja

---

### Primer 3

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesite izraz: 5 ? 7  
|| Greska: nepoznat operator.
```

### Primer 4

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesite izraz: 19 / 5  
|| Rezultat je: 3
```

**Zadatak 1.2.25** Napisati program koji za uneti datum u formatu *dan.mesec.* ispisuje godišnje doba kojem pripadaju. NAPOMENA: *Pretpostaviti da je unos ispravan.*

### Primer 1

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesite dan i mesec: 14.10.  
|| jesen
```

### Primer 2

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesite dan i mesec: 2.8.  
|| leto
```

### Primer 3

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesite dan i mesec: 27.2.  
|| zima
```

**Zadatak 1.2.26** Napisati program koji za unetu godinu i mesec ispisuje naziv meseca kao i koliko dana ima u tom mesecu te godine. U slučaju neispravnog unosa, ispisati odgovarajuću poruku o grešci.

### Primer 1

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesite godinu: 2018  
|| Unesite mesec: 1  
|| Januar, 31 dan
```

### Primer 2

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesite godinu: 2000  
|| Unesite mesec: 2  
|| Februar, 29 dana
```

### Primer 3

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesite godinu: 2018  
|| Unesite mesec: 13  
|| Greska: neispravan unos  
|| meseca.
```

**Zadatak 1.2.27** Napisati program koji za uneti datum u formatu *dan.mesec.godina.* proverava da li je korektan.

### Primer 1

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

### Primer 2

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

**Zadatak 1.2.28** Napisati program koji za korektno unet datum u formatu *dan.mesec.godina.* ispisuje datum prethodnog dana.

### Primer 1

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

### Primer 2

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

### Primer 3

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesite datum:  
|| 1.1.2019.  
|| Prethodni datum:  
|| 31.12.2018.
```

**Zadatak 1.2.29** Napisati program koji za korektno unet datum u formatu *dan.mesec.godina.* ispisuje datum narednog dana.

*Primer 1*

```
INTERAKCIJA SA PROGRAMOM:
Unesite datum:
30.4.2008.
Naredni datum:
1.5.2008.
```

*Primer 2*

```
INTERAKCIJA SA PROGRAMOM:
Unesite datum:
1.12.2005.
Naredni datum:
2.12.2005.
```

*Primer 3*

```
INTERAKCIJA SA PROGRAMOM:
Unesite datum:
31.12.2008.
Naredni datum:
1.1.2009.
```

\* **Zadatak 1.2.30** Polje šahovske table se definiše parom celih brojeva  $(x, y)$ ,  $1 \leq x, y \leq 8$ , gde je  $x$  redni broj reda, a  $y$  redni broj kolone. Napisati program koji za unete parove  $(k, l)$  i  $(m, n)$  proverava

- (a) da li su polja  $(k, l)$  i  $(m, n)$  iste boje
- (b) da li kraljica sa  $(k, l)$  ugrožava polje  $(m, n)$
- (c) da li konj sa  $(k, l)$  ugrožava polje  $(m, n)$

Pretpostaviti da je polje  $(1, 1)$  crno i da predstavlja donji levi ugao šahovske table. U slučaju neispravnog unosa, ispisati odgovarajuću poruku o grešci.

*Primer 1*

```
INTERAKCIJA SA PROGRAMOM:
Unesite (k,l): 1 1
Unesite (m,n): 2 2
Polja su iste boje.
Kraljica sa (1,1) ugrozava (2,2).
Konj sa (1,1) ne ugrozava (2,2).
```

*Primer 2*

```
INTERAKCIJA SA PROGRAMOM:
Unesite (k,l): 1 1
Unesite (m,n): 3 2
Polja su razlicite boje.
Kraljica sa (1,1) ne ugrozava (3,2).
Konj sa (1,1) ugrozava (3,2).
```

*Primer 3*

```
INTERAKCIJA SA PROGRAMOM:
Unesite (k,l): 5 4
Unesite (m,n): 3 3
Polja su razlicite boje.
Kraljica sa (5,4) ne ugrozava (3,3).
Konj sa (5,4) ugrozava (3,3).
```

*Primer 4*

```
INTERAKCIJA SA PROGRAMOM:
Unesite (k,l): 0 1
Unesite (m,n): 3 9
Greska: neispravna pozicija.
```

### 1.3 Petlje

**Zadatak 1.3.1** Napisati program koji pet puta ispisuje tekst *Mi volimo da programiramo.*

*Primer 1*

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

**Zadatak 1.3.2** Napisati program koji učitava pozitivan ceo broj  $n$  i  $n$  puta ispisuje tekst *Mi volimo da programiramo.* U slučaju neispravnog unosa, ispisati odgovarajuću poruku o grešci.

*Primer 1*

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesite broj n: 6  
|| Mi volimo da programiramo.  
|| Mi volimo da programiramo.  
|| Mi volimo da programiramo.  
|| Mi volimo da programiramo.  
|| Mi volimo da programiramo.  
|| Mi volimo da programiramo.
```

*Primer 2*

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesite broj n: 0  
|| Greska: pogresan unos broja n.
```

*Primer 3*

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesite broj n: -5  
|| Greska: pogresan unos broja n.
```

*Primer 4*

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesite broj n: 1  
|| Mi volimo da programiramo.
```

**Zadatak 1.3.3** Napisati program koji učitava nenegativan ceo broj  $n$  a potom ispisuje sve cele brojeve od 0 do  $n$ . U slučaju neispravnog unosa, ispisati odgovarajuću poruku o grešci.

*Primer 1*

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

*Primer 2*

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesite broj n: -10  
|| Greska: pogresan unos broja n.
```

**Zadatak 1.3.4** Napisati program koji učitava dva cela broja  $n$  i  $m$  ( $n \leq m$ ) i ispisuje sve cele brojeve iz intervala  $[n, m]$ . Pri rešavanju zadatka:

- (a) koristiti `while` petlju
- (b) koristiti `for` petlju

(c) koristiti `do-while` petlju

U slučaju neispravnog unosa, ispisati odgovarajuću poruku o grešci.

*Primer 1*

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite granice intervala: -2 4
|| -2 -1 0 1 2 3 4
```

*Primer 2*

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite granice intervala: 10 6
|| Greska: pogresan unos granica.
```

**Zadatak 1.3.5** Napisati program koji učitava nenegativan ceo broj  $n$  i izračunava njegov faktorijel. U slučaju neispravnog unosa, ispisati odgovarajuću poruku o grešci.

*Primer 1*

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite broj n: 18
|| 18! = 6402373705728000
```

*Primer 2*

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite broj n: 8
|| 8! = 40320
```

*Primer 3*

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite broj n: 40
|| Pri racunanju 40! ce doći do prekoracenja.
```

*Primer 4*

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite broj n: -5
|| Greska: neispravan unos.
```

**Zadatak 1.3.6** Napisati program koji učitava realan broj  $x$  i ceo nenegativan broj  $n$  i izračunava  $n$ -ti stepen broja  $x$ , tj.  $x^n$ . U slučaju neispravnog unosa, ispisati odgovarajuću poruku o grešci.

*Primer 1*

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

*Primer 2*

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

*Primer 3*

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite redom brojeve x i n: 11.43 -6
|| Greska: neispravan unos broja n.
```

*Primer 4*

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

**Zadatak 1.3.7** Napisati program koji učitava realan broj  $x$  i ceo broj  $n$  i izračunava  $n$ -ti stepen broja  $x$ .

*Primer 1*

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

*Primer 2*

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

**Zadatak 1.3.8** Pravi delioci celog broja su svi delioci osim jedinice i samog tog broja. Napisati program koji za uneti pozitivan ceo broj  $n$  ispisuje sve njegove prave delioce. U slučaju neispravnog unosa, ispisati odgovarajuću poruku o grešci.

*Primer 1*

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesite broj n: 100  
|| Pravi delioci: 2 4 5 10 20 25 50
```

*Primer 2*

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

**Zadatak 1.3.9** Napisati program koji za uneti ceo broj ispisuje broj dobijen uklanjanjem svih nula sa desne strane unetog broja.

*Primer 1*

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

*Primer 2*

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesite broj: 0  
|| Rezultat: 0
```

*Primer 3*

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesite broj: -1400  
|| Rezultat: -14
```

*Primer 4*

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesite broj: 147  
|| Rezultat: 147
```

**Zadatak 1.3.10** Napisati program koji učitava ceo broj i ispisuje njegove cifre u obrnutom poretku.

*Primer 1*

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

*Primer 2*

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

**Zadatak 1.3.11** Napisati program koji za uneti pozitivan ceo broj ispisuje da li je on deljiv sumom svojih cifara. U slučaju neispravnog unosa, ispisati odgovarajuću poruku o grešci.

*Primer 1*

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesite broj: 12  
|| Broj 12 je deljiv sa 3.
```

*Primer 2*

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesite broj: 2564  
|| Broj 2564 nije deljiv sa 17.
```

## Primer 3

```

INTERAKCIJA SA PROGRAMOM:
Unesite broj: -4
Greska: neispravan ulaz.

```

## Primer 4

```

INTERAKCIJA SA PROGRAMOM:
Unesite broj: 0
Greska: neispravan ulaz.

```

**Zadatak 1.3.12** Knjigovođa vodi evidenciju o transakcijama jedne firme i treba da napiše izveštaj o godišnjem poslovanju te firme. Firma je tokom godine imala  $t$  transakcija. Transakcije su predstavljene celim brojevima i u slučaju da je vrednost transakcije pozitivna, ta transakcija označava prihod firme, a u slučaju da je negativna rashod. Napisati program koji učitava nenegativan ceo broj  $t$  i podatke o  $t$  transakcijama i zatim izračunava i ispisuje ukupan prihod, ukupan rashod i zaradu, odnosno gubitak, koji je firma ostvarila tokom godine. U slučaju neispravnog unosa, ispisati odgovarajuću poruku o grešci.

## Primer 1

```

INTERAKCIJA SA PROGRAMOM:
Unesite broj t: 7
Unesite transakcije:
8 -50 45 2007 -67 -123 14
Prihod: 2074
Rashod: -240
Zarada: 1834

```

## Primer 2

```

INTERAKCIJA SA PROGRAMOM:
Unesite broj t: 5
Unesite transakcije:
-5 -20 -4 -200 -8
Prihod: 0
Rashod: -237
Gubitak: 237

```

## Primer 3

```

INTERAKCIJA SA PROGRAMOM:
Unesite broj t: 2
Unesite transakcije:
120 -120
Prihod: 120
Rashod: -120
Zarada: 0

```

## Primer 4

```

INTERAKCIJA SA PROGRAMOM:
Unesite broj t: -6
Greska: neispravan unos.

```

## Primer 5

```

INTERAKCIJA SA PROGRAMOM:
Unesite broj n: 0
Nema evidentiranih transakcija.

```

**Zadatak 1.3.13** Napisati program koji učitava pozitivan ceo broj  $n$ , a potom i  $n$  celih brojeva. Izračunati i ispisati zbir onih brojeva koji su istovremeno neparni i negativni. U slučaju neispravnog unosa, ispisati odgovarajuću poruku o grešci.

## Primer 1

```

INTERAKCIJA SA PROGRAMOM:
Unesite broj n: 5
Unesite n brojeva:
1 -5 -6 3 -11
Zbir neparnih i negativnih: -16

```

## Primer 2

```

INTERAKCIJA SA PROGRAMOM:
Unesite broj n: 4
Unesite n brojeva:
5 8 13 17
Zbir neparnih i negativnih: 0

```

## Primer 3

```

INTERAKCIJA SA PROGRAMOM:
Unesite broj n: -4
Greska: neispravan unos.

```

## Primer 4

```

INTERAKCIJA SA PROGRAMOM:
Unesite broj n: 0
Greska: neispravan unos.

```

## 1 Osnovni elementi imperativnog programiranja

---

**Zadatak 1.3.14** Napisati program koji učitava cele brojeve sve do unosa broja nula i ispisuje proizvod onih unetih brojeva koji su pozitivni.

### Primer 1

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

### Primer 2

```
INTERAKCIJA SA PROGRAMOM:
Unesite brojeve:
0
Nije unet nijedan broj.
```

### Primer 3

```
INTERAKCIJA SA PROGRAMOM:
Unesite brojeve:
-5 -200 -43 0
Medju unetim brojevima nema pozitivnih.
```

### Primer 4

```
INTERAKCIJA SA PROGRAMOM:
Unesite brojeve:
1 0
Proizvod pozitivnih brojeva je 1.
```

**Zadatak 1.3.15** Napisati program koji za uneti ceo broj proverava i ispisuje da li se cifra 5 nalazi u njegovom zapisu.

### Primer 1

```
INTERAKCIJA SA PROGRAMOM:
Unesite broj: 1857
Broj 1857 sadrzi cifru 5.
```

### Primer 2

```
INTERAKCIJA SA PROGRAMOM:
Unesite broj: 84
Broj 84 ne sadrzi cifru 5.
```

### Primer 3

```
INTERAKCIJA SA PROGRAMOM:
Unesite broj: -2515
Broj -2515 sadrzi cifru 5.
```

**Zadatak 1.3.16** Napisati program koji učitava cele brojeve sve do unosa broja nula, a zatim izračunava i ispisuje aritmetičku sredinu unetih brojeva na četiri decimale.

### Primer 1

```
INTERAKCIJA SA PROGRAMOM:
Unesite brojeve:
8 5 6 3 0
Aritmeticka sredina:
5.5000
```

### Primer 2

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

### Primer 3

```
INTERAKCIJA SA PROGRAMOM:
Unesite brojeve:
762 -12 800 2010
-356 899 -101 0
Aritmeticka sredina:
571.7143
```

**Zadatak 1.3.17** U prodavnici se nalaze artikli čije su cene pozitivni realni brojevi. Napisati program koji učitava cene artikala sve do unosa broja nula i izračunava i ispisuje prosečnu vrednost cena u radnji. U slučaju neispravnog unosa, ispisati odgovarajuću poruku o grešci.

### Primer 1

```
INTERAKCIJA SA PROGRAMOM:
Unesite cene:
8 5.2 6.11 3 0
Prosečna cena: 5.5775
```

### Primer 2

```
INTERAKCIJA SA PROGRAMOM:
Unesite cene:
6.32 -9
Greska: neispravan unos
cene.
```

### Primer 3

```
INTERAKCIJA SA PROGRAMOM:
Unesite cene:
0
Nisu unete cene.
```



**Zadatak 1.3.18** Napisati program koji učitava pozitivan ceo broj  $n$ , a potom i  $n$  realnih brojeva. Program ispisuje koliko puta je prilikom unosa došlo do *promene znaka*. Do promena znaka dolazi kada se nakon pozitivnog broja unese negativan broj ili kada se nakon negativnog broja unese pozitivan broj. U slučaju neispravnog unosa, ispisati odgovarajuću poruku o grešci.

*Primer 1*

```
INTERAKCIJA SA PROGRAMOM:
Unesite broj n: 9
Unesite brojeve:
7.82 4.3 -1.2 56.8 -3.4 -72.1 8.9 11.2 -11.2
Broj promena je 5.
```

*Primer 2*

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

*Primer 3*

```
INTERAKCIJA SA PROGRAMOM:
Unesite broj n: -6
Greska: neispravan unos.
```

*Primer 4*

```
INTERAKCIJA SA PROGRAMOM:
Unesite broj n: 0
Greska: neispravan unos.
```

**Zadatak 1.3.19** U prodavnici se nalazi  $n$  artikala čije su cene pozitivni realni brojevi. Napisati program koji učitava  $n$ , a potom i cenu svakog od  $n$  artikala i određuje i ispisuje najmanju cenu. U slučaju neispravnog unosa, ispisati odgovarajuću poruku o grešci.

*Primer 1*

```
INTERAKCIJA SA PROGRAMOM:
Unesite broj artikla: 6
Unesite cene artikala:
12 3.4 90 100.53 53.2 12.8
Najmanja cena: 3.400000
```

*Primer 2*

```
INTERAKCIJA SA PROGRAMOM:
Unesite broj artikla: 3
Unesite cene artikala:
4 -8 92
Greska: neispravan unos
cene.
```

*Primer 3*

```
INTERAKCIJA SA PROGRAMOM:
Unesite broj artikla: -9
Greska: neispravan unos.
```

**Zadatak 1.3.20** Nikola želi da obraduje baku i da joj kupi jedan poklon u radnji. On na raspolaganju ima  $m$  dinara. U radnji se nalazi  $n$  artikala i zanima ga koliko ima artikala u radnji čija cena je manja ili jednaka  $m$ . Napisati program koji pomaže Nikoli da odredi broj takvih artikala. Program učitava realan nenegativan broj  $m$ , ceo nenegativan broj  $n$  i  $n$  pozitivnih realnih brojeva. Ispisati koliko artikala ima cenu čija je vrednost manja ili jednaka  $m$ . **NAPOMENA:** *Pretpostaviti da je unos ispravan.*

*Primer 1*

```
INTERAKCIJA SA PROGRAMOM:
Nikolin budzet: 12.37
Unesite broj artikala: 5
Unesite cene artikala: 11 54.13 6 13 8
Ukupno artikala: 3
```

*Primer 2*

```
INTERAKCIJA SA PROGRAMOM:
Nikolin budzet: 2
Unesite broj artikala: 4
Unesite cene artikala: 1 11 4.32 3
Ukupno artikala: 1
```

## 1 Osnovni elementi imperativnog programiranja

---

**Zadatak 1.3.21** Napisati program koji učitava ceo nenegativan broj  $n$ ,  $n$  celih brojeva i zatim izračunava i ispisuje tražene vrednosti. U slučaju neispravnog unosa, ispisati odgovarajuću poruku o grešci.

- (a) Broj sa najvećom cifrom desetica. Ukoliko ima više takvih, ispisati prvi.

### Primer 1

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

### Primer 2

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

### Primer 3

```
INTERAKCIJA SA PROGRAMOM:
Unesite broj n: 4
Unesite brojeve:
100 200 300 400
Broj sa najvećom cifrom desetica: 100.
```

### Primer 4

```
INTERAKCIJA SA PROGRAMOM:
Unesite broj n: -12
Greska: neispravan unos.
```

- (b) Broj sa najvećim brojem cifara. Ukoliko ima više takvih, ispisati prvi.

### Primer 1

```
INTERAKCIJA SA PROGRAMOM:
Unesite broj n: 5
Unesite n brojeva: 18 -365 251 1 78
Najviše cifara ima broj -365.
```

### Primer 2

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

### Primer 3

```
INTERAKCIJA SA PROGRAMOM:
Unesite broj n: 0
Nisu uneti brojevi.
```

### Primer 4

```
INTERAKCIJA SA PROGRAMOM:
Unesite broj n: -7
Greska: neispravan unos.
```

### Primer 5

```
INTERAKCIJA SA PROGRAMOM:
Unesite broj n: 5
Unesite n brojeva: 0 1 2 -3 4
Najviše cifara ima broj 0.
```

### Primer 6

```
INTERAKCIJA SA PROGRAMOM:
Unesite broj n: 5
Unesite n brojeva: -5 4 -3 2 1
Najviše cifara ima broj -5.
```

- (c) Broj sa najvećom vodećom cifrom. Vodeća cifra je cifra najveće težine u

zapisu broja. Ukoliko ima više takvih, ispisati prvi.

#### Primer 1

```
INTERAKCIJA SA PROGRAMOM:
Unesite broj n: 5
Unesite n brojeva: 8 964 -32 511 27
Broj sa najvecom vodecom cifrom je 964.
```

#### Primer 2

```
INTERAKCIJA SA PROGRAMOM:
Unesite broj n: 3
Unesite n brojeva: 0 0 0
Broj sa najvecom vodecom cifrom je 0.
```

#### Primer 3

```
INTERAKCIJA SA PROGRAMOM:
Unesite broj n: 3
Unesite n brojeva: 41 669 -8
Broj sa najvecom vodecom cifrom je -8.
```

#### Primer 4

```
INTERAKCIJA SA PROGRAMOM:
Unesite broj n: 0
Nisu uneti brojevi.
```

**Zadatak 1.3.22** Vršena su merenja nadmorskih visina na određenom delu teritorije i naučnike zanima razlika između najveće i najmanje nadmorske visine. Napisati program koji učitava cele brojeve koji predstavljaju nadmorske visine sve do unosa broja nula i ispisuje razliku najveće i najmanje nadmorske visine.

#### Primer 1

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

#### Primer 2

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

#### Primer 3

```
INTERAKCIJA SA PROGRAMOM:
Unesite brojeve:
0
Nisu unete nadmorske
visine.
```

#### Primer 4

```
INTERAKCIJA SA PROGRAMOM:
Unesite brojeve:
-500 0
Razlika: 0
```

#### Primer 5

```
INTERAKCIJA SA PROGRAMOM:
Unesite brojeve:
-500 -300 -5000 0
Razlika: 4700
```

**Zadatak 1.3.23** Napisati program koji učitava ceo broj  $n$  ( $n > 1$ ), nenegativan ceo broj  $d$ , a zatim i  $n$  celih brojeva i izračunava i ispisuje koliko ima parova uzastopnih brojeva među unetim brojevima koji se nalaze na rastojanju  $d$ . Rastojanje između brojeva je definisano sa  $d(x, y) = |y - x|$ . U slučaju neispravnog unosa, ispisati odgovarajuću poruku o grešci.

#### Primer 1

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

#### Primer 2

```
INTERAKCIJA SA PROGRAMOM:
Unesite brojeve n i d: 5 0
Unesite n brojeva: 1 1 1 1 1
Broj parova: 4
```

## 1 Osnovni elementi imperativnog programiranja

---

### Primer 3

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

### Primer 4

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesite brojeve n i d: 1 3  
|| Greska: neispravan unos.
```

**Zadatak 1.3.24** Napisati program koji uneti pozitivan ceo broj transformiše tako što svaku parnu cifru u zapisu broja uveća za jedan. Ispisati dobijeni broj. U slučaju neispravnog unosa, ispisati odgovarajuću poruku o grešci.

### Primer 1

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

### Primer 2

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

### Primer 3

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

**Zadatak 1.3.25** Napisati program koji učitava jedan ceo broj i zatim formira i ispisuje broj koji se dobija izbacivanjem svake druge cifre polaznog broja, idući sa desna na levo.

### Primer 1

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesite broj: 21854  
|| Rezultat: 284
```

### Primer 2

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesite broj: -18  
|| Rezultat: -8
```

### Primer 3

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesite broj: 1  
|| Rezultat: 1
```

\* **Zadatak 1.3.26** Napisati program koji na osnovu unetog pozitivnog celog broja formira i ispisuje broj koji se dobija izbacivanjem cifara koje su u polaznom broju jednake zbiru svojih suseda. U slučaju neispravnog unosa, ispisati odgovarajuću poruku o grešci.

### Primer 1

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesite broj: 28631  
|| Rezultat: 2631
```

### Primer 2

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesite broj: 440  
|| Rezultat: 40
```

### Primer 3

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesite broj: -5  
|| Greska: neispravan unos.
```

\* **Zadatak 1.3.27** Broj je *palindrom* ukoliko se isto čita i sa leve i sa desne strane. Napisati program koji učitava pozitivan ceo broj i proverava da li je učitani broj palindrom. U slučaju neispravnog unosa, ispisati odgovarajuću poruku o grešci.

*Primer 1*

```

INTERAKCIJA SA PROGRAMOM:
Unesite broj: 25452
Broj je palindrom.

```

*Primer 2*

```

INTERAKCIJA SA PROGRAMOM:
Unesite broj: 895
Broj nije palindrom.

```

*Primer 3*

```

INTERAKCIJA SA PROGRAMOM:
Unesite broj: 5
Broj je palindrom.

```

**Zadatak 1.3.28** Fibonačijev niz počinje članovima 0 i 1, a svaki naredni član se dobija kao zbir prethodna dva. Napisati program koji učitava nenegativan ceo broj  $n$  i određuje i ispisuje  $n$ -ti član Fibonačijevog niza. Niz se indeksira počevši od nule. U slučaju neispravnog unosa, ispisati odgovarajuću poruku o grešci.

*Primer 1*

```

INTERAKCIJA SA PROGRAMOM:
Unesite broj n: 10
F[10] = 55

```

*Primer 2*

```

INTERAKCIJA SA PROGRAMOM:
Unesite broj n: -100
Greska: neispravan unos.

```

*Primer 3*

```

INTERAKCIJA SA PROGRAMOM:
Unesite broj n: 40
F[40] = 102334155

```

*Primer 4*

```

INTERAKCIJA SA PROGRAMOM:
Unesite broj n: 20
F[20] = 6765

```

**Zadatak 1.3.29** Niz prirodnih brojeva formira se prema sledećem pravilu:

$$a_{n+1} = \begin{cases} \frac{a_n}{2} & \text{ako je } a_n \text{ parno} \\ \frac{3 \cdot a_n + 1}{2} & \text{ako je } a_n \text{ neparno} \end{cases}$$

Napisati program koji za uneti početni član niza  $a_0$  (pozitivan ceo broj) štampa niz brojeva sve do onog člana niza koji je jednak 1. U slučaju neispravnog unosa, ispisati odgovarajuću poruku o grešci.

*Primer 1*

```

INTERAKCIJA SA PROGRAMOM:
Unesite prvi clan: 56
Clanovi niza:
56 28 14 7 11 17 26 13
20 10 5 8 4 2 1

```

*Primer 2*

```

INTERAKCIJA SA PROGRAMOM:
Unesite prvi clan: -48
Greska: neispravan unos.

```

*Primer 3*

```

INTERAKCIJA SA PROGRAMOM:
Unesite prvi clan: 67
Clanovi niza:
67 101 152 76 38 19 29
44 22 11 17 26 13 20 10
5 8 4 2 1

```

**\* Zadatak 1.3.30** Papir  $A_0$  ima površinu  $1m^2$  i odnos stranica  $1 : \sqrt{2}$ . Papir  $A_1$  dobija se podelom papira  $A_0$  po dužoj ivici. Papir  $A_2$  dobija se podelom  $A_1$  papira po dužoj ivici itd. Napisati program koji za uneti nenegativan broj  $k$  ispisuje dimenzije papira  $A_k$  u milimetrima. Rezultat ispisati kao celobrojne vrednosti. U slučaju neispravnog unosa, ispisati odgovarajuću poruku o grešci.

### Primer 1

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesite format papira: 4  
|| Dimenzije papira: 210 297
```

### Primer 2

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesite format papira: 0  
|| Dimenzije papira: 840 1189
```

### Primer 3

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesite format papira: -7  
|| Greska: neispravan unos.
```

### Primer 4

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesite format papira: 9  
|| Dimenzije papira: 37 52
```

**Zadatak 1.3.31** Napisati program koji učitava karaktere dok se ne unese karakter tačka, i ako je karakter malo slovo ispisuje odgovarajuće veliko, ako je karakter veliko slovo ispisuje odgovarajuće malo, a u suprotnom ispisuje isti karakter kao i uneti.

### Primer 1

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

### Primer 2

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

**Zadatak 1.3.32** Napisati program koji učitava karaktere sve do kraja ulaza, a potom ispisuje broj velikih slova, broj malih slova, broj cifara, broj belina i zbir unetih cifara.

### Primer 1

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesite tekst:  
|| Beograd - Nis 230km  
|| Uzice - Cacak 56.3km  
|| Subotica - Ruma 139km  
|| Velika slova: 6  
|| Mala slova: 32  
|| Cifre: 9  
|| Beline: 12  
|| Suma cifara: 32
```

### Primer 2

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesite tekst:  
|| Isli smo u Afriku da sadimo papriku.  
|| Velika slova: 2  
|| Mala slova: 27  
|| Cifre: 0  
|| Beline: 7  
|| Suma cifara: 0
```

**Zadatak 1.3.33** Program učitava pozitivan ceo broj  $n$ , a potom i  $n$  karaktera. Za svaki od samoglasnika ispisati koliko puta se pojavio među unetim karakterima. Ne praviti razliku između malih i velikih slova. U slučaju neispravnog unosa, ispisati odgovarajuću poruku o grešci.

## Primer 1

```

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

```

## Primer 2

```

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

```

## Primer 3

```

INTERAKCIJA SA PROGRAMOM:
Unesite broj n: -7
Greska: neispravan unos.

```

**Zadatak 1.3.34** Program učitava pozitivan ceo broj  $n$ , a zatim i  $n$  karaktera. Napisati program koji proverava da li se od unetih karaktera može napisati reč *Zima*. U slučaju neispravnog unosa, ispisati odgovarajuću poruku o grešci.

## Primer 1

```

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

```

## Primer 2

```

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

```

## Primer 3

```

INTERAKCIJA SA PROGRAMOM:
Unesite broj n: 0
Greska: neispravan unos.

```

**Zadatak 1.3.35** Napisati program koji učitava pozitivan ceo broj  $n$  i ispisuje vrednost sume kubova brojeva od 1 do  $n$ , odnosno  $S = 1 + 2^3 + 3^3 + \dots + n^3$ . U slučaju neispravnog unosa, ispisati odgovarajuću poruku o grešci.

## Primer 1

```

INTERAKCIJA SA PROGRAMOM:
Unesite broj n: 14
Suma kubova: 11025

```

## Primer 2

```

INTERAKCIJA SA PROGRAMOM:
Unesite broj n: 25
Suma kubova: 105625

```

## Primer 3

```

INTERAKCIJA SA PROGRAMOM:
Unesite broj n: 0
Greska: neispravan unos.

```

**Zadatak 1.3.36** Napisati program koji učitava pozitivan ceo broj  $n$  i ispisuje sumu kubova,  $S = 1 + 2^3 + 3^3 + \dots + k^3$ , za svaku vrednost  $k = 1, \dots, n$ . U slučaju neispravnog unosa, ispisati odgovarajuću poruku o grešci.

### Primer 1

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite broj n: 5
|| [k=1] Suma kubova: 1
|| [k=2] Suma kubova: 9
|| [k=3] Suma kubova: 36
|| [k=4] Suma kubova: 100
|| [k=5] Suma kubova: 225
```

### Primer 2

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite broj n: 8
|| [k=1] Suma kubova: 1
|| [k=2] Suma kubova: 9
|| [k=3] Suma kubova: 36
|| [k=4] Suma kubova: 100
|| [k=5] Suma kubova: 225
|| [k=6] Suma kubova: 441
|| [k=7] Suma kubova: 784
|| [k=8] Suma kubova: 1296
```

### Primer 3

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite broj n: -5
|| Greska: neispravan unos.
```

**Zadatak 1.3.37** Napisati program koji učitava realan broj  $x$  i pozitivan ceo broj  $n$  i izračunava i ispisuje sumu  $S = x + 2 \cdot x^2 + 3 \cdot x^3 + \dots + n \cdot x^n$ . U slučaju neispravnog unosa, ispisati odgovarajuću poruku o grešci.

### Primer 1

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

### Primer 2

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

### Primer 3

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite redom brojeve x i n: 5.5 0
|| Greska: neispravan unos.
```

### Primer 4

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite redom brojeve x i n: -0.5 -5
|| Greska: neispravan unos.
```

**Zadatak 1.3.38** Napisati program koji učitava realan broj  $x$  i pozitivan ceo broj  $n$  i izračunava i ispisuje sumu  $S = 1 + \frac{1}{x} + \frac{1}{x^2} + \dots + \frac{1}{x^n}$ . U slučaju neispravnog unosa, ispisati odgovarajuću poruku o grešci.

### Primer 1

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

### Primer 2

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

### Primer 3

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite redom brojeve x i n: 5.5 0
|| Greska: neispravan unos.
```

### Primer 4

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite redom brojeve x i n: -0.5 -5
|| Greska: neispravan unos.
```

\* **Zadatak 1.3.39** Napisati program koji učitava realne brojeve  $x$  i  $eps$  i sa tačnošću  $eps$  izračunava i ispisuje sumu  $S = 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \dots$ . Za sumu  $S$  se kaže da je izračunata sa tačnošću  $eps$  ako je apsolutna vrednost poslednjeg člana



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

## Primer 1

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

## Primer 2

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

\* **Zadatak 1.3.40** Napisati program koji učitava realne brojeve  $x$  i  $\text{eps}$  i sa zadatom tačnošću  $\text{eps}$  izračunava i ispisuje sumu  $S = 1 - x + \frac{x^2}{2!} - \frac{x^3}{3!} + \frac{x^4}{4!} - \frac{x^5}{5!} \dots$ . NAPOMENA: Voditi računa o efikasnosti rešenja i o mogućnosti prekoračenja.

## Primer 1

```
INTERAKCIJA SA PROGRAMOM:
Unesite x: 3
Unesite tacnost eps: 0.000001
S = 0.049787
```

## Primer 2

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

**Zadatak 1.3.41** Napisati program koji učitava realan broj  $x$  i pozitivan ceo broj  $n$  i izračunava proizvod  $P = (1 + \cos(x)) \cdot (1 + \cos(x^2)) \cdot \dots \cdot (1 + \cos(x^n))$ . U slučaju neispravnog unosa, ispisati odgovarajuću poruku o grešci. NAPOMENA: Voditi računa o efikasnosti rešenja.

## Primer 1

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

## Primer 2

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

## Primer 3

```
INTERAKCIJA SA PROGRAMOM:
Unesite redom brojeve x i n: 12 0
Greska: neispravan unos.
```

## Primer 4

```
INTERAKCIJA SA PROGRAMOM:
Unesite redom brojeve x i n: 12 -6
Greska: neispravan unos.
```

\* **Zadatak 1.3.42** Napisati program koji učitava pozitivan ceo broj  $n$  i ispisuje vrednost razlomka

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

U slučaju neispravnog unosa, ispisati odgovarajuću poruku o grešci.

*Primer 1*

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite broj n: 4
|| R = 0.697674
```

*Primer 2*

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite broj n: 20
|| R = 0.697775
```

*Primer 3*

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

\* **Zadatak 1.3.43** Napisati program koji učitava realan broj  $x$  i pozitivan ceo broj  $n$  i računa sumu

$$1 - \frac{x^2}{2!} + \frac{x^4}{4!} - \dots + (-1)^n \frac{x^{2n}}{(2n)!}.$$

U slučaju neispravnog unosa, ispisati odgovarajuću poruku o grešci. NAPOMENA: Voditi računa o efikasnosti rešenja i o mogućnosti prekoračenja.

*Primer 1*

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite x i n: 5.6 8
|| S = 0.779792
```

*Primer 2*

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite x i n: 14.32 11
|| S = -6714.066406
```

*Primer 3*

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite x i n: 2 -6
|| Greska: neispravan unos.
```

\* **Zadatak 1.3.44** Napisati program koji učitava pozitivan ceo broj  $n$  i koji računa proizvod

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

U slučaju neispravnog unosa, ispisati odgovarajuću poruku o grešci. NAPOMENA: Voditi računa o efikasnosti rešenja i o mogućnosti prekoračenja.

*Primer 1*

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

*Primer 2*

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

*Primer 3*

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

*Primer 4*

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

\* **Zadatak 1.3.45** Napisati program koji učitava neparan ceo broj  $n$  ( $n \geq 5$ ) i izračunava i ispisuje sumu

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

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

*Primer 1*

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

*Primer 2*

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

*Primer 3*

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite broj n: 20
|| Greska: neispravan unos.
```

*Primer 4*

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

**Zadatak 1.3.46** Napisati program koji učitava realne brojeve  $x$  i  $a$  i pozitivan ceo broj  $n$  i zatim izračunava i ispisuje vrednost izraza

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

U slučaju neispravnog unosa, ispisati odgovarajuću poruku o grešci.

*Primer 1*

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite brojeve x i a: 3.2 0.2
|| Unesite broj n: 5
|| Izraz = 135380494030332048.000000
```

*Primer 2*

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite brojeve x i a: 2 1
|| Unesite broj n: 3
|| Izraz = 10201.000000
```

*Primer 3*

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite brojeve x i a: 2.6 0.3
|| Unesite broj n: 3
|| Izraz = 5800.970129
```

*Primer 4*

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite brojeve x i a: 5.4 7
|| Unesite broj n: -2
|| Greska: neispravan unos.
```

**Zadatak 1.3.47** Napisati program koji za unetu pozitivnu celobrojnu vrednost  $n$  ispisuje odgovarajuće brojeve. NAPOMENA: *Pretpostaviti da je unos ispravan.*

- (a) Ispisati tablicu množenja brojeva od 1 do  $n$ .

Primer 1

```
INTERAKCIJA SA PROGRAMOM:
Unesite broj n: 1
1
```

Primer 2

```
INTERAKCIJA SA PROGRAMOM:
Unesite broj n: 2
1 2
2 4
```

Primer 3

```
INTERAKCIJA SA PROGRAMOM:
Unesite broj n: 4
1 2 3 4
2 4 6 8
3 6 9 12
4 8 12 16
```

- (b) Ispisati sve brojeve od 1 do  $n^2$ , po  $n$  brojeva u jednoj vrsti.

Primer 1

```
INTERAKCIJA SA PROGRAMOM:
Unesite broj n: 3
1 2 3
4 5 6
7 8 9
```

Primer 2

```
INTERAKCIJA SA PROGRAMOM:
Unesite broj n: 4
1 2 3 4
5 6 7 8
9 10 11 12
13 14 15 16
```

- (c) Ispisati tablicu brojeva tako da su u prvoj vrsti svi brojevi od 1 do  $n$ , a u svakoj narednoj brojevi dobijeni rotiranjem prethodne vrste za jedno mesto u levo.

Primer 1

```
INTERAKCIJA SA PROGRAMOM:
Unesite broj n: 3
1 2 3
2 3 1
3 1 2
```

Primer 2

```
INTERAKCIJA SA PROGRAMOM:
Unesite broj n: 4
1 2 3 4
2 3 4 1
3 4 1 2
4 1 2 3
```

- (d) Ispisati pravougli „trougao” sačinjen od „koordinata” svojih tačaka. „Koordinata” tačke je oblika  $(i, j)$  pri čemu  $i, j = 0, \dots, n$ . Prav ugao se nalazi u gornjem levom uglu slike i njegova koordinata je  $(0, 0)$ . Koordinata  $i$  se uvećava po vrsti, a koordinata  $j$  po koloni, pa je zato koordinata tačke koja je ispod tačke  $(0, 0)$  jednaka  $(1, 0)$ , a koordinata tačke koja je desno od tačke  $(0, 0)$  jednaka  $(0, 1)$ .

Primer 1

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

Primer 2

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

Primer 3

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

**Zadatak 1.3.48** Napisati program koji za uneti pozitivan ceo broj  $n$  zvezdicama iscrtava odgovarajuću sliku. NAPOMENA: *Pretpostaviti da je unos ispravan.*

- (a) Slika predstavlja kvadrat stranice  $n$  sastavljen od zvezdica.

Primer 1

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

Primer 2

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

- (b) Slika predstavlja rub kvadrata dimenzije  $n$ .

Primer 1

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

Primer 2

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

- (c) Slika predstavlja rub kvadrata dimenzije  $n$  koji i na glavnoj dijagonali ima zvezdice.

Primer 1

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

Primer 1

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

\* **Zadatak 1.3.49** Napisati program koji za uneti pozitivan ceo broj  $n$  zvezdicama iscrtava slovo  $X$  dimenzije  $n$ . NAPOMENA: *Pretpostaviti da je unos ispravan.*

Primer 1

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

Primer 2

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

## 1 Osnovni elementi imperativnog programiranja

---

\* **Zadatak 1.3.50** Napisati program koji za uneti neparan pozitivan broj  $n$  korišćenjem znaka  $+$  iscrtava veliko plus dimenzije  $n$ . NAPOMENA: *Pretpostaviti da je unos ispravan.*

Primer 1

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

Primer 2

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

**Zadatak 1.3.51** Napisati program koji učitava pozitivan ceo broj  $n$ , a potom iscrtava odgovarajuću sliku. NAPOMENA: *Pretpostaviti da je unos ispravan.*

- (a) Slika predstavlja pravougli trougao sastavljen od zvezdica. Kateta trougla je dužine  $n$ , a prav ugao se nalazi u gornjem levom uglu slike.

Primer 1

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

Primer 1

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

- (b) Slika predstavlja pravougli trougao sastavljen od zvezdica. Kateta trougla je dužine  $n$ , a prav ugao se nalazi u donjem levom uglu slike.

Primer 1

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

Primer 2

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

- (c) Slika predstavlja pravougli trougao sastavljen od zvezdica. Kateta trougla je dužine  $n$ , a prav ugao se nalazi u gornjem desnom uglu slike.

Primer 1

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

Primer 1

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

- (d) Slika predstavlja pravougli trougao sastavljen od zvezdica. Kateta trougla je dužine  $n$ , a prav ugao se nalazi u donjem desnom uglu slike.

Primer 1

```

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

```

Primer 2

```

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

```

- (e) Slika predstavlja trougao sastavljen od zvezdica. Trougao se dobija spajanjem dva pravougla trougla kateta dužine  $n$ , pri čemu je prav ugao prvog trougla u njegovom donjem levom uglu, dok je prav ugao drugog trougla u njegovom gornjem levom uglu, a spajanje se vrši po horizontalnoj kateti.

Primer 1

```

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

```

Primer 2

```

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

```

- (f) Slika predstavlja rub jednakokrakog pravouglog trougla čije su katete dužine  $n$ . Program učitava karakter  $c$  i taj karakter koristi za iscrtavanje ruba trougla.

Primer 1

```

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

```

Primer 2

```

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

```

**Zadatak 1.3.52** Napisati program koji učitava pozitivan ceo broj  $n$ , a potom iscrtava odgovarajuću sliku. NAPOMENA: *Pretpostaviti da je unos ispravan.*

- (a) Slika predstavlja jednakostranični trougao stranice  $n$  koji je sastavljen od zvezdica.

Primer 1

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

Primer 2

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

- (b) Slika predstavlja jednakostranični trougao stranice  $n$  koji je sastavljen od zvezdica pri čemu je vrh trougla na dnu slike.

Primer 1

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

Primer 2

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

- (c) Slika predstavlja trougao koji se dobija spajanjem dva jednakostranična trougla stranice  $n$  koji su sastavljeni od zvezdica.

Primer 1

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

Primer 2

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

- (d) Slika predstavlja rub jednakostraničnog trougla čija stranica je dužine  $n$ .

Primer 1

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

Primer 1

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

- (e) Slika se dobija spajanjem dva jednakostranična trougla čija stranica je dužine  $n$ . Iscrtavati samo rub trouglova.



Primer 1

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

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

Primer 2

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

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

\* **Zadatak 1.3.53** Napisati program koji za uneti pozitivan ceo broj  $n$  iscrta strelice dimenzije  $n$ . NAPOMENA: *Pretpostaviti da je unos ispravan.*

Primer 1

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

*
 *
***
 *
*
```

Primer 2

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

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

\* **Zadatak 1.3.54** Napisati program koji učitava pozitivan ceo broj  $n$  i iscrta sliku koja se dobija na sledeći način: u prvom redu je jedna zvezdica, u drugom redu su dve zvezdice razdvojene razmakom, treći red je sastavljen od zvezdica i iste je dužine kao i drugi red, četvrti red se sastoji od tri zvezdice razdvojene razmakom, a peti red je sastavljen od zvezdica i iste je dužine kao i četvrti red itd. Ukupna visina slike je  $n$ . NAPOMENA: *Pretpostaviti da je unos ispravan.*

Primer 1

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

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

\* **Zadatak 1.3.55** Napisati program koji učitava pozitivne cele brojeve  $m$  i  $n$  i iscrtava jedan do drugog  $n$  kvadrata čija je svaka stranica sastavljena od  $m$  zvezdica razdvojenih prazninom. NAPOMENA: *Pretpostaviti da je unos ispravan.*

### Primer 1

```
INTERAKCIJA SA PROGRAMOM:
Unesite brojeve n i m: 5 3
*****
*           *           *
*         *           *
*       *           *
*     *           *
*   *           *
* * * * *
*****
```

Primer 2

```

INTERAKCIJA SA PROGRAMOM:
  Unesite brojeve n i m: 4 4
  * * * * *
  *       *       *       *       *
  *       *       *       *       *
  * * * * *

```

\* **Zadatak 1.3.56** Napisati program koji učitava pozitivan ceo broj  $n$  i iscrtava romb sastavljen od minusa u pravougaoniku sastavljenom od zvezdica. NAPOMENA: *Pretpostaviti da je unos ispravan.*

### Primer 1

```
INTERAKCIJA SA PROGRAMOM:  
Unesite broj n: 6  
*****  
*****-----*****  
***------***  
**-----**  
*------*  
  
**-----**  
***-----***  
****-----****  
*****-----*****  
*****
```

## Primer 2

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

**Zadatak 1.3.57** Napisati program koji učitava ceo broj  $n$  ( $n \geq 2$ ) i koji iscrtava sliku kuće sa krovom: kuća je kvadrat stranice  $n$ , a krov jednakostranični trougao stranice  $n$ . NAPOMENA: *Pretpostaviti da je unos ispravan.*

## Primer 1

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

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

## Primer 2

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

\* **Zadatak 1.3.58** Napisati program koji učitava pozitivan ceo broj  $n$  i ispisuje brojeve od 1 do  $n$ , zatim od 2 do  $n - 1$ , 3 do  $n - 2$ , itd. Ispis se završava kada nije moguće ispisati ni jedan broj. NAPOMENA: *Pretpostaviti da je unos*

ispravan.

#### Primer 1

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite broj n: 5
|| 1 2 3 4 5 2 3 4 3
```

#### Primer 2

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite broj n: 6
|| 1 2 3 4 5 6 2 3 4 5 3 4
```

#### Primer 3

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite broj n: 7
|| 1 2 3 4 5 6 7 2 3 4 5 6 3 4 5 4
```

#### Primer 4

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite broj n: 3
|| 1 2 3 2
```

\* **Zadatak 1.3.59** Napisati program koji učitava pozitivan ceo broj  $n$  i ispisuje izabrane brojeve u  $n$  redova. U prvom redu, program ispisuje sve brojeve iz intervala  $[1, n]$ . U drugom redu, program ispisuje svaki drugi broj iz ovog intervala. U trećem redu, program ispisuje svaki treći broj iz ovog intervala, i tako redom. Na kraju, u  $n$ -tom redu, program ispisuje samo broj 1. NAPOMENA: *Pretpostaviti da je unos ispravan.*

#### Primer 1

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite broj n: 3
|| 1 2 3
|| 1 3
|| 1
```

#### Primer 2

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite broj n: 7
|| 1 2 3 4 5 6 7
|| 1 3 5 7
|| 1 4 7
|| 1 5
|| 1 6
|| 1 7
|| 1
```

#### Primer 3

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite broj n: 1
|| 1
```

### 1.4 Funkcije

**Zadatak 1.4.1** Napisati funkciju `int min(int x, int y, int z)` koja izračunava minimum tri broja. Napisati program koji učitava tri cela broja i ispisuje njihov minimum.

*Primer 1*

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

*Primer 2*

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

**Zadatak 1.4.2** Napisati funkciju `float razlomljeni_deo(float x)` koja izračunava razlomljeni deo broja  $x$ . Napisati program koji učitava jedan realan broj i ispisuje njegov razlomljeni deo na šest decimala.

*Primer 1*

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

*Primer 2*

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

**Zadatak 1.4.3** Napisati funkciju `int zbir_delilaca(int n)` koja izračunava zbir delilaca broja  $n$ . Napisati program koji učitava ceo pozitivan broj  $k$  i ispisuje zbir delilaca svakog broja od 1 do  $k$ . U slučaju neispravnog unosa, ispisati odgovarajuću poruku o grešci.

*Primer 1*

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

*Primer 2*

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesite broj k: -2  
|| Greska: neispravan unos.
```

**Zadatak 1.4.4** Napisati funkciju `int je_stepen(unsigned x, unsigned n)` koja za dva broja  $x$  i  $n$  utvrđuje da li je  $x$  neki stepen broja  $n$ . Ukoliko jeste, funkcija vraća izložilac stepena, a u suprotnom vraća  $-1$ . Napisati program koji učitava dva neoznačena broja i ispisuje da li vrednost prvog broja odgovara vrednosti nekog stepena drugog broja (i kog).

*Primer 1*

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesite dva broja: 81 3  
|| Jeste: 81 = 3^4
```

*Primer 2*

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesite dva broja: 162 11  
|| Broj 162 nije stepen broja 11.
```

**Zadatak 1.4.5** Napisati funkciju `int euklid(int x, int y)` koja za dva data cela broja određuje najveći zajednički delilac primenom Euklidovog algoritma. Napisati program koji učitava dva cela broja i ispisuje vrednost njihovog najvećeg zajedničkog delioca.

*Primer 1*

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite dva cela broja: 1024 832
|| Najveci zajednicki delilac: 64
```

*Primer 2*

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite dva cela broja: -900 112
|| Najveci zajednicki delilac: 4
```

**Zadatak 1.4.6** Napisati funkciju `float zbir_reciprocnih(int n)` koja za dato  $n$  vraća zbir recipročnih vrednosti brojeva od 1 do  $n$ . Napisati program koji učitava ceo pozitivan broj  $n$  i ispisuje odgovarajući zbir zaokružen na dve decimale. U slučaju neispravnog unosa, ispisati odgovarajuću poruku o grešci.

*Primer 1*

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite broj n: 10
|| Zbir reciprocnih: 2.93
```

*Primer 2*

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite broj n: 100
|| Zbir reciprocnih: 5.19
```

*Primer 3*

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite broj n: -100
|| Greska: neispravan unos.
```

**Zadatak 1.4.7** Napisati funkciju `int prebrojavanje(float x)` koja prebrojava koliko puta se broj  $x$  pojavljuje u nizu brojeva koji se unose sve do unosa broja nula. Napisati program koji učitava vrednost broja  $x$  i ispisuje koliko puta se njegova vrednost pojavila u unetom nizu.

*Primer 1*

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

*Primer 2*

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

**Zadatak 1.4.8** Broj je prost ako je deljiv samo sa 1 i samim sobom.

- Napisati funkciju `int prost(int x)` koja ispituje da li je dati ceo broj prost. Funkcija treba da vrati jedinicu ako je broj prost ili nulu u suprotnom.
- Napisati funkciju `void prvih_n_prostih(int n)` koja ispisuje prvih  $n$  prostih brojeva.
- Napisati funkciju `void prosti_brojevi_manji_od_n(int n)` koja ispisuje sve proste brojeve manje od broja  $n$ .

## 1 Osnovni elementi imperativnog programiranja

---

Napisati program koji učitava pozitivan ceo broj  $n$  i ispisuje prvih  $n$  prostih brojeva, kao i sve proste brojeve manje od  $n$ . U slučaju neispravnog unosa, ispisati odgovarajuću poruku o grešci.

### Primer 1

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite broj n: 5
|| Prvih n prostih: 2 3 5 7 11
|| Prosti manji od n: 2 3
```

### Primer 2

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite broj n: 2
|| Prvih n prostih: 2 3
|| Prosti manji od n: ne postoje
```

### Primer 3

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite broj n: -11
|| Greska: neispravan unos.
```

**Zadatak 1.4.9** Rešiti sledeće zadatke korišćenjem funkcija.

- (a) Zadatak 1.1.2 rešiti korišćenjem funkcija `int kvadrat(int x)` koja računa kvadrat datog broja i `int kub(int x)` koja računa kub datog broja.
- (b) Zadatak 1.2.2 rešiti korišćenjem funkcije `float apsolutna_vrednost(float x)` koja izračunava apsolutnu vrednost datog broja.
- (c) Zadatak 1.3.7 rešiti korišćenjem funkcije `float stepen(float x, int n)` koja računa vrednost  $n$ -tog stepena realnog broja  $x$ .
- (d) Zadatak 1.3.28 rešiti korišćenjem funkcije `int fibonaci(int n)` koja računa  $n$ -ti element Fibonačijevog niza.

**Zadatak 1.4.10** Napisati funkciju `float aritmeticka_sredina(int n)` koja računa aritmetičku sredinu cifara datog broja. Napisati i program koji učitava ceo broj i ispisuje aritmetičku sredinu njegovih cifara zaokruženu na tri decimale.

### Primer 1

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

### Primer 2

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

### Primer 3

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

**Zadatak 1.4.11** Napisati funkciju `int sadrzi(int x, int c)` koja ispituje da li se cifra  $c$  nalazi u zapisu celog broja  $x$ . Funkcija treba da vrati jedinicu ako se cifra nalazi u broju, a nulu inače. Napisati program koji učitava jedan ceo broj i jednu cifru i u zavisnosti od toga da li se uneta cifra nalazi u zapisu unetog broja, ispisuje odgovarajuću poruku. U slučaju neispravnog unosa, ispisati odgovarajuću poruku o grešci.

*Primer 1*

```

|| INTERAKCIJA SA PROGRAMOM:
|| Unesite broj i cifru: 17890 7
|| Cifra 7 se nalazi u zapisu broja 17890.

```

*Primer 2*

```

|| INTERAKCIJA SA PROGRAMOM:
|| Unesite broj i cifru: 19 6
|| Cifra 6 se ne nalazi u zapisu broja 19.

```

*Primer 3*

```

|| INTERAKCIJA SA PROGRAMOM:
|| Unesite broj i cifru: 17890 26
|| Greska: neispravan unos.

```

*Primer 4*

```

|| INTERAKCIJA SA PROGRAMOM:
|| Unesite broj i cifru: -1982 9
|| Cifra 9 se nalazi u zapisu broja -1982.

```

**Zadatak 1.4.12** Napisati funkciju `int broj_neparnih_cifara(int x)` koja određuje broj neparnih cifara u zapisu datog celog broja. Napisati program koji učitava cele brojeve sve do unosa broja nula i ispisuje broj neparnih cifara svakog unetog broja.

*Primer 1*

```

|| INTERAKCIJA SA PROGRAMOM:
|| Unesite cele brojeve:
|| 2341
|| Broj neparnih cifara: 2
|| 78
|| Broj neparnih cifara: 1
|| 800
|| Broj neparnih cifara: 0
|| -99761
|| Broj neparnih cifara: 4
|| 0

```

*Primer 2*

```

|| INTERAKCIJA SA PROGRAMOM:
|| Unesite cele brojeve:
|| 987611
|| Broj neparnih cifara: 4
|| 135
|| Broj neparnih cifara: 3
|| -701
|| Broj neparnih cifara: 2
|| 602
|| Broj neparnih cifara: 0
|| -884
|| Broj neparnih cifara: 0
|| 79901
|| Broj neparnih cifara: 4
|| 0

```

**Zadatak 1.4.13** Napisati program za ispitivanje svojstava cifara datog celog broja.

- Napisati funkciju `int sve_parne_cifre(int x)` koja ispituje da li se dati ceo broj sastoji isključivo iz parnih cifara. Funkcija treba da vrati jedinicu ako su sve cifre broja parne, a nulu inače.
- Napisati funkciju `int sve_cifre_jednake(int x)` koja ispituje da li su sve cifre datog celog broja jednake. Funkcija treba da vrati jedinicu ako su sve cifre broja jednake, a nulu inače.

Program učitava ceo broj `i` u zavisnosti od toga da li su navedena svojstva ispunjena ili ne, ispisuje odgovarajuću poruku.

## 1 Osnovni elementi imperativnog programiranja

---

### Primer 1

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite broj: 86422
|| Sve cifre broja su parne.
|| Cifre broja nisu jednake.
```

### Primer 2

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite broj: 55555
|| Broj sadrži bar jednu neparnu cifru.
|| Cifre broja su jednake.
```

### Primer 3

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite broj: -88
|| Sve cifre broja su parne.
|| Cifre broja su jednake.
```

### Primer 4

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite broj i cifru: -342
|| Broj sadrži bar jednu neparnu cifru.
|| Cifre broja nisu jednake.
```

**Zadatak 1.4.14** Napisati funkciju `int ukloni(int n, int p)` koja menja broj  $n$  tako što iz njegovog zapisa uklanja cifru na poziciji  $p$ . Pozicije se broje sa desna na levo. Cifra jedinica ima poziciju 1. Napisati program koji učitava redni broj pozicije i zatim za cele brojeve koji se unose sve do unosa broja nula, ispisuje brojeve kojima je uklonjena cifra na poziciji  $p$ . U slučaju neispravnog unosa, ispisati odgovarajuću poruku o grešci.

### Primer 1

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite poziciju: 3
|| Unesite broj: 1210
|| Novi broj: 110
|| Unesite broj: 18
|| Novi broj: 18
|| Unesite broj: 3856
|| Novi broj: 356
|| Unesite broj: 0
```

### Primer 2

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite poziciju: 1
|| Unesite broj: -9632
|| Novi broj: -963
|| Unesite broj: -2
|| Novi broj: 0
|| Unesite broj: 246
|| Novi broj: 24
|| Unesite broj: -52
|| Novi broj: -5
|| Unesite broj: 0
```

### Primer 3

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite poziciju: 0
|| Greska: neispravan unos.
```

**Zadatak 1.4.15** Napisati funkciju `int zapis(int x, int y)` koja proverava da li se brojevi  $x$  i  $y$  zapisuju pomoću istih cifara. Funkcija treba da vrati jedinicu ako je uslov ispunjen, a nulu inače. Napisati program koji učitava dva cela broja i ispisuje da li je za njih pomenuti uslov ispunjen ili ne.

### Primer 1

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

### Primer 2

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

### Primer 3

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

### Primer 4

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite dva broja: -777 77
|| Uslov nije ispunjen.
```



**Zadatak 1.4.16** Napisati funkciju `int neopadajuce(int n)` koja ispituje da li su cifre datog celog broja u neopadajućem poretку. Funkcija treba da vrati jedinicu ako cifre ispunjavaju uslov, a nulu inače. Napisati program koji učitava ceo broj i ispisuje poruku da li su cifre unetog broja u neopadajućem poretку.

*Primer 1*

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite broj: 2289
|| Cifre su u neopadajućem poretку.
```

*Primer 2*

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite broj: 5
|| Cifre su u neopadajućem poretку.
```

*Primer 3*

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite broj: 6628
|| Cifre nisu u neopadajućem poretку.
```

*Primer 4*

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite broj: -23
|| Cifre su u neopadajućem poretку.
```

**Zadatak 1.4.17** Napisati funkciju `int par_nepar(int n)` koja ispituje da li su cifre datog celog broja naizmenično parne i neparne. Funkcija treba da vrati jedinicu ako cifre ispunjavaju uslov, a nulu inače. Napisati program koji učitava ceo broj i ispisuje da li on ispunjava pomenuti uslov ili ne.

*Primer 1*

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

*Primer 2*

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

*Primer 3*

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

*Primer 4*

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite broj n: 5
|| Broj ispunjava uslov.
```

**Zadatak 1.4.18** Napisati funkciju `int rotacija(int n)` koja rotira cifre zadanog celog broja za jednu poziciju u levo. Napisati program koji za brojeve koji se unose sve do unosa broja nula ispisuje odgovarajuće rotirane brojeve.

*Primer 1*

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

*Primer 2*

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

**Zadatak 1.4.19** Za dati broj može se formirati niz tako da je svaki sledeći član niza dobijen kao suma cifara prethodnog člana niza. Broj je *srećan* ako

## 1 Osnovni elementi imperativnog programiranja

---

se dati niz završava jedinicom. Napisati funkciju `int srecan(int x)` koja vraća jedinicu ako je broj srećan, a nulu inače. Napisati program koji za uneti pozitivan ceo broj  $n$  ispisuje sve srećne brojeve od 1 do  $n$ . U slučaju neispravnog unosa, ispisati odgovarajuću poruku o grešci.

### Primer 1

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite broj n: 100
|| Srećni brojevi:
|| 1 10 19 28 37 46 55 64 73 82 91 100
```

### Primer 2

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

**Zadatak 1.4.20** Prirodan broj  $a$  je Armstrongov ako je jednak sumi  $n$ -tih stepena svojih cifara, pri čemu je  $n$  broj cifara broja  $a$ . Napisati funkciju `int armstrong(int x)` koja vraća jedinicu ako je broj Armstrongov, a nulu inače. Napisati program koji za učitani pozitivan ceo broj proverava da li je Armstrongov. U slučaju neispravnog unosa, ispisati odgovarajuću poruku o grešci.

### Primer 1

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

### Primer 2

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

### Primer 3

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite broj: 0
|| Greska: neispravan unos.
```

**Zadatak 1.4.21** Napisati funkciju `double e_na_x(double x, double eps)` koja računa vrednost  $e^x$  kao parcijalnu sumu reda  $\sum_{n=0}^{\infty} \frac{x^n}{n!}$ , pri čemu se sumiranje sprovodi sve dok je sabirak po apsolutnoj vrednosti veći od date tačnosti  $\text{eps}$ . Napisati program koji učitava dva realna broja  $x$  i  $\text{eps}$  i ispisuje izračunatu vrednost  $e^x$ .

### Primer 1

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite broj x: 5
|| Unesite eps: 0.001
|| Rezultat: 148.412951
```

### Primer 2

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite broj x: -3
|| Unesite eps: 0.0001
|| Rezultat: 0.049796
```

**Zadatak 1.4.22** Napisati funkciju `void ispis(float x, float y, int n)` koja za dva realna broja  $x$  i  $y$  i jedan pozitivan ceo broj  $n$  ispisuje vrednosti sinusne funkcije u  $n$  ravnomerno raspoređenih tačaka intervala  $[x, y]$ . Napisati program koji učitava granice intervala i broj tačaka i ispisuje odgovarajuće vrednosti sinusne funkcije zaokružene na četiri decimale. U slučaju neispravnog unosa,

ispisati odgovarajuću poruku o grešci.

#### Primer 1

```
INTERAKCIJA SA PROGRAMOM:
Unesite dva realna broja: 7 31
Unesite broj n: 6
Rezultat:
sin(7.0000) = 0.6570
sin(11.8000) = -0.6935
sin(16.6000) = -0.7784
sin(21.4000) = 0.5573
sin(26.2000) = 0.8759
sin(31.0000) = -0.4040
```

#### Primer 2

```
INTERAKCIJA SA PROGRAMOM:
Unesite dva realna broja: -8.32 20.5
Unesite broj n: 5
Rezultat:
sin(-8.3200) = -0.8934
sin(-1.1150) = -0.8979
sin(6.0900) = -0.1920
sin(13.2950) = 0.6658
sin(20.5000) = 0.9968
```

#### Primer 3

```
INTERAKCIJA SA PROGRAMOM:
Unesite dva realna broja: 8 8
Greska: neispravan unos.
```

#### Primer 1

```
INTERAKCIJA SA PROGRAMOM:
Unesite dva realna broja: 7 32
Unesite broj n: -10
Greska: neispravan unos.
```

**Zadatak 1.4.23** Napisati funkciju `char sifra(char c, int k)` koja za dati karakter  $c$  određuje šifru na sledeći način: ukoliko je  $c$  slovo, šifra je slovo koje se nalazi  $k$  pozicija pre njega u engleskoj abecedi. Karakteri koji nisu slova se ne šifruju. Šifrovanje treba da bude kružno, što znači da je, na primer, šifra za karakter  $b$  i pomeraj 2 karakter  $z$ . Napisati program koji učitava nenegativan ceo broj  $k$ , a zatim i karaktere sve do kraja ulaza i nakon svakog učitanoog karaktera ispisuje njegovu šifru. U slučaju neispravnog unosa, ispisati odgovarajuću poruku o grešci.

#### Primer 1

```
INTERAKCIJA SA PROGRAMOM:
Unesite broj k: 2
Unesite tekst (CTRL+D za prekid):
U svetu postoji jedno carstvo
S qtcrs nmqrmhg hcbml aypqrtm
U njemu caruje drugarstvo.
S lhcks aypshc bpseyqrtm.
```

#### Primer 2

```
INTERAKCIJA SA PROGRAMOM:
Unesite broj k: -2
Greska: neispravan unos.
```

**Zadatak 1.4.24** Rešiti sledeće zadatke korišćenjem funkcija.

- Zadatak 1.3.31 rešiti korišćenjem funkcije `char konverzija(char c)` koja malo slovo pretvara u odgovarajuće veliko i obrnuto.
- Zadatak 1.3.32 rešiti korišćenjem funkcije `void prebrojavanje()` koja učitava karaktere sve do kraja ulaza i ispisuje broj malih slova, velikih slova, cifara, belina, kao i sumu svih unetih cifara.

**Zadatak 1.4.25** Napisati program koji učitava tri cela broja koja predstavljaju dan, mesec i godinu i ispisuje datum sledećeg dana. Zadatak rešiti korišćenjem narednih funkcija.

- (a) `int prestupna(int godina)` koja za zadatu godinu proverava da li je prestupna. Funkcija treba da vrati jedinicu ako je godina prestupna ili nulu ako nije.
- (b) `int broj_dana(int mesec, int godina)` koja za dati mesec i godinu vraća broj dana u datom mesecu.
- (c) `int ispravan(int dan, int mesec, int godina)` koja za dati datum proverava da li je ispravan.
- (d) `void sledeci_dan(int dan, int mesec, int godina)` koja za dati datum ispisuje datum sledećeg dana.

U slučaju neispravnog unosa, ispisati odgovarajuću poruku o grešci.

### Primer 1

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesite datum: 24.8.1998.  
|| Datum sledeceg dana je: 25.8.1998.
```

### Primer 2

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesite datum: 31.12.1789.  
|| Datum sledeceg dana je: 1.1.1790.
```

### Primer 3

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesite datum: 28.2.2003.  
|| Datum sledeceg dana je: 1.3.2004.
```

### Primer 4

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesite datum: 31.4.2004.  
|| Greska: neispravan unos.
```

**Zadatak 1.4.26** Napisati funkciju `int od_nove_godine(int dan, int mesec, int godina)` koja određuje koliko je dana proteklo od Nove godine do datog datuma. Napisati program koji učitava tri cela broja koja predstavljaju dan, mesec i godinu i ispisuje koliko dana je proteklo od Nove godine. U slučaju neispravnog unosa, ispisati odgovarajuću poruku o grešci.

### Primer 1

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesite datum: 24.8.1998.  
|| Broj dana od Nove godine je: 235
```

### Primer 2

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesite datum: 31.12.1680.  
|| Broj dana od Nove godine je: 366
```

### Primer 3

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesite datum: 28.2.2003.  
|| Broj dana od Nove godine je: 58
```

### Primer 4

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesite datum: 31.4.2004.  
|| Greska: neispravan unos.
```

**Zadatak 1.4.27** Napisati funkciju `int do_kraja_godine(int dan, int mesec, int godina)` koja određuje broj dana od datog datuma do kraja godine. Napisati program koji učitava tri cela broja koja predstavljaju dan, mesec i godinu i ispisuje broj dana do kraja godine. U slučaju neispravnog unosa, ispisati odgovarajuću poruku o grešci.

*Primer 1*

```
INTERAKCIJA SA PROGRAMOM:
Unesite datum: 24.8.1998.
Broj dana do Nove godine je: 129
```

*Primer 2*

```
INTERAKCIJA SA PROGRAMOM:
Unesite datum: 31.12.1680.
Broj dana do Nove godine je: 0
```

*Primer 3*

```
INTERAKCIJA SA PROGRAMOM:
Unesite datum: 28.2.2004.
Broj dana do Nove godine je: 307
```

*Primer 4*

```
INTERAKCIJA SA PROGRAMOM:
Unesite datum: 31.4.2004.
Greska: neispravan unos.
```

**Zadatak 1.4.28** Napisati funkciju `int broj_dana_izmedju(int dan1, int mesec1, int godina1, int dan2, int mesec2, int godina2)` koja određuje broj dana između dva datuma. Napisati program koji učitava dva datuma u formatu `dd.mm.gggg` i ispisuje broj dana između ta dva datuma. U slučaju neispravnog unosa, ispisati odgovarajuću poruku o grešci.

*Primer 1*

```
INTERAKCIJA SA PROGRAMOM:
Unesite prvi datum: 12.3.2008.
Unesite drugi datum: 5.12.2008.
Broj dana izmedju dva datuma je: 268
```

*Primer 2*

```
INTERAKCIJA SA PROGRAMOM:
Unesite prvi datum: 26.9.1986.
Unesite drugi datum: 2.2.1701.
Broj dana izmedju dva datuma je: 104301
```

*Primer 3*

```
INTERAKCIJA SA PROGRAMOM:
Unesite prvi datum: 24.8.1998.
Unesite drugi datum: 12.10.2010.
Broj dana izmedju dva datuma je: 4440
```

*Primer 4*

```
INTERAKCIJA SA PROGRAMOM:
Unesite prvi datum: 24.8.1998.
Unesite drugi datum: 31.4.2004.
Greska: neispravan unos.
```

**Zadatak 1.4.29** Napisati funkciju `void romb(int n)` koja iscrtava romb čija je stranica dužine  $n$ . Napisati program koji učitava ceo pozitivan broj  $i$  i iscrtava odgovarajuću sliku. U slučaju neispravnog unosa, ispisati odgovarajuću poruku o grešci.

Primer 1

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

Primer 2

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

Primer 3

```
INTERAKCIJA SA PROGRAMOM:
Unesite broj n: -5
Greska: neispravan unos.
```

**Zadatak 1.4.30** Napisati funkciju `void grafikon_h(int a, int b, int c, int d)` koja iscrtava horizontalni prikaz zadatih vrednosti. Napisati program koji učitava četiri pozitivna cela broja i iscrtava odgovarajuću sliku. U slučaju neispravnog unosa, ispisati odgovarajuću poruku o grešci.

Primer 1

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

Primer 2

```
INTERAKCIJA SA PROGRAMOM:
Unesite brojeve: 5 2 2 10
*****
**
**
*****
```

Primer 3

```
INTERAKCIJA SA PROGRAMOM:
Unesite brojeve: 8 -2 5 4
Greska: neispravan unos.
```

**Zadatak 1.4.31** Napisati funkciju `void grafikon_v(int a, int b, int c, int d)` koja iscrtava vertikalni prikaz zadatih vrednosti. Napisati program koji učitava četiri pozitivna cela broja i iscrtava odgovarajuću sliku. U slučaju neispravnog unosa, ispisati odgovarajuću poruku o grešci.

Primer 1

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

Primer 2

```
INTERAKCIJA SA PROGRAMOM:
Unesite brojeve: 5 2 2 4
*
* *
* *
****
****
```

Primer 3

```
INTERAKCIJA SA PROGRAMOM:
Unesite brojeve: 8 -2 5 4
Greska: neispravan unos.
```

## 2

# Napredni tipovi podataka

## 2.1 Nizovi

**Zadatak 2.1.1** Napisati program koji učitava dimenziju niza, elemente niza i zatim ispisuje:

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

Maksimalni broj elemenata niza je 100. U slučaju neispravnog unosa, ispisati odgovarajuću poruku o grešci.

### *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: neispravan unos.
```

**Zadatak 2.1.2** Napisati program koji učitava dimenziju niza, elemente niza i zatim menja uneti niz tako što kvadrira sve negativne elemente niza. Maksimalni

## 2 Napredni tipovi podataka

broj elemenata niza je 100. U slučaju neispravnog unosa, ispisati odgovarajuću poruku o grešci.

### Primer 1

```
INTERAKCIJA SA PROGRAMOM:
Unesite dimenziju niza: 6
Unesite elemente niza:
12.34 -6 1 8 32.4 -16
Rezultujući niz:
12.34 36 1 8 32.4 256
```

### Primer 2

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

### Primer 3

```
INTERAKCIJA SA PROGRAMOM:
Unesite dimenziju niza: 4
Unesite elemente niza:
9.53 5 1 4.89
Rezultujući niz:
9.53 5 1 4.89
```

### Primer 4

```
INTERAKCIJA SA PROGRAMOM:
Unesite dimenziju niza: 104
Greska: neispravan unos.
```

**Zadatak 2.1.3** Ako su  $a = (a_1, \dots, a_n)$  i  $b = (b_1, \dots, b_n)$  vektori dimenzije  $n$ , njihov skalarni proizvod se definiše kao  $a \cdot b = a_1 \cdot b_1 + \dots + a_n \cdot b_n$ . 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. U slučaju neispravnog unosa, ispisati odgovarajuću poruku o grešci.

### 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: 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: 0
```

### Primer 3

```
INTERAKCIJA SA PROGRAMOM:
Unesite dimenziju vektora: 0
Greska: neispravan unos.
```

### Primer 4

```
INTERAKCIJA SA PROGRAMOM:
Unesite dimenziju vektora: 1
Unesite koordinate vektora a:
-1
Unesite koordinate vektora b:
1
Skalarni proizvod: -1
```

**Zadatak 2.1.4** Napisati program koji učitava dimenziju niza, elemente niza, a potom i ceo broj  $k$  i ispisuje indekse elemenata koji su deljivi sa  $k$ . Maksimalni broj elemenata niza je 100. U slučaju neispravnog unosa, ispisati odgovarajuću poruku o grešci.



## Primer 1

```

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

```

## Primer 4

```

INTERAKCIJA SA PROGRAMOM:
Unesite dimenziju niza: 6
Unesite elemente niza: 1 2 3 4 5 6
Unesite broj k: 0
Greska: neispravan unos.

```

**Zadatak 2.1.5** 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. Napisati program koji učitava broj autobusa  $n$ ,  $n$  celih brojeva koji označavaju vreme putovanja tih autobusa i vrednosti  $k$ ,  $t$  i  $m$  i ispisuje vreme putovanja svih autobusa nakon unetih izmena. Maksimalni broj autobusa je 200. U slučaju neispravnog unosa, ispisati odgovarajuću poruku o grešci.

## Primer 1

```

INTERAKCIJA SA PROGRAMOM:
Unesite broj autobusa: 8
Unesite vreme putovanja:
24 78 13 124 56 90 205 45
Unesite vrednosti k, t i m:
3 6 23
Vreme putovanja nakon izmena:
24 78 36 147 79 113 205 45

```

## Primer 2

```

INTERAKCIJA SA PROGRAMOM:
Unesite broj autobusa: 8
Unesite vreme putovanja:
24 78 13 124 56 90 205 45
Unesite vrednosti k, t i m:
3 15 3
Greska: neispravan unos.

```

**Zadatak 2.1.6** 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

```

## 2 Napredni tipovi podataka

### 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
```

**Zadatak 2.1.7** Napisati program koji učitava karaktere sve do unosa karaktera \*, a zatim ih ispisuje u redosledu suprotnom od redosleda čitanja. Maksimalni broj karaktera je 500.

### 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
```

**Zadatak 2.1.8** Napisati program koji učitava karaktere sve do kraja ulaza, a potom 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:
Unesite tekst:
Mis je dobio grip.
Karakter b se pojavljuje 1 puta
Karakter d se pojavljuje 1 puta
Karakter e se pojavljuje 1 puta
Karakter g se pojavljuje 1 puta
Karakter i se pojavljuje 3 puta
Karakter j se pojavljuje 1 puta
Karakter o se pojavljuje 2 puta
Karakter p se pojavljuje 1 puta
Karakter r se pojavljuje 1 puta
Karakter s se pojavljuje 1 puta
Karakter M se pojavljuje 1 puta
```

### Primer 2

```
INTERAKCIJA SA PROGRAMOM:
Unesite tekst:
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
```

**Zadatak 2.1.9** Napisati program koji učitava jednu liniju teksta i ispisuje koliko puta se pojavilo svako od slova engleske abecede 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
```

**Zadatak 2.1.10** Takmičari na Beogradskom maratonu su označeni rednim brojevima počevši od 0. Vremena za koja su takmičari istrčali maraton izražena u minutima se zadaju nizom celih brojeva u kojem indeks elementa niza označava redni broj takmičara. Napisati sledeće funkcije za obradu navedenih podataka:

- `void ucitaj(int a[], int n)` koja učitava elemente niza  $a$  dimenzije  $n$ .
- `void ispisi(int a[], int n)` koja ispisuje elemente niza  $a$  dimenzije  $n$ .
- `int suma(int a[], int n)` koja računa i vraća ukupno vreme trčanja svih takmičara.
- `float prosek(int a[], int n)` koja računa i vraća prosečno vreme (aritmetičku sredinu) trčanja takmičara.
- `int maksimum(int a[], int n)` koja izračunava i vraća najduže vreme trčanja takmičara.
- `int pozicija_minimum(int a[], int n)` koja vraća redni broj pobednika Beogradskog maratona, tj. onog takmičara koji je najkraće trčao. U slučaju da ima više takvih takmičara, vratiti onog sa najmanjim rednim brojem.

Napisati program koji učitava podatke o rezultatima takmičara na maratonu i ispisuje učitane podatke, ukupno, prosečno i maksimalno vreme trčanja, kao i redni broj pobednika maratona. Maksimalni broj takmičara je 1000. U slučaju

## 2 Napredni tipovi podataka

---

neispravnog unosa, ispisati odgovarajuću poruku o grešci.

### Primer 1

```
INTERAKCIJA SA PROGRAMOM:
Unesite dimenziju niza:
5
Unesite elemente niza: 140 126 170 220 130
Vreme trcanja takmicara: 140 126 170 220 130
Ukupno vreme: 786
Prosecno vreme trcanja: 157.20
Maksimalno vreme trcanja: 220
Indeks pobednika: 1
```

### Primer 2

```
INTERAKCIJA SA PROGRAMOM:
Unesite dimenziju niza:
-5
Greska: neispravan unos.
```

**Zadatak 2.1.11** Napisati funkciju koja izračunava broj elemenata celobrojnog niza koji su manji od poslednjeg elementa niza. Napisati program koji učitava dimenziju niza i elemente niza, a zatim ispisuje broj elemenata koji zadovoljavaju pomenuti uslov. Maksimalni broj elemenata niza je 100. U slučaju neispravnog unosa, ispisati odgovarajuću poruku o grešci.

### Primer 1

```
INTERAKCIJA SA PROGRAMOM:
Unesite dimenziju niza: 4
Unesite elemente niza: 11 2 4 9
Rezultat: 2
```

### Primer 2

```
INTERAKCIJA SA PROGRAMOM:
Unesite dimenziju niza: 7
Unesite elemente niza: 7 2 1 14 65 2 8
Rezultat: 4
```

### Primer 3

```
INTERAKCIJA SA PROGRAMOM:
Unesite dimenziju niza: 5
Unesite elemente niza: 25 18 29 30 14
Rezultat: 0
```

### Primer 4

```
INTERAKCIJA SA PROGRAMOM:
Unesite dimenziju niza: -45
Greska: neispravan unos.
```

**Zadatak 2.1.12** Napisati funkciju koja izračunava broj parnih elemenata celobrojnog niza koji prethode maksimalnom elementu niza. Napisati program koji učitava dimenziju niza i elemente niza, a zatim ispisuje broj elemenata koji prethode maksimalnom elementu. Maksimalni broj elemenata niza je 100. U slučaju neispravnog unosa, ispisati odgovarajuću poruku o grešci.

### Primer 1

```
INTERAKCIJA SA PROGRAMOM:
Unesite dimenziju niza: 4
Unesite elemente niza: 11 2 4 9
Rezultat: 0
```

### Primer 2

```
INTERAKCIJA SA PROGRAMOM:
Unesite dimenziju niza: 7
Unesite elemente niza: 7 2 1 14 65 2 8
Rezultat: 2
```

## Primer 3

```

INTERAKCIJA SA PROGRAMOM:
Unesite dimenziju niza: 5
Unesite elemente niza: 25 18 29 30 14
Rezultat: 1

```

## Primer 4

```

INTERAKCIJA SA PROGRAMOM:
Unesite dimenziju niza: 105
Greska: neispravan unos.

```

**Zadatak 2.1.13** 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 program koji učitava dimenziju niza, elemente niza i vrednosti  $i$  i  $j$  i zatim ispisuje zbir u datom opsegu. Maksimalni broj elemenata niza je 100. U slučaju neispravnog unosa, ispisati odgovarajuću poruku o grešci.

## Primer 1

```

INTERAKCIJA SA PROGRAMOM:
Unesite dimenziju 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 dimenziju niza: 3
Unesite elemente niza: -2 8 1
Unesite vrednosti za i i j: 1 12
Greska: neispravan unos.

```

## Primer 3

```

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

```

## Primer 4

```

INTERAKCIJA SA PROGRAMOM:
Unesite dimenziju niza: 4
Unesite elemente niza: 9 5 7 6
Unesite vrednosti za i i j: 2 2
Zbir je: 7

```

**Zadatak 2.1.14** Napisati funkciju `float zbir_pozitivnih(float a[], int n, int k)` koja izračunava zbir prvih  $k$  pozitivnih elemenata realnog niza  $a$  dužine  $n$ . Napisati program koji učitava dimenziju niza, elemente niza i broj  $k$ , a zatim ispisuje zbir prvih  $k$  pozitivnih elemenata niza. Maksimalni broj elemenata niza je 100. U slučaju neispravnog unosa, ispisati odgovarajuću poruku o grešci.

## Primer 1

```

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

```

## Primer 2

```

INTERAKCIJA SA PROGRAMOM:
Unesite dimenziju niza: 3
Unesite elemente niza:
-6.598 -8.14 -15
Unesite vrednost k: 4
Zbir je: 0.00

```

### Primer 3

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

### Primer 4

```
INTERAKCIJA SA PROGRAMOM:
Unesite dimenziju niza: 3
Unesite elemente niza:
-0.11 5.29 -4.17
Unesite vrednost k: -15
Greska: neispravan unos.
```

**Zadatak 2.1.15** Napisati funkciju koja menja niz tako što razmenjuje mesta najmanjem i najvećem elementu niza. Ukoliko se neki od ovih elemenata javlja više puta, uzeti u obzir prvo pojavljivanje. Napisati program koji učitava dimenziju niza, elemente niza, a zatim ispisuje izmenjeni niz. Maksimalni broj elemenata niza je 100. U slučaju neispravnog unosa, ispisati odgovarajuću poruku o grešci.

### Primer 1

```
INTERAKCIJA SA PROGRAMOM:
Unesite dimenziju niza: 5
Unesite elemente niza: 8 -2 11 19 4
Rezultujući niz:
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
Rezultujući niz:
46 -2 51 8 66 -5 2 8 3 14
```

### Primer 3

```
INTERAKCIJA SA PROGRAMOM:
Unesite dimenziju niza: 145
Greska: neispravan unos.
```

**Zadatak 2.1.16** Napisati program koji vrši pretragu niza nadmorskih visina.

- Napisati funkciju koja proverava da li niz sadrži zadati broj  $m$ . Povratna vrednost funkcije je 1 ako je vrednost sadržana u nizu ili 0 ako nije.
- Napisati funkciju koja vraća vrednost prve pozicije na kojoj se nalazi element koji ima vrednost  $m$  ili  $-1$  ukoliko element nije u nizu.
- Napisati funkciju koja vraća vrednost poslednje pozicije na kojoj se nalazi element koji ima vrednost  $m$  ili  $-1$  ukoliko element nije u nizu.

Program učitava podatke o nadmorskim visinama i ceo broj  $m$ , a zatim ispisuje da li u nizu postoji podatak o unetoj nadmorskoj visini. Ukoliko postoji, ispisuje i poziciju prvog i poslednjeg pojavljivanja vrednosti  $m$  u nizu. Pozicije se broje od 0. Maksimalni broj elemenata niza je 100. U slučaju neispravnog unosa, ispisati odgovarajuću poruku o grešci.

*Primer 1*

```

INTERAKCIJA SA PROGRAMOM:
Unesite dimenziju niza:
7
Unesite podatke:
800 1100 -200 1400 -200 1100 800
Unesite vrednost m:
1100
Nadmorska visina 1100 se nalazi medju podacima.
Pozicija prvog pojavljivanja: 1
Pozicija poslednjeg pojavljivanja: 5

```

*Primer 2*

```

INTERAKCIJA SA PROGRAMOM:
Unesite dimenziju niza:
-5
Greska: neispravan unos.

```

**Zadatak 2.1.17** Marko skuplja sličice za Svetsko prvenstvo u fudbalu. Marko je primetio da mu se neke sličice ponavljaju i rešio je da ih razmeni sa drugarima. Napisati funkciju `int duplikati(int a[], int n, int b[])` koja od niza *a* dimenzije *n* formira niz *b* koji sadrži sve različite elemente niza *a* koji se pojavljuju bar dva puta u nizu. Funkcija kao povratnu vrednost vraća dimenziju niza *b*. Napisati program koji učitava brojeve Markovih sličica i ispisuje sve duplikate. Maksimalni broj elemenata niza je 600. U slučaju neispravnog unosa, ispisati odgovarajuću poruku o grešci.

*Primer 1*

```

INTERAKCIJA SA PROGRAMOM:
Unesite dimenziju 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 dimenziju 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 dimenziju niza: 2
Unesite elemente niza a:
9 5
Elementi niza b:

```

*Primer 4*

```

INTERAKCIJA SA PROGRAMOM:
Unesite dimenziju niza: 0
Greska: neispravan unos.

```

**Zadatak 2.1.18** Palindrom je tekst koji se isto čita i sa leve i sa desne strane. Napisati funkciju koja proverava da li je tekst zadat nizom karaktera palindrom (zanemariti razliku između malih i velikih slova). Napisati program koji učitava dužinu niza i niz karaktera, a zatim ispisuje da li je uneti tekst palindrom. Maksimalni broj elemenata niza je 200. U slučaju neispravnog unosa, ispisati odgovarajuću poruku o grešci.

### Primer 1

```
INTERAKCIJA SA PROGRAMOM:
Unesite dimenziju niza: 15
Unesite elemente niza:
AnaVoliMilovana
Niz jeste palindrom.
```

### Primer 2

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

### Primer 3

```
INTERAKCIJA SA PROGRAMOM:
Unesite dimenziju niza: 1
Unesite elemente niza:
a
Niz jeste palindrom.
```

### Primer 4

```
INTERAKCIJA SA PROGRAMOM:
Unesite dimenziju niza: 226
Greska: neispravan unos.
```

**Zadatak 2.1.19** Napisati funkciju koja proverava da li su elementi celobrojnog niza uređeni neopadajuće. Napisati program koji učitava dimenziju niza, elemente niza, a zatim ispisuje da li je pomenuti uslov ispunjen. Maksimalni broj elemenata niza je 300. U slučaju neispravnog unosa, ispisati odgovarajuću poruku o grešci.

### Primer 1

```
INTERAKCIJA SA PROGRAMOM:
Unesite dimenziju niza: 7
Unesite elemente niza: -40 -8 -8 2 30 30 46
Niz jeste uredjen neopadajuće.
```

### Primer 2

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

### Primer 3

```
INTERAKCIJA SA PROGRAMOM:
Unesite dimenziju niza: 1
Unesite elemente niza: 5
Niz jeste uredjen neopadajuće.
```

### Primer 4

```
INTERAKCIJA SA PROGRAMOM:
Unesite dimenziju niza: 304
Greska: neispravan unos.
```

**Zadatak 2.1.20** U celobrojnom nizu se čuvaju informacije o prodaji artikala jedne prodavnice. Svaki indeks niza označava jedan dan u mesecu, a elementi niza predstavljaju broj artikala koji se prodao tog dana. Napisati funkciju koja računa najdužu uzastopnu seriju dana za koju važi da broj prodatih artikala nije opao. Napisati program koji učitava broj dana u mesecu, broj prodatih artikala za svaki dan u mesecu i zatim ispisuje dužinu izračunate serije. U slučaju neispravnog unosa, ispisati odgovarajuću poruku o grešci.



## 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 neopadajuceg
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 neopadajuceg
prodavanja je 3.

```

## Primer 3

```

INTERAKCIJA SA PROGRAMOM:
Unesite dimenziju niza: -5
Greska: neispravan unos.

```

## Primer 4

```

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
Greska: neispravan unos.

```

**Zadatak 2.1.21** Napisati funkciju koja određuje dužinu najduže serije jednakih uzastopnih elemenata u datom nizu brojeva. Napisati program koji učitava dimenziju niza i elemente niza, a zatim ispisuje dužinu najduže serije jednakih elemenata niza. Maksimalni broj elemenata niza je 100. U slučaju neispravnog unosa, ispisati odgovarajuću poruku o grešci.

## 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.

```

## Primer 3

```

INTERAKCIJA SA PROGRAMOM:
Unesite dimenziju niza: 8
Unesite elemente niza: 1 2 3 4 5 6 7 8
Duzina najduze serije je 1.

```

## Primer 4

```

INTERAKCIJA SA PROGRAMOM:
Unesite dimenziju niza: 108
Greska: neispravan unos.

```

**Zadatak 2.1.22** Napisati funkciju koja određuje da li se jedan niz javlja kao (uzastopni) podniz drugog niza.

- Niz  $b$  je uzastopni podniz niza  $a$  ako su elementi niza  $b$  uzastopni elementi niza  $a$ .
- Niz  $b$  je podniz niza  $a$  ako je redosled pojavljivanja elemenata niza  $b$  u nizu  $a$  isti i ne nužno uzastopan.

Napisati program koji učitava dimenzije i elemente dvaju nizova, a zatim ispisuje

## 2 Napredni tipovi podataka

da li je drugi niz podniz prvog niza. Maksimalni broj elemenata nizova  $a$  i  $b$  je 100. U slučaju neispravnog unosa, ispisati odgovarajuću poruku o grešci.

### 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.
```

**Zadatak 2.1.23** 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  $b$  dimenzije  $n$  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.*

Napisati program koji učitava dimenziju niza i elemente niza i ispisuje da li je uneti niz permutacija. Maksimalni broj elemenata niza je 100. U slučaju neispravnog unosa, ispisati odgovarajuću poruku o grešci.

### Primer 1

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

### Primer 2

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

### Primer 3

```
INTERAKCIJA SA PROGRAMOM:
Unesite dimenziju niza: 1
Unesite elemente niza: 1
Uneti niz je permutacija.
```

### Primer 4

```
INTERAKCIJA SA PROGRAMOM:
Unesite dimenziju niza: 101
Greska: neispravan unos.
```

**Zadatak 2.1.24** Napisati program koji učitava dva cela broja i proverava da li se uneti brojevi zapisuju pomoću istih cifara.

*Primer 1*

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

*Primer 2*

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

*Primer 3*

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

*Primer 4*

```
INTERAKCIJA SA PROGRAMOM:
Unesite dva broja: -1 1
Brojevi se zapisuju istim ciframa.
```

**Zadatak 2.1.25** Napisati program koji vrši transformacije niza.

- Napisati funkciju koja obrće elemente niza.
- Napisati funkciju koja rotira niz ciklično za jedno mesto ulevo.
- Napisati funkciju koja rotira niz ciklično za  $k$  mesta ulevo.

Program učitava dimenziju niza, elemente niza i pozitivan ceo broj  $k$ , a zatim ispisuje niz koji se dobija nakon obrtanja početnog niza, niz koji se dobija rotiranjem tako dobijenog niza za jedno mesto ulevo i niz koji se dobija rotiranjem novodobijenog niza za  $k$  mesta ulevo. Maksimalni broj elemenata niza je 100. U slučaju neispravnog unosa, ispisati odgovarajuću poruku o grešci.

*Primer 1*

```
INTERAKCIJA SA PROGRAMOM:
Unesite dimenziju niza: 6
Unesite elemente niza: 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: neispravan unos.
```

**Zadatak 2.1.26** Napisati funkciju `void ukrsti(int a[], int b[], int n, int c[])` koja formira niz  $c$  koji se dobija naizmeničnim raspoređivanjem elemenata nizova  $a$  i  $b$ , tj.  $c = [a_0, b_0, a_1, b_1, \dots, a_{n-1}, b_{n-1}]$ . Napisati program koji učitava dimenziju i elemente dvaju nizova i ispisuje niz koji se dobija ukrštanjem unetih nizova. Maksimalni broj elemenata niza  $a$  i  $b$  je 100. U slučaju neispravnog unosa, ispisati odgovarajuću poruku o grešci.

*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: neispravan unos.
```

**Zadatak 2.1.27** Napisati funkciju `void spoji(int a[], int b[], int n, int c[])` koja od nizova  $a$  i  $b$  dimenzije  $n$  formira niz  $c$  čija prva polovina odgovara elementima niza  $b$ , a druga polovina elementima niza  $a$ , tj.  $c = [b_0, b_1, \dots, b_{n-1}, a_0, a_1, \dots, a_{n-1}]$ . Napisati program koji učitava dimenziju i elemente dvaju nizova i ispisuje niz koji se dobija spajanjem unetih nizova na pomenuti način. Maksimalni broj elemenata niza  $a$  i  $b$  je 100. U slučaju neispravnog unosa, ispisati odgovarajuću poruku o grešci.

*Primer 1*

```
INTERAKCIJA SA PROGRAMOM:
Unesite dimenziju nizova: 3
Unesite elemente niza a: 4 -8 32
Unesite elemente niza b: 5 2 11
Rezultujući niz:
5 2 11 4 -8 32
```

*Primer 2*

```
INTERAKCIJA SA PROGRAMOM:
Unesite dimenziju nizova: 145
Greska: neispravan unos.
```

\* **Zadatak 2.1.28** Napisati funkciju `void spoji_sortirano(int a[], int b[], int n, int c[])` koja od nizova  $a$  i  $b$  dimenzije  $n$  koji su uređeni neopadajuće formira niz  $c$  koji je uređen na isti način. Napisati program koji učitava dimenziju i elemente uređenih nizova  $a$  i  $b$  i ispisuje niz koji se dobija spajanjem ovih nizova na pomenuti način. Maksimalni broj elemenata niza  $a$  i  $b$  je 100. U slučaju neispravnog unosa, ispisati odgovarajuću poruku o grešci.

*Primer 1*

```
INTERAKCIJA SA PROGRAMOM:
Unesite dimenziju nizova: 5
Unesite elemente sortiranog niza:
2 11 28 40 63
Unesite elemente sortiranog niza:
-19 -5 5 11 52
Rezultujući niz:
-19 -5 2 5 11 11 28 40 52 63
```

*Primer 2*

```
INTERAKCIJA SA PROGRAMOM:
Unesite dimenziju nizova: 3
Unesite elemente sortiranog niza:
-2 4 8
Unesite elemente sortiranog niza:
6 15 19
Rezultujući niz:
-2 4 6 8 15 19
```

*Primer 3*

```
INTERAKCIJA SA PROGRAMOM:
Unesite dimenziju nizova: 145
Greska: neispravan unos.
```

**Zadatak 2.1.29** Napisati funkciju `void promeni_redosled(int a[], int n)` koja menja redosled elementima niza  $a$  dimenzije  $n$  tako da se parni elementi niza nalaze na početku niza, a neparni na kraju. Napisati program koji učitava dimenziju niza i elemente niza i ispisuje niz koji je izmenjen na pomenuti način. Maksimalni broj elemenata niza je 100. U slučaju neispravnog unosa, ispisati

odgovarajuću poruku o grešci. NAPOMENA: *Ne koristiti pomoćne nizove.*

#### Primer 1

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

**Zadatak 2.1.30** Napisati funkciju koja iz datog niza briše sve elemente koji su prosti brojevi. Funkcija kao povratnu vrednost treba da vrati broj elemenata niza nakon brisanja. Napisati program koji učitava dimenziju niza i elemente niza i ispisuje niz koji se dobija brisanjem pomenutih elemenata. Maksimalni broj elemenata niza je 100. U slučaju neispravnog unosa, ispisati odgovarajuću poruku o grešci. NAPOMENA: *Zadatak rešiti uz korišćenje pomoćnog niza.*

#### Primer 1

```
INTERAKCIJA SA PROGRAMOM:
Unesite dimenziju niza: 5
Unesite elemente niza: 11 5 6 48 8
Rezultujući niz: 6 48 8
```

#### Primer 2

```
INTERAKCIJA SA PROGRAMOM:
Unesite dimenziju niza: 4
Unesite elemente niza: 11 5 19 21
Rezultujući niz: 21
```

#### Primer 3

```
INTERAKCIJA SA PROGRAMOM:
Unesite dimenziju niza: 5
Unesite elemente niza: 12 18 9 31 7
Rezultujući niz: 12 18 9
```

#### Primer 4

```
INTERAKCIJA SA PROGRAMOM:
Unesite dimenziju niza: 5
Unesite elemente niza: -2 15 -11 8 7
Rezultujući niz: 15 8
```

**Zadatak 2.1.31** Napisati funkciju koja iz datog niza briše sve neparne elemente. Funkcija kao povratnu vrednost treba da vrati broj elemenata niza nakon brisanja. Napisati program koji učitava dimenziju niza i elemente niza i ispisuje niz koji se dobija brisanjem neparnih elemenata. Maksimalni broj elemenata niza je 100. U slučaju neispravnog unosa, ispisati odgovarajuću poruku o grešci. NAPOMENA: *Zadatak rešiti bez korišćenja pomoćnog niza.*

#### Primer 1

```
INTERAKCIJA SA PROGRAMOM:
Unesite dimenziju niza: 4
Unesite elemente niza:
8 9 15 12
Rezultujući niz: 8 12
```

#### Primer 2

```
INTERAKCIJA SA PROGRAMOM:
Unesite dimenziju niza: 6
Unesite elemente niza:
21 5 3 22 19 188
Rezultujući niz: 22 188
```

#### Primer 3

```
INTERAKCIJA SA PROGRAMOM:
Unesite dimenziju niza: 4
Unesite elemente niza: 133 129 121 101
Rezultujući niz:
```

#### Primer 4

```
INTERAKCIJA SA PROGRAMOM:
Unesite dimenziju niza: 8
Unesite elemente niza:
15 -22 -23 13 18 46 14 -31
Rezultujući niz: -22 18 46 14
```

**Zadatak 2.1.32** Napisati funkciju koja iz datog niza briše sve elemente koji nisu deljivi svojom poslednjom cifrom. Izuzetak su elementi čija je poslednja cifra nula. Funkcija kao povratnu vrednost treba da vrati broj elemenata niza nakon brisanja. Napisati program koji učitava dimenziju niza i elemente niza i ispisuje niz koji se dobija brisanjem pomenutih elemenata. Maksimalni broj elemenata niza je 100. U slučaju neispravnog unosa, ispisati odgovarajuću poruku o grešci. NAPOMENA: *Zadatak rešiti bez korišćenja pomoćnog niza.*

### Primer 1

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

### Primer 2

```
INTERAKCIJA SA PROGRAMOM:
Unesite dimenziju niza: 0
Greska: neispravan unos.
```

**Zadatak 2.1.33** Napisati funkciju koja iz datog niza briše sve brojeve koji nisu deljivi svojim indeksom. Ne razmatrati da li je u novom nizu, nakon brisanja i pomeranja, element deljiv svojim indeksom. Funkcija kao povratnu vrednost treba da vrati broj elemenata niza nakon brisanja. Napisati program koji učitava dimenziju niza i elemente niza i ispisuje niz koji se dobija brisanjem pomenutih elemenata. Maksimalni broj elemenata niza je 700. U slučaju neispravnog unosa, ispisati odgovarajuću poruku o grešci. NAPOMENA: *Nulti element niza treba zadržati jer nije dozvoljeno deljenje nulom. Zadatak rešiti bez korišćenja pomoćnog niza.*

### Primer 1

```
INTERAKCIJA SA PROGRAMOM:
Unesite dimenziju niza: 10
Unesite elemente niza:
4 2 1 6 7 8 10 2 16 3
Rezultujući niz: 4 2 6 16
```

### Primer 2

```
INTERAKCIJA SA PROGRAMOM:
Unesite dimenziju niza: 10
Unesite elemente niza:
-8 5 10 6 7 10 8 2 16 27
Rezultujući niz: -8 5 10 6 10 16 27
```

**Zadatak 2.1.34** Korišćenjem nizova moguće je predstaviti skupove podataka. Napisati program koji demonstrira osnovne operacije nad skupovima (uniju, presek i razliku). Pomoću dva niza predstaviti dva skupa celih brojeva, a zatim ispisati njihovu uniju, presek i razliku. Maksimalni broj elemenata dva uneta niza je 500. U slučaju neispravnog unosa, ispisati odgovarajuću poruku o

grešci.

#### Primer 1

```
INTERAKCIJA SA PROGRAMOM:
Unesite broj elemenata niza a: 5
Unesite elemente niza a: 1 2 3 4 5
Unesite broj elemenata niza b: 3
Unesite elemente niza b: 5 4 9
Unija: 1 2 3 4 5 9
Presek: 4 5
Razlika: 1 2 3
```

#### Primer 2

```
INTERAKCIJA SA PROGRAMOM:
Unesite broj elemenata niza a: 3
Unesite elemente niza a: 11 4 -5
Unesite broj elemenata niza b: 2
Unesite elemente niza b: 18 9
Unija: 11 4 -5 18 9
Presek:
Razlika: 11 4 -5
```

#### Primer 3

```
INTERAKCIJA SA PROGRAMOM:
Unesite broj elemenata niza a: 6
Unesite elemente niza a: 12 7 9 12 5 1
Greska: skup ne moze imati duplikate.
```

#### Primer 4

```
INTERAKCIJA SA PROGRAMOM:
Unesite broj elemenata niza a: -2
Greska: neispravan unos.
```

**Zadatak 2.1.35** Da bi opsluživanje klijenata bilo efikasno i udobno, prilikom ulaska u banku svaki klijent dobija redni broj opsluživanja. Redni brojevi se čuvaju u nizu, počinju od vrednosti 1 i iznova se generišu svakog radnog dana. Postoje i specijalni klijenti (npr. oni koji podižu stambeni kredit) koji mogu dobiti i negativni redni broj da bi se razlikovali od uobičajenih klijenata. Pomozite radniku obezbeđenja da lakše prati redosled opsluživanja klijenata.

- Napisati funkciju koja ubacuje redni broj klijenta  $x$  na kraj niza (klijenta koji je poslednji došao).
- Napisati funkciju koja ubacuje redni broj klijenta  $x$  na početak niza (klijenta koji će biti prvi uslužen, na primer, lica sa posebnim potrebama, trudnice ili stara lica).
- Napisati funkciju koja ubacuje redni broj klijenta  $x$  na poziciju  $k$  koju bira radnik obezbeđenja (manje prioriteta lica, recimo službena lica ili roditelji sa decom).
- Napisati funkciju koja izbacuje prvi redni broj iz niza (redni broj usluženog klijenta).
- Napisati funkciju koja izbacuje poslednji redni broj iz niza (redni broj klijenta koji je odustao jer je shvatio da ima mnogo klijenata ispred njega).
- Napisati funkciju koja izbacuje redni broj iz niza sa pozicije  $k$  (redni broj klijenta koji je odustao jer je dugo čekao).

Napisati program koji testira rad navedenih funkcija. Maksimalni broj klijenata u jednom danu je 2000. U slučaju neispravnog unosa, ispisati odgovarajuću poruku o grešci.

## 2 Napredni tipovi podataka

---

### 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 prioriternog klijenta kojeg treba ubaciti u niz: 36
Niz nakon ubacivanja klijenta: 36 2 5 -2 16 33 19 8 11 35
Unesite prioriternog 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
```



## 2.2 Pokazivači

**Zadatak 2.2.1** Napisati funkciju `void uredi(int *pa, int *pb)` koja uređuje svoja dva celobrojna argumenta tako da se u prvom nalazi manja vrednost, a u drugom veća. Napisati program koji učitava dva cela broja i ispisuje uređene brojeve.

### Primer 1

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite dva broja: 2 5
|| Uredjene promenljive: 2, 5
```

### Primer 2

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite dva broja: 11 -4
|| Uredjene promenljive: -4, 11
```

**Zadatak 2.2.2** Napisati funkciju `void rgb_u_cmy(int r, int g, int b, float *c, float *m, float *y)` koja datu boju u *rgb* formatu konvertuje u boju u *cmy* formatu po sledećim formulama:

$$c = 1 - r/255, \quad m = 1 - g/255, \quad y = 1 - b/255$$

Napisati program koji učitava boju u *rgb* formatu i ispisuje vrednosti unete boje u *cmy* formatu. U slučaju neispravnog unosa, ispisati odgovarajuću poruku o grešci. NAPOMENA: Vrednosti boja u *rgb* formatu su u opsegu  $[0, 255]$ .

### Primer 1

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite boju u rgb formatu: 56 111 24
|| cmy: (0.78, 0.56, 0.91)
```

### Primer 2

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite boju u rgb formatu: 156 -90 5
|| Greska: neispravan unos.
```

### Primer 3

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite boju u rgb formatu: 9 0 237
|| cmy: (0.96, 1.00, 0.07)
```

### Primer 4

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite boju u rgb formatu: 300 11 27
|| Greska: neispravan unos.
```

**Zadatak 2.2.3** Napisati funkciju `int presek(float k1, float n1, float k2, float n2, float *px, float *py)` koja za dve razne prave date svojim koeficijentima pravca i slobodnim članovima određuje njihovu tačku preseka. Funkcija treba da vrati jedinicu ako se prave seku, a nulu ako nemaju tačku preseka (ako su paralelne). Napisati program koji učitava podatke o pravama i ukoliko prave imaju presek, ispisuje koordinate tačke preseka, a ako nemaju, ispisuje odgovarajuću poruku.

### Primer 1

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite k i n za prvu pravu: 4 5
|| Unesite k i n za drugu pravu: 11 -4
|| Prave se seku u tacki (1.29, 10.14).
```

### Primer 2

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite k i n za prvu pravu: 0.5 -4.7
|| Unesite k i n za drugu pravu: 0.5 9.1
|| Prave su paralelne.
```

## 2 Napredni tipovi podataka

**Zadatak 2.2.4** Napisati funkciju koja za dva cela broja izračunava njihov količnik i ostatak pri deljenju. Funkcija treba da vrati jedinicu ukoliko je uspešno izračunala vrednosti, a nulu ukoliko deljenje nije moguće. Napisati program koji učitava dva cela broja i ispisuje njihov količnik i ostatak pri deljenju. U slučaju neispravnog unosa, ispisati odgovarajuću poruku o grešci.

### Primer 1

```
INTERAKCIJA SA PROGRAMOM:  
Unesite brojeve: 4 5  
Kolicnik: 0  
Ostatak: 4
```

### Primer 2

```
INTERAKCIJA SA PROGRAMOM:  
Unesite brojeve: 4 0  
Greska: neispravan unos.
```

### Primer 3

```
INTERAKCIJA SA PROGRAMOM:  
Unesite brojeve: -123 11  
Kolicnik: -11  
Ostatak: -2
```

**Zadatak 2.2.5** Napisati funkciju koja za dužinu trajanja filma koja je data u sekundama, određuje ukupno trajanje filma u satima, minutima i sekundama. Napisati program koji učitava trajanje filma u sekundama i ispisuje odgovarajuće vreme trajanja u formatu *broj\_satih:broj\_minutam:broj\_sekundis*. U slučaju neispravnog unosa, ispisati odgovarajuću poruku o grešci.

### Primer 1

```
INTERAKCIJA SA PROGRAMOM:  
Trajanje fima u sekundama: 5000  
1h:23m:20s
```

### Primer 2

```
INTERAKCIJA SA PROGRAMOM:  
Trajanje fima u sekundama: -300  
Greska: neispravan unos.
```

### Primer 3

```
INTERAKCIJA SA PROGRAMOM:  
Trajanje fima u sekundama: 2500  
0h:41m:40s
```

### Primer 4

```
INTERAKCIJA SA PROGRAMOM:  
Trajanje fima u sekundama: 7824  
2h:10m:24s
```

**Zadatak 2.2.6** Napisati funkciju koja sa ulaza učitava karakter po karakter sve do kraja ulaza, a zatim prebrojava sva pojavljivanja karaktera tačka i sva pojavljivanja karaktera zarez. Napisati program koji za uneti tekst ispisuje koliko puta se pojavila tačka, a koliko puta se pojavio zarez.

### Primer 1

```
INTERAKCIJA SA PROGRAMOM:  
Unesite tekst:  
Bio jednom jedan lav...  
Kakav lav?  
Strasan lav,  
narogusen i ljut sav!  
Broj tacaka: 3  
Broj zareza: 1
```

### Primer 2

```
INTERAKCIJA SA PROGRAMOM:  
Unesite tekst:  
Bavite se sportom,  
ne moze da skodi,  
sportisti su bili  
i bice u modi.  
Kondicije puni,  
uvek vedri, zdravi.  
Svako dete treba  
sportom da se bavi.  
Broj tacaka: 3  
Broj zareza: 4
```

### Primer 3

```
INTERAKCIJA SA PROGRAMOM:  
Unesite tekst:  
Na sirokom carskom drumu  
sto preseca prasumu  
sreli se beli slon  
i jedan crni telefon!  
Broj tacaka: 0  
Broj zareza: 0
```

**Zadatak 2.2.7** Napisati funkciju `void par_nepar(int a[], int n, int parni[], int *np, int neparni[], int *nn)` koja razbija niz *a* na niz parnih i niz neparnih brojeva. Vrednost na koju pokazuje pokazivač *np* treba da bude jednaka broju elemenata niza **parni**, a vrednost na koju pokazuje pokazivač *nn* treba da bude jednaka broju elemenata niza **neparni**. Maksimalan broj elemenata niza je 50. Napisati program koji učitava dimenziju niza, a zatim i elemente niza i ispisuje odgovarajuće nizove parnih, odnosno neparnih elemenata unetog niza. U slučaju neispravnog unosa, ispisati odgovarajuću poruku o grešci.

*Primer 1*

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

*Primer 2*

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

*Primer 3*

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

*Primer 4*

```
INTERAKCIJA SA PROGRAMOM:
Unesite broj elemenata niza: 0
Greska: neispravan unos.
```

**Zadatak 2.2.8** Napisati funkciju koja izračunava najmanji i najveći element niza realnih brojeva. Napisati program koji učitava niz realnih brojeva maksimalne dužine 50 i ispisuje vrednosti najmanjeg i najvećeg elementa niza, zaokružene na tri decimale. U slučaju neispravnog unosa, ispisati odgovarajuću poruku o grešci.

*Primer 1*

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

*Primer 2*

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

*Primer 3*

```
INTERAKCIJA SA PROGRAMOM:
Unesite broj elemenata niza: 1
Unesite elemente niza: 4.16
Najmanji: 4.160
Najveci: 4.160
```

*Primer 4*

```
INTERAKCIJA SA PROGRAMOM:
Unesite broj elemenata niza: -3
Greska: neispravan unos.
```

### 2.3 Niske

**Zadatak 2.3.1** Napisati funkciju `void konvertuj(char s[])` koja menja nisku `s` tako što mala slova zamenjuje odgovarajućim velikim slovima, a velika slova zamenjuje odgovarajućim malim slovima. Napisati program koji učitava nisku maksimalne dužine 10 karaktera i ispisuje konvertovanu nisku.

#### Primer 1

```
INTERAKCIJA SA PROGRAMOM:
Unesite nisku: BeoGrad
Konvertovana niska: bEOgRAD
```

#### Primer 2

```
INTERAKCIJA SA PROGRAMOM:
Unesite nisku: A+B+C
Konvertovana niska: a+b+c
```

#### Primer 3

```
INTERAKCIJA SA PROGRAMOM:
Unesite nisku: 12345
Konvertovana niska: 12345
```

**Zadatak 2.3.2** Napisati funkciju `void ubaci_zvezdice(char s[])` koja menja nisku `s` tako što u njoj svaki drugi karakter zamenjuje zvezdicom. Napisati program koji učitava nisku maksimalne dužine 20 karaktera i ispisuje izmenjenu nisku.

#### Primer 1

```
INTERAKCIJA SA PROGRAMOM:
Unesite nisku: *a*b*c*
Izmenjena niska: *****
```

#### Primer 2

```
INTERAKCIJA SA PROGRAMOM:
Unesite nisku: zimA
Izmenjena niska: z*m*
```

#### Primer 3

```
INTERAKCIJA SA PROGRAMOM:
Unesite nisku: 123abc789
Izmenjena niska: 1*3*b*7*9
```

**Zadatak 2.3.3** Napisati program koji vrši poređenje niski. Napisati funkcije:

- (a) `int jednake(char s1[], char s2[])` koja vraća jedinicu ako su  $s_1$  i  $s_2$  jednake niske, a nulu inače.
- (b) `void u_velika_slova(char s[])` koja pretvara sva slova niske `s` u velika slova, a ostale karaktere ne menja.

Program učitava dve reči maksimalne dužine 20 karaktera i ispituje da li su unete reči jednake. Pri poređenju treba zanemariti razliku između malih i velikih slova.

#### Primer 1

```
INTERAKCIJA SA PROGRAMOM:
Unesite niske:
isPit2010
IsPiT2010
Niske su jednake.
```

#### Primer 2

```
INTERAKCIJA SA PROGRAMOM:
Unesite niske:
Prog1
prog2
Niske nisu jednake.
```

#### Primer 3

```
INTERAKCIJA SA PROGRAMOM:
Unesite niske:
jun
JUNSKI
Niske nisu jednake.
```

**Zadatak 2.3.4** Napisati program koji proverava da li se uneta niska završava samoglasnikom. Napisati funkcije:

- (a) `int samoglasnik(char c)` koja ispituje da li je karakter `c` samoglasnik i vraća 1 ako jeste ili 0 ako nije.

- (b) `int samoglasnik_na_kraju(char s[])` koja ispituje da li se niska *s* završava samoglasnikom.

Program učitava reč maksimalne dužine 20 karaktera i ispisuje da li se reč završava samoglasnikom ili ne.

*Primer 1*

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite nisku: kestenje
|| Niska se završava samoglasnikom.
```

*Primer 2*

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite nisku: vetar
|| Niska se ne završava samoglasnikom.
```

*Primer 3*

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite nisku: OLUJA
|| Niska se završava samoglasnikom.
```

*Primer 4*

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite nisku: Programiranje1
|| Niska se ne završava samoglasnikom.
```

**Zadatak 2.3.5** Napisati funkciju `int sadrzi_veliko(char s[])` koja proverava da li niska *s* sadrži veliko slovo. Napisati program koji za učitanu nisku maksimalne dužine 20 karaktera proverava da li sadrži veliko slovo i ispisuje odgovarajuću poruku.

*Primer 1*

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite nisku:
|| naocare
|| Ne sadrzi veliko slovo.
```

*Primer 2*

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite nisku:
|| DiopTrija0.75
|| Sadrzi veliko slovo.
```

*Primer 3*

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite nisku:
|| 21.06.2017.
|| Ne sadrzi veliko slovo.
```

**Zadatak 2.3.6** Napisati program koji za učitanu nisku *s* i karakter *c* ispituje da li se karakter *c* pojavljuje u niski *s*. Ako je to slučaj, program treba da ispiše indeks prvog pojavljivanja karaktera *c* u niski *s*, a u suprotnom -1. Pretpostaviti da niska može da ima najviše 20 karaktera.

*Primer 1*

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite nisku: bazen
|| Unesite karakter: z
|| Pozicija: 2
```

*Primer 2*

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite nisku: lezaljka
|| Unesite karakter: a
|| Pozicija: 3
```

*Primer 3*

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite nisku: limunada
|| Unesite karakter: b
|| Pozicija: -1
```

**Zadatak 2.3.7** Napisati funkciju `int podniska(char s[], char t[])` koja proverava da li je niska *t* uzastopna podniska niske *s*. Napisati program koji učitava dve niske maksimalne dužine 10 karaktera i ispisuje da li je druga niska

## 2 Napredni tipovi podataka

podniska prve.

### Primer 1

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

### Primer 2

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

### Primer 3

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

**Zadatak 2.3.8** Napisati funkciju `void skрати(char s[])` koja uklanja beline sa kraja date niske. Napisati program koji učitava liniju maksimalne dužine 100 karaktera i ispisuje učitano i izmenjenu nisku između zvezdica.

### Primer 1

```
INTERAKCIJA SA PROGRAMOM:
Unesite nisku:
rep belina
Ucitana niska:
*rep belina
Izmenjena niska:
*rep belina*
```

### Primer 2

```
INTERAKCIJA SA PROGRAMOM:
Unesite nisku:
tri tabulatora na kraju
Ucitana niska:
*tri tabulatora na kraju
Izmenjena niska:
*tri tabulatora na kraju*
```

**Zadatak 2.3.9** Napisati funkciju `void ukloni_slova(char s[])` koja iz niske `s` uklanja sva mala i sva velika slova. Napisati program koji za učitano nisku maksimalne dužine 20 karaktera ispisuje odgovarajuću izmenjenu nisku.

### Primer 1

```
INTERAKCIJA SA PROGRAMOM:
Unesite nisku: a1b2c3def
Rezultat: 123
```

### Primer 2

```
INTERAKCIJA SA PROGRAMOM:
Unesite nisku: 1+2=3
Rezultat: 1+2=3
```

### Primer 3

```
INTERAKCIJA SA PROGRAMOM:
Unesite nisku: malaVELIKA
Rezultat:
```

**Zadatak 2.3.10** Napisati funkciju `void ukloni(char *s)` koja iz niske uklanja sva slova iza kojih sledi slovo koje je u engleskoj abecedi nakon njih, pri čemu se veličina slova zanemaruje. Pravilo se ne primenjuje na nisku dobijenu uklanjanjem. Napisati program koji učitava liniju teksta koja ima najviše 100 karaktera i ispisuje liniju koja se dobije nakon uklanjanja pomenutih karaktera.

### Primer 1

```
INTERAKCIJA SA PROGRAMOM:
Unesite nisku:
Zdravo svima!
Izmenjena niska:
Zrvo vma!
```

### Primer 2

```
INTERAKCIJA SA PROGRAMOM:
Unesite nisku:
Danas je 10 stepeni.
Izmenjena niska:
Dns j 10 tpni.
```

### Primer 3

```
INTERAKCIJA SA PROGRAMOM:
Unesite nisku:
Ima vetra, kise i hladnoce.
Izmenjena niska:
ma vtra, kse i loe.
```

**Zadatak 2.3.11** Napisati program koji učitava nisku `s` maksimalne dužine 30 karaktera i formira nisku `t` trostrukim nadovezivanjem niske `s`.

## Primer 1

```
INTERAKCIJA SA PROGRAMOM:
Unesite nisku: dan
Rezultujuca niska:
dandandan
```

## Primer 2

```
INTERAKCIJA SA PROGRAMOM:
Unesite nisku: 3sesira
Rezultujuca niska:
3sesira3sesira3sesira
```

## Primer 3

```
INTERAKCIJA SA PROGRAMOM:
Unesite nisku: a-b=5
Rezultujuca niska:
a-b=5a-b=5a-b=5
```

**Zadatak 2.3.12** Napisati program koji za unetu reč maksimalne dužine 20 karaktera i pozitivan broj  $n$  manji od 10, formira rezultujuću reč tako što unetu reč kopira  $n$  puta pri čemu se između svaka dva kopiranja umeće crtica. U slučaju neispravnog unosa, ispisati odgovarajuću poruku o grešci.

## Primer 1

```
INTERAKCIJA SA PROGRAMOM:
Unesite nisku: ana
Unesite broj n: 4
Rezultujuca niska:
ana-ana-ana-ana
```

## Primer 2

```
INTERAKCIJA SA PROGRAMOM:
Unesite nisku: 123
Unesite broj n: 1
Rezultujuca niska:
123
```

## Primer 3

```
INTERAKCIJA SA PROGRAMOM:
Unesite nisku: x*y
Unesite broj n: 3
Rezultujuca niska:
x*y-x*y-x*y
```

**Zadatak 2.3.13** Napisati funkciju `void kopiraj_n(char t[], char s[], int n)` koja kopira najviše  $n$  karaktera niske  $s$  u nisku  $t$ . Napisati program koji testira rad napisane funkcije. Pretpostaviti da je maksimalna dužina niske  $s$  20 karaktera. U slučaju neispravnog unosa, ispisati odgovarajuću poruku o grešci.

## Primer 1

```
INTERAKCIJA SA PROGRAMOM:
Unesite nisku: petar
Unesite broj n: 3
Rezultujuca niska: pet
```

## Primer 2

```
INTERAKCIJA SA PROGRAMOM:
Unesite nisku: gromobran
Unesite broj n: 4
Rezultujuca niska: grom
```

## Primer 3

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

**Zadatak 2.3.14** Napisati funkciju `void dupliranje(char t[], char s[])` koja na osnovu niske  $s$  formira nisku  $t$  tako što duplira svaki karakter niske  $s$ . Napisati program koji učitava nisku maksimalne dužine 20 karaktera i ispisuje nisku koja se dobije nakon dupliranja karaktera.

## Primer 1

```
INTERAKCIJA SA PROGRAMOM:
Unesite nisku: zima
Rezultujuca niska: zziimmaa
```

## Primer 2

```
INTERAKCIJA SA PROGRAMOM:
Unesite nisku: C++
Rezultujuca niska: CC+++
```

## Primer 3

```
INTERAKCIJA SA PROGRAMOM:
Unesite nisku: C
Rezultujuca niska: CC
```

**Zadatak 2.3.15** Napisati program koji učitava nisku cifara sa eventualnim

## 2 Napredni tipovi podataka

---

vodećim znakom i pretvara je u ceo broj. NAPOMENA: *Pretpostaviti da je unos ispravan.*

### Primer 1

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite nisku: -1238
|| Rezultat: -1238
```

### Primer 2

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite nisku: 73
|| Rezultat: 73
```

### Primer 3

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite nisku: +1
|| Rezultat: 1
```

**Zadatak 2.3.16** Napisati program koji učitava ceo broj, pretvara ga u nisku i ispisuje dobijenu nisku.

### Primer 1

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite ceo broj: -6543
|| Rezultat: -6543
```

### Primer 2

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite ceo broj: 84
|| Rezultat: 84
```

### Primer 3

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite ceo broj: 5
|| Rezultat: 5
```

**Zadatak 2.3.17** Napisati funkciju `int heksadekadni_broj(char s[])` koja proverava da li je niskom *s* zadat korektan heksadekadni broj. Funkcija treba da vrati vrednost 1 ukoliko je uslov ispunjen, odnosno 0 ako nije. Napisati program koji za učitanu nisku maksimalne dužine 7 karaktera ispisuje da li je korektan heksadekadni broj. UPUTSTVO: *Heksadekadni broj je korektno zadat ako počinje prefiksom 0x ili 0X i ako sadrži samo cifre i mala ili velika slova A, B, C, D, E i F.*

### Primer 1

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

### Primer 2

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

### Primer 3

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

**Zadatak 2.3.18** Napisati funkciju `int dekadna_vrednost(char s[])` koja izračunava dekadnu vrednost heksadekadnog broja zadatog niskom *s*. Napisati program koji za učitanu nisku maksimalne dužine 7 karaktera ispisuje odgovarajuću dekadnu vrednost. Pretpostaviti da je uneta niska korektan heksadekadni broj.

### Primer 1

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

### Primer 2

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite nisku: 0Xff2
|| Rezultat: 4082
```

### Primer 3

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



**Zadatak 2.3.19** Napisati funkciju `int ucitaj_liniju(char s[], int n)` koja učitava liniju maksimalne dužine  $n$  u nisku  $s$  i vraća dužinu učitane linije. Napisati program koji učitava linije do kraja ulaza i ispisuje najdužu liniju i njenu dužinu. Ukoliko ima više linija maksimalne dužine, ispisati prvu. Pretpostaviti da svaka linija sadrži najviše 80 karaktera. NAPOMENA: *Linija može da sadrži blanko znakove, ali ne može sadržati znak za novi red ili EOF.*

*Primer 1*

```
INTERAKCIJA SA PROGRAMOM:
Unesite tekst:
Dobar dan!
Kako ste, sta ima novo?
Ja sam dobro.
Najduza linija:
Kako ste, sta ima novo?
Duzina: 23
```

*Primer 2*

```
INTERAKCIJA SA PROGRAMOM:
Unesite tekst:
Prva linija
Druga linija
Trecu linija
Najduza linija:
Druga linija
Duzina: 12
```

*Primer 3*

```
INTERAKCIJA SA PROGRAMOM:
Unesite tekst:
Danas je lep dan.
Najduza linija:
Danas je lep dan.
Duzina: 17
```

\* **Zadatak 2.3.20** Napisati funkcije za rad sa rečenicama:

- (a) `int procitaj_recenicu(char s[], int n)` koja učitava rečenicu sa ulaza i smešta je u nisku  $s$ . Funkcija vraća dužinu učitane rečenice. Učitavanje se završava nakon učitanoj karaktera `.`, nakon  $n$  učitanih karaktera ili ako se dođe do kraja ulaza.
- (b) `void prebroj(char s[], int *broj_malih, int *broj_velikih)` koja prebrojava mala i velika slova u niski  $s$ .

Napisati program koji učitava rečenice do kraja ulaza i ispisuje onu rečenicu kod koje je apsolutna razlika broja malih i velikih slova najveća. Pri učitavanju rečenica zanemariti sve beline koje se nalaze između dve rečenice. Pretpostaviti da jedna rečenica sadrži najviše 80 karaktera.

*Primer 1*

```
INTERAKCIJA SA PROGRAMOM:
Unesite tekst:
U ovom poglavlju se govori o niskama. Niske su nizovi karaktera ciji je
poslednji element terminalna nula.
U ovom zadatku je potrebno ucitati recenice. Svaka recenica pocinje sa bilo
kojim karakterom koji nije belina. Na kraju recenice se nalazi tacka.
Rezultujuca recenica:
Niske su nizovi karaktera ciji je poslednji element terminalna nula.
```

**Zadatak 2.3.21** Napisati funkciju `char* strchr_klon(char s[], char c)` koja vraća pokazivač na prvo pojavljivanje karaktera  $c$  u niski  $s$  ili `NULL` ukoliko se karakter  $c$  ne pojavljuje u niski  $s$ .<sup>1</sup> Napisati program koji za učitano nisku

<sup>1</sup>Funkcija `strchr_klon` odgovara funkciji `strchr` čija se deklaracija nalazi u zaglavlju `string.h`.

## 2 Napredni tipovi podataka

---

maksimalne dužine 20 karaktera i karakter *c* ispisuje indeks prvog pojavljivanja karaktera *c* u okviru učitane niske ili  $-1$  ukoliko učitana niska ne sadrži uneti karakter.

### Primer 1

```
INTERAKCIJA SA PROGRAMOM:
Unesite nisku s: programiranje
Unesite karakter c: a
Pozicija: 5
```

### Primer 2

```
INTERAKCIJA SA PROGRAMOM:
Unesite nisku s: 123456789
Unesite karakter c: y
Pozicija: -1
```

### Primer 3

```
INTERAKCIJA SA PROGRAMOM:
Unesite nisku s: leto2017
Unesite karakter c: 0
Pozicija: 5
```

### Primer 4

```
INTERAKCIJA SA PROGRAMOM:
Unesite nisku s: jedrilica
Unesite karakter c: I
Pozicija: -1
```

**Zadatak 2.3.22** Napisati funkciju `int strspn_klon(char t[], char s[])` koja izračunava dužinu prefiksa niske *t* sastavljenog od karaktera niske *s*. Napisati program koji za učitane dve niske maksimalne dužine 20 karaktera ispisuje rezultat poziva napisane funkcije.

### Primer 1

```
INTERAKCIJA SA PROGRAMOM:
Unesite nisku t: program
Unesite nisku s: pero
Rezultat: 3
```

### Primer 2

```
INTERAKCIJA SA PROGRAMOM:
Unesite nisku t: Barcelona
Unesite nisku s: Brazil
Rezultat: 3
```

### Primer 3

```
INTERAKCIJA SA PROGRAMOM:
Unesite nisku t: 24.10.2017.
Unesite nisku s: 0123456789
Rezultat: 2
```

### Primer 4

```
INTERAKCIJA SA PROGRAMOM:
Unesite nisku t: 12345
Unesite nisku s: 9876543210
Rezultat: 5
```

**Zadatak 2.3.23** Napisati funkciju `int strcspn_klon(char t[], char s[])` koja izračunava dužinu prefiksa niske *t* sastavljenog isključivo od karaktera koji se ne nalaze u niski *s*. Napisati program koji testira ovu funkciju za dve unete niske maksimalne dužine 100 karaktera.

---

Slično važi i za ostale *klon* funkcije iz narednih zadataka.

## Primer 1

```
INTERAKCIJA SA PROGRAMOM:
Unesite nisku t:
programiranje
Unesite nisku s:
pero
Rezultat: 0
```

## Primer 2

```
INTERAKCIJA SA PROGRAMOM:
Unesite nisku t:
programiranje
Unesite nisku s:
analiza
Rezultat: 5
```

## Primer 3

```
INTERAKCIJA SA PROGRAMOM:
Unesite nisku t:
programiranje
Unesite nisku s:
1.10.
Rezultat: 13
```

**Zadatak 2.3.24** Napisati funkciju `char* strstr_klon(char s[], char t[])` koja vraća pokazivač na prvo pojavljivanje niske *t* u niski *s* ili *NULL* ukoliko se niska *t* ne pojavljuje u niski *s*. Napisati program koji testira napisanu funkciju tako što učitava pet linija i ispisuje redne brojeve svih linija koje sadrže nisku *program*. Ukoliko ne postoji linija sa niskom *program*, ispisati odgovarajuću poruku. Pretpostaviti da je svaka linija maksimalne dužine 100 karaktera kao i da se linije numerišu od broja 1.

## Primer 1

```
INTERAKCIJA SA PROGRAMOM:
Unesite pet linija:
tu program
c prog. jezik
c++ programskih jezik
Programski odbor
<b>program</b>
Rezultat: 1 3 5
```

## Primer 2

```
INTERAKCIJA SA PROGRAMOM:
Unesite pet linija:
Programske paradigme
su predmet na
trecoj godini
programerskih
smerova.
Rezultat: 4
```

## Primer 3

```
INTERAKCIJA SA PROGRAMOM:
Unesite pet linija:
U narednim
linijama
necemo navoditi
nisku koja se
trazi.
Nijedna linija ne sadrzi
nisku program.
```

**Zadatak 2.3.25** Napisati funkciju `int strcmp_klon(char s[], char t[])` koja vraća 0 ako su niske *s* i *t* jednake, neku pozitivnu vrednost ako je *s* leksikografski iza *t*, a neku negativnu vrednost inače. Napisati program koji učitava dve niske maksimalne dužine 20 karaktera i ako su različite, ispisuje učitane niske u rastućem leksikografskom poretku, a ako su jednake, ispisuje samo jednu nisku.

## Primer 1

```
INTERAKCIJA SA PROGRAMOM:
Unesite nisku s: Beograd
Unesite nisku t: Amsterdam
Rezultat:
Amsterdam
Beograd
```

## Primer 2

```
INTERAKCIJA SA PROGRAMOM:
Unesite nisku s: Beograd
Unesite nisku t: Beograd
Rezultat:
Beograd
```

## Primer 3

```
INTERAKCIJA SA PROGRAMOM:
Unesite nisku s: radnik
Unesite nisku t: radnica
Rezultat:
radnica
radnik
```

**Zadatak 2.3.26** Napisati funkciju `void obrni(char s[])` koja obrće nisku *s*. Napisati program koji obrće učitane niske maksimalne dužine 20 karaktera i

## 2 Napredni tipovi podataka

---

ispisuje obrnutu nisku.

### Primer 1

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite nisku: kisobran
|| Rezultat: narbosik
```

### Primer 2

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite nisku: Aleksandar
|| Rezultat: radnaskela
```

### Primer 3

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite nisku: kajak
|| Rezultat: kajak
```

**Zadatak 2.3.27** Napisati funkciju `void rotiraj(char s[], int k)` koja rotira nisku *s* za *k* mesta ulevo. Napisati program koji učitava nisku maksimalne dužine 20 karaktera i nenegativan ceo broj *k* i ispisuje rotiranu nisku. U slučaju neispravnog unosa, ispisati odgovarajuću poruku o grešci.

### Primer 1

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite nisku i broj k:
|| sveska 2
|| Rezultat: eskav
```

### Primer 2

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite nisku i broj k:
|| olovka 6
|| Rezultat: olovka
```

### Primer 3

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite nisku i broj k:
|| rezac 8
|| Rezultat: acrez
```

**Zadatak 2.3.28** Napisati program koji šifrue unetu nisku tako što svako slovo zamenjuje sledećim slovom engleske abecede (slova 'z' i 'Z' zamenjuje, redom, sa 'a' i 'A'), a ostale karaktere ostavlja nepromenjene. Ispisati nisku dobijenu na ovaj način. Pretpostaviti da uneta niska nije duža od 20 karaktera.

### Primer 1

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite nisku: bundeva
|| Rezultat: cvcoefwb
```

### Primer 2

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite nisku: zimzelen
|| Rezultat: ajnafmfo
```

### Primer 3

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite nisku: Oktobar17
|| Rezultat: Plupcbs17
```

**Zadatak 2.3.29** Napisati funkciju `void sifruj(char rec[], char si-fra[])` koja na osnovu date reči formira šifru tako što se svako slovo u reči zameni sa naredna tri slova engleske abecede (nakon slova 'z' tj. 'Z' sledi slovo 'a' tj. 'A'). Napisati program koji testira napisanu funkciju za reč maksimalne dužine 20 karaktera.

### Primer 1

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite nisku: tamo
|| Rezultat: uvwbcndoppqr
```

### Primer 2

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite nisku: Zec
|| Rezultat: ABCfghdef
```

### Primer 3

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite nisku: a+b=c
|| Rezultat: bcd+cde=def
```

**Zadatak 2.3.30** Napisati funkciju `void formiraj(char s1[], char s2[], char c1, char c2)` koja na osnovu niske *s*<sub>1</sub> formira nisku *s*<sub>2</sub> udvajanjem svih karaktera *c*<sub>1</sub> u niski *s*<sub>1</sub> i izbacivanjem svih karaktera *c*<sub>2</sub> iz niske *s*<sub>1</sub>, dok ostali karakteri ostaju nepromenjeni. Napisati program koji testira ovu funkciju za unetu nisku i dva uneta karaktera. Pretpostaviti da uneta niska nije duža od 20

karaktera.

#### Primer 1

```
INTERAKCIJA SA PROGRAMOM:
Unesite nisku: flomaster
Unesite prvi karakter: s
Unesite drugi karakter: m
Rezultat: floasster
```

#### Primer 2

```
INTERAKCIJA SA PROGRAMOM:
Unesite nisku: bojica
Unesite prvi karakter: b
Unesite drugi karakter: a
Rezultat: bbojic
```

#### Primer 3

```
INTERAKCIJA SA PROGRAMOM:
Unesite nisku: patentara
Unesite prvi karakter: t
Unesite drugi karakter: a
Rezultat: pttenttr
```

\* **Zadatak 2.3.31** Napisati program za rad sa brojevima zapisanim u različitim brojevnim sistemima.

- Napisati funkciju `unsigned int u_dekadni_sistem(char broj[], unsigned int osnova)` koja određuje dekadnu vrednost zapisa datog neoznačenog broja *broj* u datoj osnovi *osnova*.
- Napisati funkciju `void iz_dekadnog_sistema(unsigned int broj, unsigned int osnova, char rezultat[])` koja datu dekadnu vrednost *broj* zapisuje u datoj osnovi *osnova* i smešta rezultat u nisku *rezultat*. Pretpostaviti da je  $0 < osnova \leq 16$ .

Napisati program koji učitava broj *n* koji se zadaje kao niska cifara i osnove  $o_1$  i  $o_2$  i ispisuje dekadnu vrednost broja *n* u osnovi  $o_1$ , kao i zapis tako dobijene dekadne vrednosti u osnovi  $o_2$ . Pretpostaviti da je maksimalna dužina zapisa broj 20 karaktera i da će svi brojevi biti ispravno zadati tj. u opsegu tipa `unsigned`.

#### Primer 1

```
INTERAKCIJA SA PROGRAMOM:
Unesite n, o1 i o2: 10101011 2 16
Dekadna vrednost broja 10101011: 171
Vrednost broja 171 u osnovi 16: AB
```

#### Primer 2

```
INTERAKCIJA SA PROGRAMOM:
Unesite n, o1 i o2: 1067 8 3
Dekadna vrednost broja 1067: 567
Zapis broja 567 u osnovi 3: 210000
```

#### Primer 3

```
INTERAKCIJA SA PROGRAMOM:
Unesite n, o1 i o2: 1010111001010 2 3
Dekadna vrednost broja 1010111001010: 5578
Zapis broja 5578 u osnovi 3: 21122121
```

#### Primer 4

```
INTERAKCIJA SA PROGRAMOM:
Unesite n, o1 i o2: 111 3 5
Dekadna vrednost broja 111: 13
Zapis broja 13 u osnovi 5: 23
```

## 2.4 Višedimenzioni nizovi

**Zadatak 2.4.1** Napisati program koji učitava i zatim ispisuje elemente učitane matrice. Sa ulaza se najpre učitavaju dva cela broja  $m$  i  $n$ , a potom i elementi matrice celih brojeva dimenzije  $m \times n$ . Pretpostaviti da je maksimalna dimenzija matrice  $50 \times 50$ . U slučaju neispravnog unosa, ispisati odgovarajuću poruku o grešci.

### Primer 1

```
INTERAKCIJA SA PROGRAMOM:
Unesite broj vrsta i
broj kolona matrice:
3 4
Unesite elemente matrice:
1 2 3 4
5 6 7 8
9 10 11 12
Matrica je:
1 2 3 4
5 6 7 8
9 10 11 12
```

### Primer 2

```
INTERAKCIJA SA PROGRAMOM:
Unesite broj vrsta i
broj kolona matrice:
5 3
Unesite elemente matrice:
1 1 2
5 0 2
7 8 9
1 2 4
0 1 1
Matrica je:
1 1 2
5 0 2
7 8 9
1 2 4
0 1 1
```

### Primer 3

```
INTERAKCIJA SA PROGRAMOM:
Unesite broj vrsta i
broj kolona matrice:
500 3
Greska: neispravan unos.
```

**Zadatak 2.4.2** Napisati program koji za učitane celobrojnu matricu<sup>2</sup> dimenzije  $m \times n$  izračunava i štampa na tri decimale njenu euklidsku normu. Pretpostaviti da je maksimalna dimenzija matrice  $50 \times 50$ . U slučaju neispravnog unosa, ispisati odgovarajuću poruku o grešci. UPUTSTVO: *Euklidska norma matrice je kvadratni koren sume kvadrata svih elemenata matrice.*

### Primer 1

```
INTERAKCIJA SA PROGRAMOM:
Unesite broj vrsta i
broj kolona matrice:
3 4
Unesite elemente matrice:
1 2 3 4
5 6 7 8
9 10 11 12
Euklidska norma: 25.495
```

### Primer 2

```
INTERAKCIJA SA PROGRAMOM:
Unesite broj vrsta i
broj kolona matrice:
5 3
Unesite elemente matrice:
1 1 2
5 0 2
7 8 9
1 2 4
0 1 1
Euklidska norma: 15.875
```

### Primer 3

```
INTERAKCIJA SA PROGRAMOM:
Unesite broj vrsta i
broj kolona matrice:
500 3
Greska: neispravan unos.
```

**Zadatak 2.4.3** Napisati funkcije za rad sa celobrojnim matricama:

<sup>2</sup>Pod pojmom *učitati matricu* ili *za datu matricu* uvek se podrazumeva da se prvo unose dimenzije matrice, a potom i sama matrica.

- (a) `void ucitaj(int a[][MAKS], int n, int m)` kojom se učitavaju elementi matrice celih brojeva  $a$  dimenzije  $m \times n$ ,
- (b) `void ispisi(int a[][MAKS], int n, int m)` kojom se ispisuju elementi matrice  $a$  dimenzije  $m \times n$ .

Napisati program koji najpre učitava, a zatim i ispisuje elemente učitane matrice. Pretpostaviti da je maksimalna dimenzija matrice  $50 \times 50$ . U slučaju neispravnog unosa, ispisati odgovarajuću poruku o grešci. NAPOMENA: U ovom i u narednim zadacima, konstanta **MAKS** u prototipu funkcije označava maksimalni broj kolona date matrice i potrebno ju je definisati u rešenju direktivom `#define`.

*Primer 1*

```
INTERAKCIJA SA PROGRAMOM:
Unesite broj vrsta i
broj kolona matrice:
3 4
Unesite elemente matrice:
1 2 3 4
5 6 7 8
9 10 11 12
Matrica je:
1 2 3 4
5 6 7 8
9 10 11 12
```

*Primer 2*

```
INTERAKCIJA SA PROGRAMOM:
Unesite broj vrsta i
broj kolona matrice:
2 5
Unesite elemente matrice:
1 1 2 3 4
5 0 2 5 7
Matrica je:
1 1 2 3 4
5 0 2 5 7
```

*Primer 3*

```
INTERAKCIJA SA PROGRAMOM:
Unesite broj vrsta i
broj kolona matrice:
500 3
Greska: neispravan unos.
```

**Zadatak 2.4.4** Napisati funkciju `void transponovana(int a[][MAKS], int m, int n, int b[][MAKS])` koja određuje matricu  $b$  koja je dobijena transponovanjem matrice  $a$ . Napisati program koji za učitane matricu celih brojeva ispisuje odgovarajuću transponovanu matricu. Pretpostaviti da je maksimalna dimenzija matrice  $50 \times 50$ . U slučaju neispravnog unosa, ispisati odgovarajuću poruku o grešci.

*Primer 1*

```
INTERAKCIJA SA PROGRAMOM:
Unesite broj vrsta i
broj kolona matrice:
3 4
Unesite elemente matrice:
1 2 3 4
5 6 7 8
9 10 11 12
Transponovana matrica je:
1 5 9
2 6 10
3 7 11
4 8 12
```

*Primer 2*

```
INTERAKCIJA SA PROGRAMOM:
Unesite broj vrsta i
broj kolona matrice:
5 3
Unesite elemente matrice:
1 1 2
5 0 2
7 8 9
1 2 4
0 1 1
Transponovana matrica je:
1 5 7 1 0
1 0 8 2 1
2 2 9 4 1
```

*Primer 3*

```
INTERAKCIJA SA PROGRAMOM:
Unesite broj vrsta i
broj kolona matrice:
500 3
Greska: neispravan unos.
```

**Zadatak 2.4.5** Napisati funkciju `void razmeni(int a[][MAKS], int m, int n, int k, int t)` u kojoj se razmenjuju elementi  $k$ -te i  $t$ -te vrste matrice  $a$  dimezije  $m \times n$ . Napisati program koji za učitane matricu celih brojeva i dva cela broja  $k$  i  $t$  ispisuje matricu dobijenu razmenjivanjem  $k$ -te i  $t$ -te vrste ulazne matrice. Pretpostaviti da je maksimalna dimenzija matrice  $50 \times 50$ . U slučaju neispravnog unosa, ispisati odgovarajuću poruku o grešci.

### Primer 1

```
INTERAKCIJA SA PROGRAMOM:
Unesite broj vrsta i
broj kolona matrice:
3 4
Unesite elemente matrice:
1 2 3 4
5 6 7 8
9 10 11 12
Unesite indekse vrsta:
0 2
Rezultujuća matrica:
9 10 11 12
5 6 7 8
1 2 3 4
```

### Primer 2

```
INTERAKCIJA SA PROGRAMOM:
Unesite broj vrsta i
broj kolona matrice:
5 3
Unesite elemente matrice:
1 1 2
5 0 2
7 8 9
1 2 4
0 1 1
Unesite indekse vrsta:
1 3
Rezultujuća matrica:
1 1 2
1 2 4
7 8 9
5 0 2
0 1 1
```

### Primer 3

```
INTERAKCIJA SA PROGRAMOM:
Unesite broj vrsta i
broj kolona matrice:
5 3
Unesite elemente matrice:
1 1 2
5 0 2
7 8 9
1 2 4
0 1 1
Unesite indekse vrsta:
-1 50
Greska: neispravan unos.
```

**Zadatak 2.4.6** Napisati program koji za učitane matricu celih brojeva ispisuje indekse onih elemenata matrice koji su jednaki zbiru svih svojih susednih elemenata. Pretpostaviti da je maksimalna dimenzija matrice  $50 \times 50$ . U slučaju neispravnog unosa, ispisati odgovarajuću poruku o grešci.

```
- - - - - s b s -
- s s s - - s s s -
- s a s - - - - -
- s s s - - - - -
- - - - - s s
- - - - - s c
```

Slika 2.1: Susedni elementi u matrici.

**UPUTSTVO:** Elementi matrice  $m$  susedni elementu  $m[i][j]$  su svi elementi matrice čiji se indeksi, po apsolutnoj vrednosti, razlikuju najviše za jedan. Element matrice može imati najviše osam suseda:  $m[i-1][j-1]$ ,  $m[i-1][j]$ ,  $m[i-1][j+1]$ ,  $m[i][j-1]$ ,  $m[i][j+1]$ ,  $m[i+1][j-1]$ ,  $m[i+1][j]$  i  $m[i+1][j+1]$ . U zavisnosti od položaja u matrici, element matrice može imati i tri ili pet suseda. Na slici 2.1 su slovom *s* obeleženi susedni elementi matrice za elemente  $m[2][2]$  (ele-



ment je na slici obeležen sa a),  $m[0][7]$  (element je na slici obeležen sa b) i  $m[5][9]$  (element je na slici obeležen sa c).

## Primer 1

```
INTERAKCIJA SA PROGRAMOM:
Unesite broj vrsta i
broj kolona matrice:
4 5
Unesite elemente matrice:
1 1 2 1 3
0 8 1 9 0
1 1 1 0 0
0 3 0 2 2
Indeksi elemenata koji su
jednaki zbiru suseda su:
1 1
3 1
3 4
```

## Primer 2

```
INTERAKCIJA SA PROGRAMOM:
Unesite broj vrsta i
broj kolona matrice:
3 4
Unesite elemente matrice:
7 10 12 20
-1 -3 1 7
0 -47 2 0
Indeksi elemenata koji su
jednaki zbiru suseda su:
0 3
1 2
```

## Primer 3

```
INTERAKCIJA SA PROGRAMOM:
Unesite broj vrsta i
broj kolona matrice:
5 -3
Greska: neispravan unos.
```

**Zadatak 2.4.7** Napisati funkciju koja formira niz  $b_0, b_1, \dots, b_{n-1}$  od matrice  $n \times m$  tako što element niza  $b_i$  izračunava kao srednju vrednost elemenata  $i$ -te vrste matrice. Napisati program koji za učitano matricu celih brojeva ispisuje dobijeni niz. Pretpostaviti da je maksimalna dimenzija matrice  $50 \times 50$ . U slučaju neispravnog unosa, ispisati odgovarajuću poruku o grešci.

## Primer 1

```
INTERAKCIJA SA PROGRAMOM:
Unesite broj vrsta i
broj kolona matrice:
4 5
Unesite elemente matrice:
1 1 2 1 3
0 8 1 9 0
1 1 1 0 0
0 3 0 2 2
Dobijeni niz je:
1.6 3.6 0.6 1.4
```

## Primer 2

```
INTERAKCIJA SA PROGRAMOM:
Unesite broj vrsta i
broj kolona matrice:
3 4
Unesite elemente matrice:
7 10 12 20
-1 -3 1 7
0 -47 2 0
Dobijeni niz je:
12.25 1 -11.25
```

## Primer 3

```
INTERAKCIJA SA PROGRAMOM:
Unesite broj vrsta i
broj kolona matrice:
51 13
Greska: neispravan unos.
```

**Zadatak 2.4.8** Relacija se može predstaviti kvadratnom matricom nula i jedinica na sledeći način: element  $i$  je u relaciji sa elementom  $j$  ukoliko se u preseku  $i$ -te vrste i  $j$ -te kolone nalazi jedinica, a nije u relaciji ukoliko se tu nalazi nula. Napisati funkcije:

- `int reflektivna(int a[][MAKS], int n)` kojom se za relaciju zadatu matricom  $a$  dimenzije  $n \times n$  ispituje da li je reflektivna;
- `int simetricna(int a[][MAKS], int n)` kojom se za relaciju zadatu matricom  $a$  dimenzije  $n \times n$  ispituje da li je simetrična;
- `int tranzitivna(int a[][MAKS], int n)` kojom se za relaciju zadatu matricom  $a$  ispituje dimenzije  $n \times n$  da li je tranzitivna;

- (d) `int ekvivalencija(int a[][MAKS], int n)` kojom se za relaciju zadatu matricom  $a$  dimenzije  $n \times n$  ispituje da li je relacija ekvivalencije.

Napisati program koji za učitanu dimenziju  $n$  i kvadratnu matricu dimenzije  $n \times n$  ispisuje osobine odgovarajuće relacije. Pretpostaviti da je maksimalna dimenzija matrice  $50 \times 50$  i da matrica za vrednosti elemenata može imati samo nule i jedinice. U slučaju neispravnog unosa, ispisati odgovarajuću poruku o grešci.

### Primer 1

```
INTERAKCIJA SA PROGRAMOM:
Unesite broj vrsta matrice:
4
Unesite elemente matrice:
1 0 0 0
0 1 1 0
0 0 1 0
0 0 0 0
Relacija nije refleksivna.
Relacija nije simetrična.
Relacija jeste tranzitivna.
Relacija nije ekvivalencija.
```

### Primer 2

```
INTERAKCIJA SA PROGRAMOM:
Unesite broj vrsta matrice:
4
Unesite elemente matrice:
1 1 0 0
1 1 1 0
0 1 1 0
0 0 0 1
Relacija jeste refleksivna.
Relacija jeste simetrična.
Relacija nije tranzitivna.
Relacija nije ekvivalencija.
```

### Primer 3

```
INTERAKCIJA SA PROGRAMOM:
Unesite broj vrsta matrice:
54
Greska: neispravan unos.
```

**Zadatak 2.4.9** Data je kvadratna matrica dimenzije  $n \times n$ .

- Napisati funkciju `float trag(float a[][MAKS], int n)` koja računa trag matrice, odnosno zbir elemenata na glavnoj dijagonali matrice.
- Napisati funkciju `float suma_sporredna(float a[][MAKS], int n)` koja računa zbir elemenata na sporednoj dijagonali matrice.
- Napisati funkciju `float suma_iznad(float a[][MAKS], int n)` koja određuje sumu elemenata iznad glavne dijagonale.
- Napisati funkciju `float suma_ispod(float a[][MAKS], int n)` koja određuje sumu elemenata ispod sporedne dijagonale matrice.

Napisati program koji za učitanu matricu realnih brojeva ispisuje na tri decimale trag matrice, sumu na sporednoj dijagonali, sumu iznad glavne dijagonale i sumu elemenata ispod sporedne dijagonale. Pretpostaviti da je maksimalna dimenzija matrice  $50 \times 50$ . U slučaju neispravnog unosa, ispisati odgovarajuću poruku o grešci.

### Primer 1

```
INTERAKCIJA SA PROGRAMOM:
Unesite broj vrsta matrice: 4
Unesite elemente matrice:
6 12.08 -1 20.5
8 90 -33.4 19.02
7.02 5 -20 14.5
8.8 -1 3 -22.8
```

### Primer 1 (nastavak)

```
Trag: 53.20
Suma na sporednoj dijagonali: 0.90
Suma iznad glavne dijagonale: 31.70
Suma ispod sporedne dijagonale: -7.28
```

## Primer 2

```

INTERAKCIJA SA PROGRAMOM:
Unesite broj vrsta matrice: 5
Unesite elemente matrice:
1 2 3 5 5
7 8 9 0 1
6 4 3 2 2
8 9 1 3 4
0 3 1 8 6

```

## Primer 2 (nastavak)

```

Trag: 21.00
Suma na sporednoj dijagonali: 17.00
Suma iznad glavne dijagonale: 33.00
Suma ispod sporedne dijagonale: 31.00

```

**Zadatak 2.4.10** Kvadratna matrica je donje trougaona ako se u gornjem trouglu (iznad glavne dijagonale, ne uključujući dijagonalu) nalaze sve nule. Napisati program koji za učitane kvadratnu matricu proverava da li je ona donje trougaona i ispisuje odgovarajuću poruku. Pretpostaviti da je maksimalna dimenzija matrice  $100 \times 100$ . U slučaju neispravnog unosa, ispisati odgovarajuću poruku o grešci.

## Primer 1

```

INTERAKCIJA SA PROGRAMOM:
Unesite broj vrsta matrice:
5
Unesite elemente matrice:
-1 0 0 0 0
2 10 0 0 0
0 1 5 0 0
7 8 20 14 0
-23 8 5 1 11
Matrica jeste donje
trougaona.

```

## Primer 2

```

INTERAKCIJA SA PROGRAMOM:
Unesite broj vrsta matrice:
3
Unesite elemente matrice:
2 -2 1
1 2 2
2 1 -2
Matrica nije donje
trougaona.

```

## Primer 3

```

INTERAKCIJA SA PROGRAMOM:
Unesite broj vrsta matrice:
200
Greska: neispravan unos.

```

**Zadatak 2.4.11** Napisati program koji za učitane celobrojnu kvadratnu matricu ispisuje redni broj kolone koja ima najveći zbir elemenata. Pretpostaviti da je maksimalna dimenzija matrice  $50 \times 50$ . U slučaju neispravnog unosa, ispisati odgovarajuću poruku o grešci.

## Primer 1

```

INTERAKCIJA SA PROGRAMOM:
Unesite broj vrsta matrice:
3
Unesite elemente matrice:
1 2 3
7 3 4
5 3 1
Indeks kolone je: 0

```

## Primer 2

```

INTERAKCIJA SA PROGRAMOM:
Unesite broj vrsta matrice:
4
Unesite elemente matrice:
7 8 9 10
7 6 11 4
3 1 2 -2
8 3 9 9
Indeks kolone je: 2

```

## Primer 3

```

INTERAKCIJA SA PROGRAMOM:
Unesite broj vrsta matrice:
104
Greska: neispravan unos.

```

**Zadatak 2.4.12** Napisati program koji za učitane kvadratnu matricu realnih brojeva izračunava i ispisuje na dve decimale razliku između zbira elemenata

## 2 Napredni tipovi podataka

gornjeg trougla i zbira elemenata donjeg trougla matrice. Gornji trougao čine svi elementi matrice koji su iznad glavne i sporedne dijagonale (ne računajući dijagonale), a donji trougao čine svi elementi ispod glavne i sporedne dijagonale (ne računajući dijagonale). Pretpostaviti da je maksimalna dimenzija matrice  $50 \times 50$ . U slučaju neispravnog unosa, ispisati odgovarajuću poruku o grešci.

### Primer 1

```
INTERAKCIJA SA PROGRAMOM:
Unesite broj vrsta matrice:
3
Unesite elemente matrice:
2 3.2 4
7 8.8 1
2.3 1 1
Razlika je: 2.20
```

### Primer 2

```
INTERAKCIJA SA PROGRAMOM:
Unesite broj vrsta matrice:
5
Unesite elemente matrice:
2.3 1 12 8 -20
4 -8.2 7 14.5 19
1 -2.5 9 11 33
3 4.3 -5.7 2 8
9 56 1.08 7 5.5
Razlika je: -30.38
```

### Primer 3

```
INTERAKCIJA SA PROGRAMOM:
Unesite broj vrsta matrice:
52
Greska: neispravan unos.
```

**Zadatak 2.4.13** Napisati program koji za učitane celobrojnu matricu dimenzije  $m \times n$  i uneta dva broja  $p$  i  $k$  ( $p \leq m$ ,  $k \leq n$ ) ispisuje sume svih podmatrica dimenzije  $p \times k$  unete matrice. Pretpostaviti da je maksimalna dimenzija matrice  $50 \times 50$ . U slučaju neispravnog unosa, ispisati odgovarajuću poruku o grešci.

### Primer 1

```
INTERAKCIJA SA PROGRAMOM:
Unesite broj vrsta i broj kolona matrice:
3 4
Unesite elemente matrice:
1 2 3 4
5 6 7 8
9 10 11 12
Unesite dva cela broja: 3 3
Sume podmatrica su: 54 63
```

### Primer 2

```
INTERAKCIJA SA PROGRAMOM:
Unesite broj vrsta i broj kolona matrice:
3 4
Unesite elemente matrice:
1 2 3 4
5 6 7 8
9 10 11 12
Unesite dva cela broja: 2 3
Sume podmatrica su: 24 30 48 54
```

### Primer 3

```
INTERAKCIJA SA PROGRAMOM:
Unesite broj vrsta i broj kolona matrice:
5 3
Unesite elemente matrice:
1 1 2
5 0 2
7 8 9
1 2 4
0 1 1
Unesite dva cela broja: 2 2
Sume podmatrica su: 7 5 20 19 18 23 4 8
```

### Primer 4

```
INTERAKCIJA SA PROGRAMOM:
Unesite broj vrsta i broj kolona matrice:
-3 200
Greska: neispravan unos.
```

**Zadatak 2.4.14** Napisati program koji za učitane celobrojnu kvadratnu matricu ispituje da li su njeni elementi po kolonama, vrstama i dijagonalama (glavnoj i sporednoj) sortirani strogo rastuće. Pretpostaviti da je maksimalna dimenzija matrice  $50 \times 50$ . U slučaju neispravnog unosa, ispisati odgovarajuću

poruku o grešci.

#### Primer 1

```
INTERAKCIJA SA PROGRAMOM:
Unesite broj vrsta matrice: 2
Unesite elemente matrice:
6 9
4 10
Elementi nisu sortirani po kolonama.
Elementi su sortirani po vrstama.
Elementi nisu sortirani po dijagonalama.
```

#### Primer 3

```
INTERAKCIJA SA PROGRAMOM:
Unesite broj vrsta matrice: 4
Unesite elemente matrice:
5 5 7 9
6 10 11 13
8 12 14 15
13 15 16 20
Elementi su sortirani po kolonama.
Elementi nisu sortirani po vrstama.
Elementi su sortirani po dijagonalama.
```

#### Primer 2

```
INTERAKCIJA SA PROGRAMOM:
Unesite broj vrsta matrice: 3
Unesite elemente matrice:
1 2 3
4 5 6
7 8 9
Elementi su sortirani po kolonama.
Elementi su sortirani po vrstama.
Elementi su sortirani po dijagonalama.
```

#### Primer 4

```
INTERAKCIJA SA PROGRAMOM:
Unesite broj vrsta matrice: 1
Unesite elemente matrice:
5
Elementi su sortirani po kolonama.
Elementi su sortirani po vrstama.
Elementi su sortirani po dijagonalama.
```

**Zadatak 2.4.15** Napisati program koji za učitane celobrojnu kvadratnu matricu ispituje da li su zbrojevi elemenata njenih kolona uređeni u strogo rastućem poretku. Pretpostaviti da je maksimalna dimenzija matrice  $10 \times 10$ . U slučaju neispravnog unosa, ispisati odgovarajuću poruku o grešci.

#### Primer 1

```
INTERAKCIJA SA PROGRAMOM:
Unesite broj vrsta matrice: 4
Unesite elemente matrice:
1 0 0 0
0 0 1 0
0 0 0 1
0 1 0 0
Sume nisu uredjene strogo rastuce.
```

#### Primer 3

```
INTERAKCIJA SA PROGRAMOM:
Unesite broj vrsta matrice: 3
Unesite elemente matrice:
2 -2 1
1 2 2
2 1 -2
Sume nisu uredjene strogo rastuce.
```

#### Primer 2

```
INTERAKCIJA SA PROGRAMOM:
Unesite broj vrsta matrice: 3
Unesite elemente matrice:
1 2 3
4 5 6
7 8 9
Sume jesu uredjene strogo rastuce.
```

#### Primer 4

```
INTERAKCIJA SA PROGRAMOM:
Unesite broj vrsta matrice: 5
Unesite elemente matrice:
-1 0 3 0 20
0 0 0 10 0
0 0 -1 0 0
0 1 0 0 0
0 0 0 0 -1
Sume jesu uredjene strogo rastuce.
```

**Zadatak 2.4.16** Matrica je *ortonormirana* ako je vrednost skalarnog proizvoda svakog para različitih vrsta jednak nuli, a vrednost skalarnog proizvoda

## 2 Napredni tipovi podataka

vrste sa samom sobom jednak jedinici. Napisati program koji za unetu celobrojnu kvadratnu matricu proverava da li je ortonormirana. Pretpostaviti da je maksimalna dimenzija matrice  $50 \times 50$ . U slučaju neispravnog unosa, ispisati odgovarajuću poruku o grešci. NAPOMENA: *Skalarni proizvod vektora*  $a = (a_1, a_2, \dots, a_n)$  i  $b = (b_1, b_2, \dots, b_n)$  je  $a_1 \cdot b_1 + a_2 \cdot b_2 + \dots + a_n \cdot b_n$ .

### Primer 1

```
INTERAKCIJA SA PROGRAMOM:
Unesite broj vrsta matrice: 4
Unesite elemente matrice:
1 0 0 0
0 0 1 0
0 0 0 1
0 1 0 0
Matrica jeste ortonormirana.
```

### Primer 2

```
INTERAKCIJA SA PROGRAMOM:
Unesite broj vrsta matrice: 3
Unesite elemente matrice:
1 2 3
4 5 6
7 8 9
Matrica nije ortonormirana.
```

### Primer 3

```
INTERAKCIJA SA PROGRAMOM:
Unesite broj vrsta matrice: 3
Unesite elemente matrice:
2 -2 1
1 2 2
2 1 -2
Matrica nije ortonormirana.
```

### Primer 4

```
INTERAKCIJA SA PROGRAMOM:
Unesite broj vrsta matrice: 5
Unesite elemente matrice:
-1 0 0 0 0
0 0 0 1 0
0 0 -1 0 0
0 1 0 0 0
0 0 0 0 -1
Matrica jeste ortonormirana.
```

**Zadatak 2.4.17** Kvadratna matrica je *magični kvadrat* ako su sume elemenata u svim vrstama i kolonama jednake. Napisati program koji proverava da li je data celobrojna kvadratna matrica magični kvadrat i ispisuje odgovarajuću poruku na standardni izlaz. Pretpostaviti da je maksimalna dimenzija matrice  $50 \times 50$ . U slučaju neispravnog unosa, ispisati odgovarajuću poruku o grešci.

### Primer 1

```
INTERAKCIJA SA PROGRAMOM:
Unesite broj vrsta matrice: 4
Unesite elemente matrice:
1 5 3 1
2 1 2 5
3 2 2 3
4 2 3 1
Matrica jeste magicni kvadrat.
```

### Primer 2

```
INTERAKCIJA SA PROGRAMOM:
Unesite broj vrsta matrice: 3
Unesite elemente matrice:
1 2 3
4 5 6
-1 3 3
Matrica nije magicni kvadrat.
```

\* **Zadatak 2.4.18** Napisati program koji učitava celobrojnu kvadratnu matricu i ispisuje elemente matrice u grupama koje su paralelne sa njenom sporednom dijagonalom, počevši od gornjeg levog ugla. Pretpostaviti da je maksimalna dimenzija matrice  $100 \times 100$ . U slučaju neispravnog unosa, ispisati odgovarajuću

poruku o grešci.

#### Primer 1

```
INTERAKCIJA SA PROGRAMOM:
Unesite broj vrsta matrice:
3
Unesite elemente matrice:
1 2 3
4 5 6
7 8 9
Ispis je:
1
2 4
3 5 7
6 8
9
```

#### Primer 2

```
INTERAKCIJA SA PROGRAMOM:
Unesite broj vrsta matrice:
5
Unesite elemente matrice:
7 -8 1 2 3
90 11 0 5 4
12 -9 14 23 8
80 6 88 17 62
-22 10 44 57 -200
Ispis je:
7
-8 90
1 11 12
2 0 -9 80
3 5 14 6 -22
4 23 88 10
8 17 44
62 57
-200
```

#### Primer 3

```
INTERAKCIJA SA PROGRAMOM:
Unesite broj vrsta matrice:
-5
Greska: neispravan unos.
```

**\* Zadatak 2.4.19** Napisati funkciju `void mnozenje(int a[][MAKS], int m, int n, int b[][MAKS], int k, int t, int c[][MAKS])` koja računa matricu  $c$  kao proizvod matrica  $a$  i  $b$ . Dimenzija matrice  $a$  je  $n \times m$ , a dimenzija matrice  $b$  je  $k \times t$ . Napisati program koji ispisuje proizvod učitanih matrica. Pretpostaviti da je maksimalna dimenzija matrica  $50 \times 50$ . Ukoliko množenje matrica nije moguće ili je došlo do greške prilikom unosa podataka, ispisati odgovarajuću poruku o grešci.

#### Primer 1

```
INTERAKCIJA SA PROGRAMOM:
Unesite broj vrsta i broj kolona matrice a:
3 4
Unesite elemente matrice:
1 2 8 9
-4 5 2 3
7 6 4 10
Unesite broj vrsta i broj kolona matrice b:
4 2
Unesite elemente matrice:
11 5
6 7
8 9
0 -3
Rezultat mnozenja je:
87 64
2 24
145 83
```

#### Primer 2

```
INTERAKCIJA SA PROGRAMOM:
Unesite broj vrsta i broj kolona matrice a:
5 2
Unesite elemente matrice:
1 7
9 0
-10 2
92 3
14 -8
Unesite broj vrsta i broj kolona matrice b:
2 4
Unesite elemente matrice:
7 8 9 10
-11 2 34 78
Rezultat mnozenja je:
-70 22 247 556
63 72 81 90
-92 -76 -22 56
611 742 930 1154
186 96 -146 -484
```

### Primer 3

```
INTERAKCIJA SA PROGRAMOM:
Unesite broj vrsta i broj kolona matrice a:
3 4
Unesite elemente matrice:
1 2 8 9
-4 5 2 3
7 6 4 10
Unesite broj vrsta i broj kolona matrice b:
5 2
Množenje matrica nije moguće.
```

### Primer 4

```
INTERAKCIJA SA PROGRAMOM:
Unesite broj vrsta i broj kolona matrice a:
-3 4
Greska: neispravan unos.
```

\* **Zadatak 2.4.20** Element matrice naziva se *sedlo* ako je istovremeno najmanji u svojoj vrsti, a najveći u svojoj koloni. Napisati program koji ispisuje indekse i vrednosti onih elemenata matrice realnih brojeva koji su sedlo. Maksimalna dimenzija matrice je  $50 \times 50$ . U slučaju neispravnog unosa, ispisati odgovarajuću poruku o grešci.

### Primer 1

```
INTERAKCIJA SA PROGRAMOM:
Unesite broj vrsta i
broj kolona matrice:
2 3
Unesite elemente matrice:
1 2 3
0 5 6
Sedlo: 0 0 1
```

### Primer 2

```
INTERAKCIJA SA PROGRAMOM:
Unesite broj vrsta i
broj kolona matrice:
3 3
Unesite elemente matrice:
10 3 20
15 5 100
30 -1 200
Sedlo: 1 1 5
```

### Primer 2

```
INTERAKCIJA SA PROGRAMOM:
Unesite broj vrsta i
broj kolona matrice:
3 -3
Greska: neispravan unos.
```

\* **Zadatak 2.4.21** Napisati program koji ispisuje elemente matrice celih brojeva u spiralnom redosledu počevši od gornjeg levog ugla krećući se u smeru kazaljke na satu. Maksimalna dimenzija matrice je  $50 \times 50$ . U slučaju neispravnog unosa, ispisati odgovarajuću poruku o grešci.

### Primer 1

```
INTERAKCIJA SA PROGRAMOM:
Unesite broj vrsta i
broj kolona matrice:
3 3
Unesite elemente matrice:
1 2 3
4 5 6
7 8 9
Ispis je:
1 2 3 6 9 8 7 4 5
```

### Primer 2

```
INTERAKCIJA SA PROGRAMOM:
Unesite broj vrsta i
broj kolona matrice:
5 7
Unesite elemente matrice:
7 -8 1 2 3 -54 87
90 11 0 5 4 9 18
12 -9 14 23 8 -22 74
80 6 88 17 62 38 41
-22 10 44 57 -200 39 55
Ispis je:
7 -8 1 2 3 -54 87 18 74 41 55
39 -200 57 44 10 -22 80 12 90
11 0 5 4 9 -22 38 62 17 88 6
-9 14 23 8
```



\* **Zadatak 2.4.22** Matrica  $a$  se sadrži u matrici  $b$  ukoliko postoji podmatrica matrice  $b$  identična matrici  $a$ . Napisati program koji za dve učitane matrice celih brojeva proverava da li se druga matrica sadrži u prvoj učitanoj matrici. Maksimalna dimenzija matrica je  $50 \times 50$ . U slučaju neispravnog unosa, ispisati odgovarajuću poruku o grešci.

### Primer 1

```
INTERAKCIJA SA PROGRAMOM:
Unesite broj vrsta i broj kolona matrice A:
3 4
Unesite elemente matrice:
1 2 8 9
-4 5 2 3
7 6 4 10
Unesite broj vrsta i broj kolona matrice B:
2 2
Unesite elemente matrice:
2 3
4 10
Druga matrica je sadrzana u prvoj matrici.
```

### Primer 2

```
INTERAKCIJA SA PROGRAMOM:
Unesite broj vrsta i broj kolona matrice A:
3 4
Unesite elemente matrice:
1 2 8 9
-4 5 2 3
7 6 4 10
Unesite broj vrsta i broj kolona matrice B:
2 2
Unesite elemente matrice:
2 8
6 4
Druga matrica nije sadrzana
u prvoj matrici.
```

### Primer 3

```
INTERAKCIJA SA PROGRAMOM:
Unesite broj vrsta i broj kolona matrice A:
5 5
Unesite elemente matrice:
7 -8 1 2 3
90 11 0 5 4
12 -9 14 23 8
80 6 88 17 62
-22 10 44 57 -200
Unesite broj vrsta i broj kolona matrice B:
3 4
Unesite elemente matrice:
90 11 0 5
12 -9 14 23
80 6 88 17
Druga matrica je sadrzana u prvoj matrici.
```

### Primer 4

```
INTERAKCIJA SA PROGRAMOM:
Unesite broj vrsta i broj kolona matrice A:
5 5
Unesite elemente matrice:
7 -8 1 2 3
90 11 0 5 4
12 -9 14 23 8
80 6 88 17 62
-22 10 44 57 -200
Unesite broj vrsta i broj kolona matrice B:
53 4
Greska: neispravan unos.
```

### 2.5 Strukture

**Zadatak 2.5.1** Definirati strukturu kojom se opisuje kompleksan broj. Napisati funkcije koje izračunavaju zbir, razliku, proizvod i količnik dva kompleksna broja. Napisati program koji za učitana dva kompleksna broja ispisuje vrednost zbira, razlike, proizvoda i količnika. U slučaju neispravnog unosa, ispisati odgovarajuću poruku o grešci.

*Primer 1*

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

**Zadatak 2.5.2** Definirati strukturu kojom se opisuje razlomak. Napisati funkcije koje izračunavaju zbir i proizvod dva razlomka. Napisati program koji za uneti ceo broj  $n$  i unetih  $n$  razlomaka ispisuje njihov ukupan zbir i proizvod. U slučaju neispravnog unosa, ispisati odgovarajuću poruku o grešci.

*Primer 1*

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

*Primer 2*

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

**Zadatak 2.5.3** Zimi su prehlade česte i treba unositi više vitamina C. Struktura *Vocka* sadrži ime voćke i količinu vitamina C u miligramima (realan broj). Napisati funkcije:

- `int ucitaj(Vocka niz[])` koja učitava voćke sa standardnog ulaza sve do kraja ulaza i kao povratnu vrednost vraća broj učitanih voćki;
- `Vocka vocka_sa_najvise_vitamina(Vocka niz[], int n)` koja pronalazi

voćku koja ima najviše vitamina C.

Napisati program koji učitava podatke o voćkama i ispisuje ime voćke sa najviše vitamina C. Pretpostaviti da broj voćki neće biti veći od 50, kao i da je ime voćke niska od najviše 20 karaktera. U slučaju neispravnog unosa, ispisati odgovarajuću poruku o grešci.

#### Primer 1

```
INTERAKCIJA SA PROGRAMOM:
Unesite ime vocke i njenu kolicinu vitamina C: jabuka 4.6
Unesite ime vocke i njenu kolicinu vitamina C: limun 83.5
Unesite ime vocke i njenu kolicinu vitamina C: kivi 71
Unesite ime vocke i njenu kolicinu vitamina C: banana 8.7
Unesite ime vocke i njenu kolicinu vitamina C: pomorandza 70.8
Unesite ime vocke i njenu kolicinu vitamina C:
Voce sa najvise vitamina C je: limun
```

**Zadatak 2.5.4** Definirati strukturu `Grad` koja sadrži ime grada i njegovu prosečnu temperaturu u toku decembra. Napisati funkcije:

- `void ucitaj(Grad gradovi[], int n)` koja učitava sa standardnog ulaza podatke o  $n$  gradova.
- `void ispisi(Grad gradovi[], int n)` koja ispisuje podatke o gradovima koji imaju idealnu temperaturu za klizanje: od 3 do 8 stepeni celzijusa.

Napisati program koji učitava imena  $n$  gradova i njihove prosečne temperature, a zatim ispisuje imena gradova sa idealnom temperaturom za klizanje. Pretpostaviti da je maksimalan broj gradova 50 i da je maksimalna dužina imena grada 20 karaktera. U slučaju neispravnog unosa, ispisati odgovarajuću poruku o grešci.

#### Primer 1

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

#### Primer 2

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

**Zadatak 2.5.5** Definirati strukturu `ParReci` koja sadrži reč na srpskom jeziku i odgovarajući prevod na engleski jezik. Napisati program koji do kraja

## 2 Napredni tipovi podataka

ulaza učitava sve parove reči, a potom za rečenicu koja se zadaje u jednoj liniji ispisuje prevod. Ako je reč u rečenici nepoznata umesto nje ispisati nisku zvezdica čija dužina odgovara dužini nepoznate reči. Pretpostaviti da je maksimalna dužina reči 50 karaktera, maksimalan broj parova reči 100, a maksimalna dužina rečenice 100 karaktera.

### Primer 1

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

### Primer 2

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

**Zadatak 2.5.6** Statistički zavod Srbije istražuje kako rade obdaništa u Srbiji. Za svako obdanište poznat je spisak koji sadrži broj dece u grupi, a zatim i ocene koje je svako dete dalo o radu obdaništa. Definirati strukturu **Dete** koja sadrži polja *pol* (*m* ili *z*), broj godina (od 3 do 6) i ocenu koju je dete dalo radu obdaništa (od 1 do 5). Napisati program koji učitava broj dece u grupi, a zatim i informacije o svakom detetu. Ispisati, na tri decimale, prosečnu ocenu koje je obdanište dobilo od dece sa unetim polom i brojem godina. Pretpostaviti da je maksimalan broj dece u obdaništu 200. U slučaju neispravnog unosa, ispisati odgovarajuću poruku o grešci.

### Primer 1

```
INTERAKCIJA SA PROGRAMOM:
Unesite broj dece u grupi: 5
Unesite podatke za svako dete (pol,
broj godina i ocenu):
m 3 5
z 3 4
m 4 2
m 5 4
m 3 4
Unesite pol i broj godina za
statistiku: m 3
Prosečna ocena je: 4.500.
```

### Primer 2

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

## Primer 3

```

INTERAKCIJA SA PROGRAMOM:
Unesite broj dece u grupi: 15
Unesite podatke za svako dete (pol,
broj godina i ocenu):
m 3 2
z 7 5
Greska: neispravan broj godina.

```

## Primer 4

```

INTERAKCIJA SA PROGRAMOM:
Unesite broj dece u grupi: 2
Unesite podatke za svako dete (pol,
broj godina i ocenu):
m 3 2
z 3 5
Unesite pol i broj godina za
statistiku: h 5
Greska: neispravan pol.

```

**Zadatak 2.5.7** Definirati strukturu kojom se opisuje student. Student se opisuje svojim imenom i prezimenom, smerom (R, I, V, N, T, M) i prosečnom ocenom. Napisati program koji učitava podatke o  $n$  studenata, a zatim i informaciju o smeru i ispisuje imena i prezimena onih studenta koji su sa datog smera, kao i podatke studenta koji ima najveći prosek. Ako ima više takvih studenata ispisati podatke o svima. Pretpostaviti da je maksimalan broj studenata 2000, a maksimalna dužina imena i prezimena po 30 karaktera. U slučaju neispravnog unosa, ispisati odgovarajuću poruku o grešci.

## Primer 1

```

INTERAKCIJA SA PROGRAMOM:
Unesite broj studenata: 5
Unesite podatke o studentima:
0. student: Kocic Marija R 9.14
1. student: Tanja Mratinkovic R 7.88
2. student: Mihailo Simic N 8.44
3. student: Milena Medar I 9.14
4. student: Ljubica Mihic N 9.00
Unesite smer: R
Studenti sa R smerom:
Kocic Marija
Tanja Mratinkovic
-----
Svi studenti koji imaju maksimalni prosek:
Kocic Marija, R, 9.14
Milena Medar, I, 9.14

```

## Primer 2

```

INTERAKCIJA SA PROGRAMOM:
Unesite broj studenata: 4
Unesite podatke o studentima:
0. student: Djordje Lazarevic N 9.05
1. student: Minja Peric W 7.70
Greska: neispravan unos smera.

```

**Zadatak 2.5.8** Definirati strukturu Djaka koja sadrži ime đaka i 9 ocena (ocene su celi brojevi od 1 do 5). Napisati program koji učitava podatke o đacima sve do kraja ulaza i na standardni izlaz ispisuje prvo imena nedovoljnih đaka, a zatim imena odličnih đaka. Đak je nedovoljan ako ima barem jednu jedinicu, a odličan ako ima prosek ocena veći ili jednak 4.5. Pretpostaviti da je maksimalna dužina imena đaka 20 karaktera, kao i da je maksimalan broj đaka 30. U slučaju neispravnog unosa, ispisati odgovarajuću poruku o grešci.

### Primer 1

```
INTERAKCIJA SA PROGRAMOM:
Unesite podatke o djaku:
Maja 4 5 2 3 4 4 3 3 4
Unesite podatke o djaku:
Nikola 5 4 5 5 5 4 4 5 5
Unesite podatke o djaku:
Jasmina 2 2 1 1 2 3 3 1 3
Unesite podatke o djaku:
Pera 5 4 5 3 5 5 1 5 5
Unesite podatke o djaku:
Pavle 4 3 2 4 3 2 4 3 2
Unesite podatke o djaku:

NEDOVOLJNI: Jasmina Pera
ODLICNI: Nikola
```

### Primer 2

```
INTERAKCIJA SA PROGRAMOM:
Unesite podatke o djaku:
Uros 3 4 2 3 4 2 3 4 4
Unesite podatke o djaku:
Nebojsa 4 5 5 5 4 5 5 5 5
Unesite podatke o djaku:
Sreten 2 3 2 4 5 4 4 4 2
Unesite podatke o djaku:

NEDOVOLJNI:
ODLICNI: Nebojsa
```

### Primer 3

```
INTERAKCIJA SA PROGRAMOM:
Unesite podatke o djaku:
Mirko 2 3 4 4 4 3 3 3 4
Unesite podatke o djaku:
Mihailo 2 3 10 5 5 2 3 4 2
Greska: neispravna ocena.
```

**Zadatak 2.5.9** Definirati strukturu *Osoba* kojom se opisuje jedan unos u imenik. Za svaku osobu su dati podaci: ime, prezime i imejl adresa. Napisati program koji učitava ceo broj  $n$ , a zatim podatke o  $n$  osoba. Ispisati imena i prezimena svih osoba koje imaju imejl adresu koja se završava sa @gmail.com. Pretpostaviti da je maksimalan broj osoba 50, kao i da je maksimalna dužina imena osobe 20 karaktera, prezimena 30 karaktera, a imejl adrese 50 karaktera. U slučaju neispravnog unosa, ispisati odgovarajuću poruku o grešci. **NAPOMENA:** Može se smatrati da je svaka imejl adresa dobro zadana i sadrži samo jedno pojavljivanje znaka @.

### Primer 1

```
INTERAKCIJA SA PROGRAMOM:
Unesite broj osoba: 3
Unesite podatke o osobama
(ime, prezime i imejl adresu):
Dusko Dugousko dusko@yahoo.com
Pink Panter panter@gmail.com
Pera Detlic pd@gmail.com
Vlasnici gmail naloga su:
Pink Panter
Pera Detlic
```

### Primer 2

```
INTERAKCIJA SA PROGRAMOM:
Unesite broj osoba: 3
Unesite podatke o osobama
(ime, prezime i imejl adresu):
Homer Simpson homer@yahoo.com
Mardz Simpson mardz@matf.bg.ac.rs
Nema vlasnika gmail naloga.
```

\* **Zadatak 2.5.10** Napisati program koji izračunava prosečnu cenu jedne potrošačke korpe. Potrošačka korpa se sastoji od broja kupljenih artikala i niza kupljenih artikala. Svaki artikal određen je svojim nazivom, količinom i cenom. Program treba da učitava broj potrošača  $n$ , zatim podatke za  $n$  potrošačkih korpi i da na osnovu učitanih podataka izračuna prosečnu cenu potrošačke korpe. Program ispisuje na dve decimale račune svake potrošačke korpe i na kraju ispisuje prosečnu cenu potrošačke korpe. Pretpostaviti da je maksimalan broj potrošačkih

korpi 100, maksimalan broj artikala u korpi 20 i da naziv svakog artikla sadrži maksimalno 30 karaktera. U slučaju neispravnog unosa, ispisati odgovarajuću poruku o grešci.

### Primer 1

```
INTERAKCIJA SA PROGRAMOM:
Unesite broj potrosackih korpi: 3
Unesite podatke o korpi:
Broj artikala: 4
Unesite artikal (naziv, kolicinu i cenu): jabuke 10 22.4
Unesite artikal (naziv, kolicinu i cenu): dezodorans 1 120.99
Unesite artikal (naziv, kolicinu i cenu): C_supa 3 36.56
Unesite artikal (naziv, kolicinu i cenu): sunka 1 230.99
Unesite podatke o korpi:
Broj artikala: 2
Unesite artikal (naziv, kolicinu i cenu): Jafa_keks 1 55.78
Unesite artikal (naziv, kolicinu i cenu): Najlepse_zelje 1 62.99
Unesite podatke o korpi:
Broj artikala: 3
Unesite artikal (naziv, kolicinu i cenu): prasak_zav_1 1199.99
Unesite artikal (naziv, kolicinu i cenu): omeksivac 1 279.99
Unesite artikal (naziv, kolicinu i cenu): protiv_kamenca 1 699.99

Korpa 0:
jabuke 10 22.40
dezodorans 1 120.99
C_supa 3 36.56
sunka 1 230.99
-----
ukupno: 685.66

Korpa 1:
Jafa_keks 1 55.78
Najlepse_zelje 1 62.99
-----
ukupno: 118.77

Korpa 2:
pracak_zav_1 1199.99
omeksivac 1 279.99
protiv_kamenca 1 699.99
-----
ukupno: 2179.97

Prosečna cena potrosacke korpe: 994.80
```

**Zadatak 2.5.11** Definirati strukturu *Lopta* sa poljima *poluprecnik* (ceo broj u centimetrima) i *boja* (enumeracioni tip koji uključuje plavu, žutu, crvenu i zelenu boju). Napisati funkcije:

- `void ucitaj(Lopta niz[], int n)` koja učitava podatke o  $n$  lopti u niz.
- `double ukupna_zapremina(Lopta niz[], int n)` koja računa ukupnu zapreminu svih lopti.

## 2 Napredni tipovi podataka

- (c) `int broj_crvenih(Lopta niz[], int n)` koja prebrojava koliko ima crvenih lopti u nizu.

Napisati program koji učitava informacije o  $n$  lopti i ispisuje ukupnu zapreminu i broj crvenih lopti. Pretpostaviti da je maksimalan broj lopti 50. U slučaju neispravnog unosa, ispisati odgovarajuću poruku o grešci.

### Primer 1

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

### Primer 2

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

### Primer 3

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

**Zadatak 2.5.12** Napisati program za predstavljanje poligona i izračunavanje dužine njegovih stranica i obima.

- (a) Definisati strukturu `Tacka` kojom se opisuje tačka dekartovske ravni čije su  $x$  i  $y$  koordinate podaci tipa `double`.
- (b) Definisati funkciju `double rastojanje(const Tacka *A, const Tacka *B)` koja izračunava rastojanje između dve tačke.
- (c) Definisati funkciju `int ucitaj_poligon(Tacka poligon[], int n)` koja učitava maksimalno  $n$  puta po dve vrednosti tipa `double` (koje predstavljaju koordinate temena poligona) i upisuje ih u zadati niz tačaka. Funkcija vraća broj uspešno učitanih tačaka.



- (d) Definirati funkciju `double obim_poligona(Tacka poligon[], int n)` koja izračunava obim poligona sa  $n$  temena u zadatom nizu. UPUTSTVO: *Prilikom računanja obima ne zaboraviti stranicu koja spaja poslednje i prvo teme.*
- (e) Definirati funkciju `double maksimalna_stranica(Tacka poligon[], int n)` koja izračunava dužinu najduže stranice poligona sa  $n$  temena u zadatom nizu.
- (f) Napisati funkciju `double povrsina_trougla(const Tacka *A, const Tacka *B, const Tacka *C)` koja izračunava površinu trougla čija su temena A, B i C.
- (g) Napisati funkciju `double povrsina_poligona(Tacka poligon[], int n)` koja izračunava površinu konveksnog poligona. UPUTSTVO: *Zadatak se može rešiti podelom poligona na trouglove i korišćenjem funkcije `povrsina_trougla`.*

Napisati program koji učitava poligon sa maksimalno  $n$  temena i za učitani poligon ispisuje na tri decimale obim, dužinu najduže stranice i površinu. Pretpostaviti da je uneti poligon konveksan. Poligon mora imati barem tri temena. Pretpostaviti da je maksimalan broj temena 1000. U slučaju neispravnog unosa, ispisati odgovarajuću poruku o grešci.

#### Primer 1

```
INTERAKCIJA SA PROGRAMOM:
Unesite maksimalan broj temena poligona: 10
Unesite temena poligona:
0 0
0 6
3 3
Obim poligona je 14.485.
Duzina maksimalne stranice je 6.000.
Povrsina poligona je 9.000.
```

#### Primer 3

```
INTERAKCIJA SA PROGRAMOM:
Unesite maksimalan broj temena poligona: 4
0 0
Greska: poligon mora imati bar tri tacke.
```

#### Primer 2

```
INTERAKCIJA SA PROGRAMOM:
Unesite maksimalan broj temena poligona: 10
Unesite temena poligona:
0 0
12 0
13 2
16 5
20 10
18 15
15 20
10 20
8 15
3 4
Obim poligona je 63.566.
Duzina maksimalne stranice je 12.083.
Povrsina poligona je 247.500.
```

\* **Zadatak 2.5.13** Definirati strukturu `Izraz` kojom se opisuje numerički izraz nad celim brojevima koji se sastoji od dva celobrojna operanda i numeričke operacije (sabiranje, oduzimanje, množenje ili celobrojno deljenje).

- (a) Napisati funkciju `int korektan_izraz(const Izraz *izraz)` koja ispituje da li je dati izraz korektno zadat i vraća jedinicu ako jeste, a nulu inače.

## 2 Napredni tipovi podataka

Podrazumeva se da je izraz korektno zadat ako je operacija  $+$ ,  $-$ ,  $*$  ili  $/$  i u slučaju deljenja drugi operand je različit od 0.

- (b) Napisati funkciju `int vrednost(const Izraz *izraz)` koja za dati izraz određuje vrednost izraza.
- (c) Napisati funkciju `void ucitaj(Izraz izrazi[], int n)` koja učitava izraze. Funkcija treba da učitava sa standardnog ulaza  $n$  izraza koji su zadati prefiksno — prvo operacija, a potom dva operanda.

Napisati program koji učitava prirodan broj  $n$ , a zatim  $n$  izraza u prefiksnoj notaciji. Program treba da ispiše maksimalnu vrednost unetih izraza i sve izraze čija vrednost je manja od polovine maksimalne vrednosti. Pretpostaviti da je maksimalan broj izraza 100. U slučaju neispravnog unosa, ispisati odgovarajuću poruku o grešci.

### Primer 1

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

### Primer 3

```
INTERAKCIJA SA PROGRAMOM:
Unesite broj izraza: 3
Unesite izraze u prefiksnoj notaciji:
* 1 2
/ 3 0
Greska: deljenje nulom.
```

### Primer 2

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

\* **Zadatak 2.5.14** Definirati strukturu kojom se opisuje polinom. Polinom je dat svojim stepenom i realnim koeficijentima.

- (a) Napisati funkciju `int ucitaj(Polinom niz[])` koja sa standardnog ulaza učitava polinome sve do kraja ulaza. Polinomi su zadati stepenom i koeficijentima počevši od slobodnog člana. Funkcija kao povratnu vrednost vraća broj učitanih polinoma.

- (b) Napisati funkciju `void ispis(const Polinom *p)` koja ispisuje polinom stepena  $n$  sa koeficijentima  $k_0, k_1, \dots, k_n$  u obliku  $k_0 \pm k_1 * x \pm k_2 * x^2 \pm k_3 * x^3 \pm \dots \pm k_n * x^n$ . Na mesto znaka  $\pm$  zapisati odgovarajući znak,  $+$  ili  $-$ , u zavisnosti od znaka odgovarajućeg koeficijenta. Koeficijente ispisivati na dve decimale. Koeficijente koji su jednaki 0 ne ispisivati.
- (c) Napisati funkciju `void integral(const Polinom *p, Polinom *tekuci_integral)` koja za dati polinom `p` određuje njegov integral `tekuci_integral`. Za vrednost slobodnog člana integrala uzeti vrednost 0.

Napisati program koji učitava polinome do kraja ulaza i za svaki učitani polinom određuje i ispisuje njegov integral. Pretpostaviti da je maksimalan broj polinoma 100, a maksimalan stepen polinoma 10. U slučaju neispravnog unosa, ispisati odgovarajuću poruku o grešci.

#### Primer 1

```
INTERAKCIJA SA PROGRAMOM:
Unesite stepen: 3
Unesite koeficijente polinoma:
1 0 3 1
Unesite stepen: 4
Unesite koeficijente polinoma:
7 9 4 0 4
Unesite stepen:

Integrali su:
1.00*x + 1.00*x^3 + 0.25*x^4
7.00*x + 4.50*x^2 + 1.33*x^3 + 0.80*x^5
```

#### Primer 2

```
INTERAKCIJA SA PROGRAMOM:
Unesite stepen: 3
Unesite koeficijente polinoma:
1 0 -4 1
Unesite stepen: 2
Unesite koeficijente polinoma:
1 2 -3
Unesite stepen: 1
Unesite koeficijente polinoma:
0 -1
Unesite stepen:

Integrali su:
1.00*x - 1.33*x^3 + 0.25*x^4
1.00*x + 1.00*x^2 - 1.00*x^3
-0.50*x^2
```

Elektronsko izdanje (2019)

## 3

# Ulaz i izlaz programa

## 3.1 Argumenti komandne linije

**Zadatak 3.1.1** Napisati program koji ispisuje broj navedenih argumenata komandne linije, a zatim i same argumente i njihove redne brojeve.

*Primer 1*

```
POKRETANJE: ./a.out d1.txt 10 13.5 d2.txt
IZLAZ:
Broj argumenata je 5.
0: ./a.out
1: d1.txt
2: 10
3: 13.5
4: d2.txt
```

*Primer 2*

```
POKRETANJE: ./a.out
IZLAZ:
Broj argumenata je 1.
0: ./a.out
```

**Zadatak 3.1.2** Napisati program koji ispisuje zbir celobrojnih argumenata komandne linije. UPUTSTVO: *Koristiti funkciju atoi.*

*Primer 1*

```
POKRETANJE: ./a.out 5 ana 9 -2 11 4 +2
IZLAZ:
Zbir celobrojnih argumenata:
29
```

*Primer 2*

```
POKRETANJE: ./a.out a1 b1 1a 1b
IZLAZ:
Zbir celobrojnih argumenata:
0
```

*Primer 3*

```
POKRETANJE: ./a.out 33 1 @matf 44 22.56
IZLAZ:
Zbir celobrojnih argumenata:
78
```

**Zadatak 3.1.3** Napisati program koji na osnovu broja  $n$  koji se zadaje kao

### 3 Ulaz i izlaz programa

---

argument komandne linije, ispisuje cele brojeve iz intervala  $[-n, n]$ . U slučaju neispravnog pokretanja programa ispisati odgovarajuću poruku o grešci.

Primer 1	Primer 2	Primer 3
<pre>POKRETANJE: ./a.out 2 IZLAZ: -2 -1 0 1 2</pre>	<pre>POKRETANJE: ./a.out IZLAZ: Greska: neispravan poziv.</pre>	<pre>POKRETANJE: ./a.out 0 IZLAZ: 0</pre>

**Zadatak 3.1.4** Napisati program koji ispisuje argumente komandne linije koji počinju karakterom @.

Primer 1	Primer 2	Primer 3
<pre>POKRETANJE: ./a.out @ana @aca #zvezda IZLAZ: Argumenti koji pocinju sa @: @ana @aca</pre>	<pre>POKRETANJE: ./a.out sanke @zapad zujanje IZLAZ: Argumenti koji pocinju sa @: @zapad</pre>	<pre>POKRETANJE: ./a.out bundeva pomorandza IZLAZ: Nema argumenata koji pocinju sa @.</pre>

**Zadatak 3.1.5** Napisati program koji ispisuje broj argumenata komandne linije koji sadrže karakter @.

Primer 1	Primer 2	Primer 3
<pre>POKRETANJE: ./a.out pera@gmail.com @ IZLAZ: Rezultat: 2</pre>	<pre>POKRETANJE: ./a.out japan caj IZLAZ: 0</pre>	<pre>POKRETANJE: ./a.out IZLAZ: Rezultat: 0</pre>

**Zadatak 3.1.6** Napisati program koji proverava da li se među zadatim argumentima komandne linije nalaze barem dva ista. NAPOMENA: *Uzeti u obzir i naziv programa koji se pokreće.*

Primer 1	Primer 2
<pre>POKRETANJE: ./a.out ulaz.txt izlaz.txt ulaz.txt IZLAZ: Medju argumentima ima istih.</pre>	<pre>POKRETANJE: ./a.out srce pik tref tref IZLAZ: Medju argumentima ima istih.</pre>

### 3.1 Argumenti komandne linije

---

#### Primer 3

```
POKRETANJE:
./a.out Riba ribi grize rep.

IZLAZ:
Medju argumentima nema istih.
```

#### Primer 4

```
POKRETANJE:
./a.out

IZLAZ:
Medju argumentima nema istih.
```

**Zadatak 3.1.7** Napisati program koji ispisuje sve opcije koje su navedene u komandnoj liniji. Opcije su karakteri argumenata komandne linije koji za koje važi da počinju karakterom `-`.

#### Primer 1

```
POKRETANJE:
./a.out -rf in.txt

IZLAZ:
Opcije su: r f
```

#### Primer 2

```
POKRETANJE:
./a.out

IZLAZ:
Medju argumentima nema opcija.
```

#### Primer 3

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

IZLAZ:
Medju argumentima nema opcija.
```

#### Primer 4

```
POKRETANJE:
./a.out in.txt -l -n 10 -fi out.txt

IZLAZ:
Opcije su: l n f i
```

## 3.2 Datoteke

**Zadatak 3.2.1** Napisati program koji prepisuje sadržaj datoteke *ulaz.txt* u datoteku *izlaz.txt* karakter po karakter. U slučaju greške, na standardni izlaz za greške ispisati odgovarajuću poruku.

#### Primer 1

```
ULAZ.TXT
Danas je 21. mart.
To je prvi dan proleca.

IZLAZ.TXT
Danas je 21. mart.
To je prvi dan proleca.
```

#### Primer 2

```
ULAZ.TXT
Ispit iz Programiranja 1 je
zakazan za 10. jun.

IZLAZ.TXT
Ispit iz Programiranja 1 je
zakazan za 10. jun.
```

#### Primer 3

```
ULAZ.TXT NE POSTOJI
IZLAZ ZA GREŠKE:
Greska: neuspesno otvaranje
datoteke ulaz.txt
```

**Zadatak 3.2.2** Napisati program koji prepisuje svaki treći karakter datoteke *ulaz.txt* u datoteku *izlaz.txt*. U slučaju greške, na standardni izlaz za greške ispisati odgovarajuću poruku.

#### Primer 1

```
ULAZ.TXT
Volim programiranje.

IZLAZ.TXT
Vipgmae
```

#### Primer 2

```
ULAZ.TXT
Zivot je lep!

IZLAZ.TXT
Zojl!
```

#### Primer 3

```
ULAZ.TXT
1234567890

IZLAZ.TXT
1470
```

#### Primer 4

```
ULAZ.TXT
Ova datoteka
sadrzi tekst
u vise
linija.

IZLAZ.TXT
0 te
diet sli.
```

#### Primer 5

```
ULAZ.TXT
U Beogradu ce biti
suncan i lep
dan.

IZLAZ.TXT
Ueruei
nn pa
```

**Zadatak 3.2.3** Napisati program koji šifruje sadržaj datoteke *podaci.txt* tako što svako slovo ciklično zamenjuje njegovim prethodnikom suprotne veličine i upisuje u datoteku *šifra.txt*. Na primer, slovo **b** se zamenjuje slovom **A**, slovo **B** slovom **a**, slovo **a** slovom **Z**, slovo **A** slovom **z**, itd. Ostali karakteri ostaju nepromenjeni. U slučaju greške, na standardni izlaz za greške ispisati odgovarajuću poruku.



*Primer 1*

```

PODACI.TXT
Matematicki fakultet
Studentski trg 16
Beograd

SIFRA.TXT
1ZSDLZSHBJH EZJTKSDS
rSTCDMSRJH SQF 16
aDNFQZC

```

*Primer 2*

```

PODACI.TXT
a=x+y;
x=b+5;

SIFRA.TXT
Z=W+X;
W=A+5;

```

*Primer 3*

```

PODACI.TXT NE POSTOJI

IZLAZ ZA GREŠKE:
Greska: neuspesno otvaranje
datoteke podaci.txt.

```

**Zadatak 3.2.4** Napisati program koji za dve datoteke čija se imena unose sa standardnog ulaza, radi sledeće:

- za svaku cifru u prvoj datoteci, u drugu datoteku upisuje 0
- za svako slovo u prvoj datoteci, u drugu datoteku upisuje 1
- za sve ostale karaktere u prvoj datoteci, u drugu datoteku upisuje 2

Pretpostaviti da je maksimalna dužina naziva datoteka 20 karaktera. U slučaju greške, na standardni izlaz za greške ispisati odgovarajuću poruku.

*Primer 1*

```

INTERAKCIJA SA PROGRAMOM:
Unesite ime prve datoteke:
kilometraze.txt
Unesite ime druge datoteke:
sifra.txt

KILOMETRAZE.TXT
Beograd - Nis 230km
Uzice - Cacak 56.3km
Subotica - Ruma 139km

SIFRA.TXT
111111122211120001121111122
21111120020112111111122211
11200011

```

*Primer 2*

```

INTERAKCIJA SA PROGRAMOM:
Unesite ime prve datoteke:
ulaz.txt
Unesite ime druge datoteke:
izlaz.txt

ULAZ.TXT
18. februar 2019.

IZLAZ.TXT
00221111111200002

```

*Primer 3*

```

INTERAKCIJA SA PROGRAMOM:
Unesite ime prve datoteke:
in.dat
Unesite ime druge datoteke:
out.dat

IN.DAT NE POSTOJI

IZLAZ ZA GREŠKE:
Greska: neuspesno otvaranje
prve datoteke.

```

**Zadatak 3.2.5** Sa standardnog ulaza učitavaju se imena dveju datoteka i jedan karakter koji označava opciju. Napisati program koji prepisuje sadržaj prve datoteke u drugu tako što u slučaju da je navedena opcija u, sva mala slova zamenjuje velikim slovima, a u slučaju da je navedena opcija l, sva velika slova zamenjuje malim slovima. Pretpostaviti da je maksimalna dužina naziva datoteka 20 karaktera. U slučaju greške, na standardni izlaz za greške ispisati odgovarajuću poruku.

### 3 Ulaz i izlaz programa

---

#### Primer 1

```
INTERAKCIJA SA PROGRAMOM:
Unesite imena datoteka i opciju:
ulaz.txt izlaz.txt u

ULAZ.TXT
  danas je lep dan
  i Ja zelim
  da postanem programer

IZLAZ.TXT
  DANAS JE LEP DAN
  I JA ZELIM
  DA POSTANEM PROGRAMER
```

#### Primer 2

```
INTERAKCIJA SA PROGRAMOM:
Unesite imena datoteka i opciju:
prva.dat druga.dat l

PRVA.DAT
  Cena soka je 30
  Cena vina je 150
  Cena limunade je 200
  Cena sendvica je 120

DRUGA.DAT
  cena soka je 30
  cena vina je 150
  cena limunade je 200
  cena sendvica je 120
```

#### Primer 3

```
INTERAKCIJA SA PROGRAMOM:
Unesite imena datoteka i opciju:
primer.c prazna.txt V

PRIMER.C
#include <stdio.h>
int main()
{
}

PRAZNA.TXT

IZLAZ ZA GREŠKE:
Greska: neispravan unos.
```

#### Primer 4

```
INTERAKCIJA SA PROGRAMOM:
Unesite imena datoteka i opciju:
primer.c prazna.txt V

PRIMER.C NE POSTOJI

IZLAZ ZA GREŠKE:
Greska: neuspesno otvaranje
prve datoteke.
```

**Zadatak 3.2.6** Napisati program koji prebrojava mala slova u datoteci *podaci.txt* i dobijeni rezultat ispisuje na standardni izlaz. U slučaju greške, na standardni izlaz za greške ispisati odgovarajuću poruku.

#### Primer 1

```
PODACI.TXT
  Matematicki fakultet
  Studentski trg 16
  Beograd

IZLAZ:
  Broj malih slova je: 36
```

#### Primer 2

```
PODACI.TXT
  PrograMiranje

IZLAZ:
  Broj malih slova je: 11
```

#### Primer 3

```
PODACI.TXT
  MATEMATIKA
  12+34=46

IZLAZ:
  Broj malih slova je: 0
```

**Zadatak 3.2.7** Napisati program koji u datoteci čije se ime unosi sa standardnog ulaza prebrojava koliko se puta pojavljuje svaka od cifara i na standardni izlaz ispisuje cifru sa najvećim brojem pojavljivanja. Ukoliko ima više takvih ci-

fara, ispisati sve. Ukoliko datoteka ne sadrži nijednu cifru, ispisati odgovarajuću poruku. Pretpostaviti da je maksimalna dužina naziva datoteke 20 karaktera. U slučaju greške, na standardni izlaz za greške ispisati odgovarajuću poruku.

*Primer 1*

```
INTERAKCIJA SA PROGRAMOM:
Unesite ime datoteke:
ulaz.txt

ULAZ.TXT
danas je lep dan
i ja zelim
da postanem programer

IZLAZ:
Datoteka ne sadrzi cifre.
```

*Primer 2*

```
INTERAKCIJA SA PROGRAMOM:
Unesite ime datoteke:
prva.dat

PRVA.DAT
Cena soka je 30
Cena vina je 150
Cena limunade je 200
Cena sendvica je 120

IZLAZ:
Najcesce cifre: 0
```

*Primer 3*

```
INTERAKCIJA SA PROGRAMOM:
Unesite ime datoteke:
primer.c

PRIMER.C
1 22 333.444

IZLAZ:
Najcesce cifre: 3 4
```

**Zadatak 3.2.8** Napisati program koji u datoteci čije je ime dato kao argument komandne linije proverava da li su zagrade pravilno uparene. U slučaju greške, na standardni izlaz za greške ispisati odgovarajuću poruku.

*Primer 1*

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

ZAGRADE.TXT
ab( cd) ..
((3+4)*5+1)*9

IZLAZ:
Zagrade jesu uparene.
```

*Primer 2*

```
POKRETANJE:
./a.out primer2.dat

PRIMER2.DAT
(7+8
nisu(
uparene

IZLAZ:
Zagrade nisu uparene.
```

*Primer 3*

```
POKRETANJE:
./a.out primer3.dat

PRIMER3.DAT
)) 7 + 6 ((

IZLAZ:
Zagrade nisu uparene.
```

**Zadatak 3.2.9** Napisati program koji prebrojava slova i cifre u datoteci.

- Napisati funkciju `int ucitaj_karaktere(char s[], FILE *f)` kojom se učitavaju karakteri iz datoteke `f` u niz karaktera `s`. Dozvoljeni karakteri za učitavanje su mala i velika slova engleske abecede, kao i cifre. Učitavanje se prekida kada se nađe na znak za novi red ili nedozvoljeni karakter. Funkcija vraća broj elemenata niza uspešno učitanih karaktera.
- Napisati funkciju `void prebroj(char s[], int n, int *broj_slova, int *broj_cifara)` kojom se određuje broj slovnih elemenata niza karaktera (velikih ili malih slova) kao i broj cifara.

Napisati program koji koristeći prethodne funkcije prebrojava cifre i slova u datoteci čije se ime zadaje kao argument komandne linije, a zatim ispisuje dobijene

### 3 Ulaz i izlaz programa

vrednosti na standardni izlaz. Pretpostaviti da je maksimalni broj karaktera datoteke 1000. U slučaju greške, na standardni izlaz za greške ispisati odgovarajuću poruku.

Primer 1

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

SKUP.TXT
OvoJeSkupKaraktera.123

IZLAZ:
Broj slova: 18
Broj cifara: 0
```

Primer 2

```
POKRETANJE:
./a.out skup2.txt

SKUP2.TXT
ovdeimamo$dolar

IZLAZ:
Broj slova: 9
Broj cifara: 0
```

Primer 3

```
POKRETANJE:
./a.out skup3.txt

SKUP3.TXT
broj3
broj5

IZLAZ:
Broj slova: 4
Broj cifara: 1
```

Primer 4

```
POKRETANJE:
./a.out skup4.txt

SKUP4.TXT
11.2.2019.

IZLAZ:
Broj slova: 0
Broj cifara: 2
```

Primer 5

```
POKRETANJE:
./a.out skup5.txt

SKUP5.TXT NE POSTOJI

IZLAZ ZA GREŠKE:
Greska: neuspesno otvaranje
ulazne datoteke.
```

Primer 6

```
POKRETANJE:
./a.out

IZLAZ ZA GREŠKE:
Greska: neispravan poziv.
```

**Zadatak 3.2.10** Napisati program koji sa standardnog ulaza učitava reč  $s$  i u datoteku *rotacije.txt* upisuje sve njene rotacije. Pretpostaviti da je maksimalna dužina reči 20 karaktera. U slučaju greške, na standardni izlaz za greške ispisati odgovarajuću poruku.

Primer 1

```
INTERAKCIJA SA PROGRAMOM:
Unesite rec: abcde

ROTACIJE.TXT
abcde
bcdea
cdeab
deabc
eabcd
```

Primer 2

```
INTERAKCIJA SA PROGRAMOM:
Unesite rec: 1234

ROTACIJE.TXT
1234
2341
3412
4123
```

Primer 3

```
INTERAKCIJA SA PROGRAMOM:
Unesite rec: a=3*x+5;

ROTACIJE.TXT
a=3*x+5;
=3*x+5;a
3*x+5;a=
*x+5;a=3
x+5;a=3*
+5;a=3*x
5;a=3*x+
;a=3*x+5
```

**Zadatak 3.2.11** Sa standardnog ulaza se učitava ime datoteke i nenegativan ceo broj  $k$ . Napisati program koji učitava reči iz datoteke (reč je niz karaktera

između blanko simbola) i svaku pročitanu reč rotira za  $k$  mesta u levo i tako dobijenu reč upisuje u datoteku čije je ime *rotirano.txt*. Pretpostaviti da je maksimalna dužina naziva datoteke 20 karaktera, da datoteka sadrži samo slova i beline i da je maksimalna dužina jedne reči u datoteci 100 karaktera. U slučaju greške, na standardni izlaz za greške ispisati odgovarajuću poruku.

*Primer 1*

```
INTERAKCIJA SA PROGRAMOM:
Unesite ime datoteke:
ulaz.txt
Unesite broj k: 3

ULAZ.TXT
jedan dva
tri cetiri

ROTIRANO.TXT
anjed dva tri iricet
```

*Primer 2*

```
INTERAKCIJA SA PROGRAMOM:
Unesite ime datoteke:
in.dat
Unesite broj k: 5

IN.DAT
Popodne ce biti kise

ROTIRANO.TXT
nePopod ec itib isek
```

*Primer 3*

```
INTERAKCIJA SA PROGRAMOM:
Unesite ime datoteke:
input.txt
Unesite broj k: 0

INPUT.TXT
Popodne ce
biti kise

ROTIRANO.TXT
Popodne ce biti kise
```

**Zadatak 3.2.12** Napisati program koji iz datoteke *razno.txt* u datoteku *palindromi.txt* prepisuje sve palindrome. Reč je palindrom ako se isto čita sa leve i desne strane bez obzira na veličinu slova. Pretpostaviti da je maksimalna dužina reči 200 karaktera. U slučaju greške, na standardni izlaz za greške ispisati odgovarajuću poruku.

*Primer 1*

```
RAZNO.TXT
Ana i melem su
primeri palindroma.

PALINDROMI.TXT:
Ana i melem
```

*Primer 2*

```
RAZNO.TXT
jabuka neven pomorandza
kuk Oko kapAk pero radar
caj

PALINDROMI.TXT:
neven kuk Oko
kapAk radar
```

*Primer 3*

```
RAZNO.TXT
ovde nema palindroma

PALINDROMI.TXT:
```

**Zadatak 3.2.13** U datoteci čije se ime zadaje sa standardnog ulaza nalazi se broj  $n$  ( $n \leq 256$ ), a zatim i  $n$  reči. Napisati program koji učitava reči iz datoteke u niz i iz niza uklanja sve duplikate i upisuje izmenjeni niz u datoteku *bez\_duplikata.txt*. Pretpostaviti da je maksimalna dužina naziva datoteke 20 karaktera, a maksimalna dužina jedne reči u datoteci 50 karaktera. U slučaju greške, na standardni izlaz za greške ispisati odgovarajuću poruku.

### 3 Ulaz i izlaz programa

#### Primer 1

```
INTERAKCIJA SA PROGRAMOM:
Unesite ime datoteke:
  imena.txt

IMENA.TXT
12
Ana Milos Ana Marko
Petar Filip Jovana Ana
Petar Ivan Nikola Filip

BEZ_DUPLIKATA.TXT:
Ana Milos Marko Petar
Filip Jovana Ivan Nikola
```

#### Primer 2

```
INTERAKCIJA SA PROGRAMOM:
Unesite ime datoteke:
  gradovi.txt

GRADOVI.TXT
10
Sombor Beograd
Nis Beograd
Beograd Indjija
Nis Ruma
Ruma Sombor

BEZ_DUPLIKATA.TXT:
Sombor Beograd Nis
Indjija Ruma
```

#### Primer 3

```
INTERAKCIJA SA PROGRAMOM:
Unesite ime datoteke:
  gradovi.txt

GRADOVI.TXT NE POSTOJI

IZLAZ ZA GREŠKE:
Greska: neuspesno otvaranje
ulazne datoteke.
```

**Zadatak 3.2.14** U datoteci čije se ime zadaje kao prvi argument komandne linije nalazi se ceo pozitivan broj  $n$ , a zatim i  $n$  celih brojeva. Napisati program koji na standardni izlaz ispisuje koliko  $k$ -tocifrenih brojeva postoji u datoteci, pri čemu se pozitivan ceo broj  $k$  zadaje kao drugi argument komandne linije. U slučaju greške, na standardni izlaz za greške ispisati odgovarajuću poruku.

#### Primer 1

```
POKRETANJE:
./a.out ulaz.txt 2

ULAZ.TXT
6
15 193 -27 9790 35 1

IZLAZ:
Broj 2-cifrenih brojeva: 3
```

#### Primer 2

```
POKRETANJE:
./a.out ulaz.txt 5

ULAZ.TXT
4
15 193 -27 9790

IZLAZ:
Broj 5-cifrenih brojeva: 0
```

#### Primer 3

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

IZLAZ ZA GREŠKE:
Greska: neispravan poziv.
```

**Zadatak 3.2.15** Napisati program koji na standardni izlaz ispisuje maksimum brojeva iz datoteke *brojevi.txt*. U slučaju greške, na standardni izlaz za greške ispisati odgovarajuću poruku.

#### Primer 1

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

IZLAZ:
Najveci broj je: 54.96
```

#### Primer 2

```
BROJEVI.TXT
10.5 183.111 -90.2 3.167

IZLAZ:
Najveci broj je: 183.111
```

#### Primer 3

```
BROJEVI.TXT
-62.7 -190.2 -2.3 -1000
-198.25 -8

IZLAZ:
Najveci broj je: -2.3
```

**Zadatak 3.2.16** Prvi red datoteke *matrica.txt* sadrži dva cela broja manja od 50 koji predstavljaju redom broj vrsta i broj kolona realne matrice  $a$ . Svaki sledeći red sadrži po jednu vrstu matrice. Napisati program koji pronalazi sve elemente matrice  $a$  koji su jednaki zbiru svih svojih susednih elemenata i štampa ih u obliku (broj vrste, broj kolone, vrednost elementa). Pretpostaviti da je sadržaj datoteke ispravan. U slučaju greške, na standardni izlaz za greške ispisati odgovarajuću poruku. UPUTSTVO: U zadatku 2.4.6 je dato objašnjenje koji elementi matrice su susedni.

<p><i>Primer 1</i></p> <pre> MATRICA.TXT 3 4 1 2 3 4 7 2 15 -3 -1 3 1 3  IZLAZ: (1, 0, 7) (1, 2, 15) </pre>	<p><i>Primer 2</i></p> <pre> MATRICA.TXT 2 2 1 1 -2 2  IZLAZ: (0, 0, 1) (0, 1, 1) </pre>	<p><i>Primer 3</i></p> <pre> MATRICA.TXT 1 4 9 3 5 2  IZLAZ: (0, 2, 5) </pre>
---	--	---

**Zadatak 3.2.17** Prvi red datoteke *ulaz.txt* sadrži dva cela broja između 2 i 50 koji predstavljaju redom broj vrsta i broj kolona realne matrice  $a$ . Svaki sledeći red sadrži po jednu vrstu matrice. Napisati program koji nalazi i štampa sve četvorke oblika  $(a(i,j), a(i+1,j), a(i,j+1), a(i+1,j+1))$  u kojima su svi elementi međusobno različiti. U slučaju greške, na standardni izlaz za greške ispisati odgovarajuću poruku.

<p><i>Primer 1</i></p> <pre> ULAZ.TXT 3 4 1 2 3 4 7 2 15 -3 -1 3 1 3  IZLAZ: (3, 15, 4, -3) (7, -1, 2, 3) (2, 3, 15, 1) (15, 1, -3, 3) </pre>	<p><i>Primer 2</i></p> <pre> ULAZ.TXT 1 4 9 3 5 2  IZLAZ ZA GREŠKE: Greska: neispravna dimenzija. </pre>	<p><i>Primer 3</i></p> <pre> ULAZ.TXT 2 2 1 1 -2 2  IZLAZ: </pre>
---	--	---

**Zadatak 3.2.18** U datoteci *tacke.txt* se nalazi broj tačaka, a zatim u posebnim redovima za svaku tačku njene  $x$  i  $y$  koordinate. Napisati program koji u datoteku *rastojanja.txt* upisuje rastojanje svake od učitanih tačaka od koordinatnog početka, a na standardni izlaz koordinate tačke koja je od njega najudaljenija. Ukoliko ima više takvih tačaka, ispisati koordinate prve. Koristiti

### 3 Ulaz i izlaz programa

strukturu Tacka sa poljima  $x$  i  $y$ , kao i funkciju kojom se računa rastojanje tačke od koordinatnog početka. Pretpostaviti da je maksimalan broj tačaka u datoteci 50. U slučaju greške, na standardni izlaz za greške ispisati odgovarajuću poruku.

#### Primer 1

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

RASTOJANJA.TXT
11.18
5.83
11.31
4.00

IZLAZ:
Najudaljenija tačka: (8, -8)
```

#### Primer 2

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

IZLAZ ZA GREŠKE:
Greska: neispravan broj tacaka.
```

**Zadatak 3.2.19** Definirati strukturu kojom se opisuje trodimenzioni vektor sa celobrojnim koordinatama  $x$ ,  $y$  i  $z$ . U datoteci *vektori.txt* nalazi se nepoznati broj vektora. Napisati program koji učitava vektore iz ove datoteke i na standardni izlaz ispisuje koordinate vektora sa najvećom dužinom. Ukoliko ima više takvih vektora, ispisati koordinate prvog. Dužina vektora se izračunava po formuli:  $|v| = \sqrt{x^2 + y^2 + z^2}$ . U slučaju greške, na standardni izlaz za greške ispisati odgovarajuću poruku.

#### Primer 1

```
VEKTORI.TXT
4 -1 7
3 1 2

IZLAZ:
4 -1 7
```

#### Primer 2

```
VEKTORI.TXT
4 -4 4
-4 -4 -4

IZLAZ:
4 -4 4
```

#### Primer 3

```
VEKTORI.TXT
0 0 0
0 1 0
1 0 0

IZLAZ:
0 1 0
```

#### Primer 4

```
VEKTORI.TXT
3 0 1
4 5 2
1 0 0
2 -1 2

IZLAZ:
4 5 2
```

#### Primer 5

```
VEKTORI.TXT NE POSTOJI

IZLAZ ZA GREŠKE:
Greska: neuspesno otvaranje
ulazne datoteke.
```

#### Primer 6

```
VEKTORI.TXT
1 1 1

IZLAZ:
1 1 1
```



**Zadatak 3.2.20** Definisati strukturu *Pravougaonik* koja sadrži dužine stranica i ime pravougaonika. Napisati program koji iz datoteke čije ime se zadaje kao argument komandne linije učitava podatke o pravougaonicima (nije poznato koliko), a zatim ispisuje imena onih pravougaonika koji su kvadrati i vrednost najveće površine među pravougaonicima koji nisu kvadrati. Pretpostaviti da je maksimalan broj pravougaonika 200, a maksimalna dužina imena pravougaonika 4. U slučaju greške, na standardni izlaz za greške ispisati odgovarajuću poruku.

*Primer 1*

```
POKRETANJE:
./a.out pravougaonici.dat

PRAVOUGAONICI.DAT
2 4 p1
3 3 p2
1 6 p3

IZLAZ:
p2 8
```

*Primer 2*

```
POKRETANJE:
./a.out dva.dat

DVA.DAT
5 2 pm
4 7 pv

IZLAZ:
28
```

*Primer 3*

```
POKRETANJE:
./a.out tri.dat

TRI.DAT
5 5 m
3 3 s
8 8 xl

IZLAZ:
m s xl
```

**Zadatak 3.2.21** U datoteci *studenti.txt* se nalaze podaci o studentima. Za svakog studenta je dato korisničko ime na Alas serveru i poslednjih pet ocena koje je dobio. Napisati program koji pronalazi studenta koji je ostvario najbolji uspeh i ispisuje njegove podatke. Ukoliko ima više takvih studenata, ispisati informacije o svima. Pretpostaviti da je maksimalni broj studenta 100. U slučaju greške, na standardni izlaz za greške ispisati odgovarajuću poruku.

*Primer 1*

```
STUDENTI.TXT
mr15239 10 9 9 8 10
mi14005 8 8 9 8 10
ml15112 9 8 8 7 10
mr15007 10 10 10 10 10
mn13208 7 7 9 6 10

IZLAZ:
Studenti sa najvećim prosekom:
Korisnicko ime: mr15007
Prosek ocena: 10.00
```

*Primer 2*

```
STUDENTI.TXT
mr16156 10 9 9 10 10
mi17234 9 9 10 10 10
ml17084 9 8 8 8 8

IZLAZ:
Studenti sa najvećim prosekom:
Korisnicko ime: mr16156
Prosek ocena: 9.6

Korisnicko ime: mi17234
Prosek ocena: 9.6
```

**Zadatak 3.2.22** Definisati strukturu *Student* koja sadrži puno ime studenta, niz njegovih ocena, broj ocena i prosečnu ocenu. U datoteci čije se ime zadaje kao argument komandne linije se nalaze podaci o studentima. Za svakog studenta dato je ime, prezime i niz ocena koji se završava nulom. Svi podaci su razdvojeni razmacima. Napisati program koji učitava podatke o studentima i na standardni izlaz ispisuje podatke za studenta sa najvećim prosekom (prosek

### 3 Ulaz i izlaz programa

ispisati na 2 decimale). Ukoliko ima više takvih studenata, ispisati informacije o prvom studentu. Pretpostaviti da je maksimalni broj ocena 10 i maksimalna dužina punog imena 100 karaktera. U slučaju greške, na standardni izlaz za greške ispisati odgovarajuću poruku. UPUTSTVO: *Ime i prezime studenta se mogu pročitati pomoću specifikatora %s a potom se za kreiranje niske **puno\_ime** u traženom formatu može iskoristiti funkcija **strcat**.*

#### Primer 1

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

STUDENTI.TXT
Marko Markovic 5 6 7 8 9 0
Jelena Jankovic 10 10 10 0
Filip Viskovic 10 9 8 7 6 0
Jana Peric 10 10 9 9 8 8 7 0

IZLAZ:
Jelena Jankovic 10 10 10 10.00
```

#### Primer 2

```
POKRETANJE: ./a.out

IZLAZ ZA GREŠKE:
Greska: neispravan poziv.
```

**Zadatak 3.2.23** Imena ulazne i izlazne datoteke se redom navode kao argumenti komandne linije. U ulaznoj datoteci se nalaze podaci o razlomcima: u prvom redu se nalazi broj razlomaka, a u svakom sledećem redu brojilac i imenilac po jednog razlomka. Definirati strukturu koja opisuje razlomak i napisati program koji učitava niz razlomaka iz datoteke, a potom:

- ukoliko je prilikom pokretanja programa navedena opcija **x**, upisati u izlaznu datoteku recipročni razlomak za svaki razlomak iz niza
- ukoliko je prilikom pokretanja programa navedena opcija **y**, upisati u izlaznu datoteku realnu vrednost recipročnog razlomka svakog razlomka iz niza

Pretpostaviti da se u ulaznoj datoteci nalazi najviše 100 razlomaka. U slučaju greške, na standardni izlaz za greške ispisati odgovarajuću poruku.

#### Primer 1

```
POKRETANJE:
./a.out d1.txt d2.txt -x

D1.TXT
4
1 5
19 3
-2 7
97 90
```

#### Primer 1 (nastavak)

```
D2.TXT
5/1
3/19
-7/2
90/97
```

#### Primer 2

```
POKRETANJE:
./a.out ulaz.txt izlaz.txt

IZLAZ ZA GREŠKE:
Greska: neispravan poziv.
```

*Primer 3*

```
POKRETANJE: ./a.out d1.txt d2.txt -y
D1.TXT
4
1 5
19 3
-2 7
97 90
```

*Primer 3 (nastavak)*

```
D2.TXT
5.000000
0.157894
-3.500000
0.927835
```

**Zadatak 3.2.24** Definirati strukturu `Automobil` koja sadrži marku, model i cenu. Napisati program koji iz datoteke čije se ime zadaje sa standardnog ulaza učitava broj automobila i podatke za svaki automobil i zatim:

- ispisuje prosečnu cenu po marki automobila
- za maksimalnu cenu koju je kupac spreman da plati, a koja se zadaje kao argument komandne linije, ispisuje automobile u tom cenovnom rangu

Pretpostaviti da se model i marka sastoje od jedne reči, da svaka od njih sadrži najviše 30 karaktera i da se u datoteci nalaze podaci za najviše 100 automobila. U slučaju greške, na standardni izlaz za greške ispisati odgovarajuću poruku.

*Primer 1*

```
POKRETANJE: ./a.out 5000
INTERAKCIJA SA PROGRAMOM:
Unesite naziv datoteke:
dat1.txt
DAT1.TXT NE POSTOJI
IZLAZ ZA GREŠKE:
Greska: neuspesno otvaranje
ulazne datoteke.
```

*Primer 2*

```
POKRETANJE: ./a.out
IZLAZ ZA GREŠKE:
Greska: neispravan poziv.
```

*Primer 3*

```
POKRETANJE: ./a.out 4000
INTERAKCIJA SA PROGRAMOM:
Unesite naziv datoteke:
dat1.txt
DAT1.TXT
7
renault twingo 2900
renault megan 6250
renault clio 3650
dacia logan 5400
dacia sandero 7800
```

*Primer 3 (nastavak)*

```
fiat bravo 4900
fiat linea 4290
IZLAZ:
Informacije o prosečnoj
ceni po markama:
renault 4266.67
dacia 6600.00
fiat 4595.00
Kola u Vasem cenovnom rangu:
renault twingo 2900
renault clio 3650
```

### 3 Ulaz i izlaz programa

**Zadatak 3.2.25** Kao argumenti komandne linije zadaju se ime datoteke i ceo broj  $k$ . Napisati program koji na standardni izlaz ispisuje sve linije zadate datoteke čija je dužina veća od  $k$ . Pretpostaviti da je maksimalna dužina linije 80 karaktera. U slučaju greške, na standardni izlaz za greške ispisati odgovarajuću poruku.

#### Primer 1

```
POKRETANJE: ./a.out test.txt 7

TEST.TXT
Teme koje su obradjuvane:
Petlje
Funkcije
Nizovi
Strukture

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

#### Primer 2

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

IZLAZ ZA GREŠKE:
Greska: neispravan poziv.
```

**Zadatak 3.2.26** Napisati program koji u datoteci čije se ime navodi kao argument komandne linije određuje liniju maksimalne dužine i ispisuje je na standardni izlaz. Ukoliko ima više takvih linija, ispisati onu koja je leksikografski prva. Pretpostaviti da je maksimalna dužina linije 80 karaktera. U slučaju greške, na standardni izlaz za greške ispisati odgovarajuću poruku.

#### Primer 1

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

TEST.TXT
Danas je veoma hladno decembarsko
popodne. Ne pada sneg, kazu mozda
ce sutra.

IZLAZ:
Danas je veoma hladno decembarsko
```

#### Primer 2

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

IN.TXT NE POSTOJI

IZLAZ ZA GREŠKE:
Greska: neuspesno otvaranje
ulazne datoteke.
```

**Zadatak 3.2.27** U datoteci čije se ime navodi kao prvi argument komandne linije navedena je reč  $r$  i niz linija. Napisati program koji u datoteku čije se ime navodi kao drugi argument komandne linije upisuje sve linije prve datoteke u kojima se reč  $r$  pojavljuje bar  $n$  puta gde je  $n$  pozitivan ceo broj koji se unosi sa standardnog ulaza. Prilikom prebrojavanja, računaju se i samostalna pojavljivanja reči  $r$  i pojavljivanja u okviru neke druge reči. Ispis treba da bude u formatu broj\_pojavljivanja:linija. Pretpostaviti da je maksimalna dužina reči 100 karaktera, a linije 500 karaktera. U slučaju greške, na standardni izlaz za greške

ispisati odgovarajuću poruku.

#### Primer 1

```
POKRETANJE: ./a.out input.txt output.txt

INTERAKCIJA SA PROGRAMOM:
  Unesite broj n: 2

INPUT.TXT
  sto
  stolica lampa
  postotak Stopiranje stopa
  presto Ostoja stotina prostoriya

OUTPUT.TXT
  2: postotak Stopiranje stopa
  4: presto Ostoja stotina prostoriya
```

#### Primer 2

```
POKRETANJE: ./a.out input.txt output.txt

INTERAKCIJA SA PROGRAMOM:
  Unesite broj n: 3

INPUT.TXT
  red
  redar za ovu nedelju
  redosled ured
  odrediti raspored

OUTPUT.TXT
```

#### Primer 3

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

IN.TXT NE POSTOJI

IZLAZ ZA GREŠKE:
  Greska: neuspesno otvaranje
  ulazne datoteke.
```

#### Primer 4

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

IZLAZ ZA GREŠKE:
  Greska: neispravan poziv.
```

**Zadatak 3.2.28** Napisati program koji prebrojava koliko se linija datoteke *ulaz.txt* završava niskom *s* koja se učitava sa standardnog ulaza. Pretpostaviti da je maksimalna dužina linije 80 karaktera, a niske *s* 20 karaktera. U slučaju greške, na standardni izlaz za greške ispisati odgovarajuću poruku.

#### Primer 1

```
ULAZ.TXT
  /home/korisnik/imena.txt
  /home/korisnik/a.out
  /home/cv.pdf
  /home/korisnik/ulaz.txt
  /home/rezultati.xlsx
  /var/log/apache2/error.log

INTERAKCIJA SA PROGRAMOM:
  Unesite nisku s: .txt
  Broj linija: 2
```

#### Primer 2

```
ULAZ.TXT
  /var/log/apache2/error.log
  /var/log/dpkg.log
  moj_log.log
  /home/korisnik.login
  /home/korisnik.log.txt

INTERAKCIJA SA PROGRAMOM:
  Unesite nisku s: .log
  Broj linija: 3
```

**Zadatak 3.2.29** Napisati program koji linije koje se učitavaju sa standardnog ulaza sve do kraja ulaza prepisuje u datoteku *izlaz.txt* i to, ako je prilikom pokretanja zadata opcija *-v* ili *-V* samo one linije koje počinju velikim slovom, ako je zadata opcija *-m* ili *-M* samo one linije koje počinju malim slovom, a ako je opcija izostavljena sve linije. Pretpostaviti da je maksimalna dužina linije 80

### 3 Ulaz i izlaz programa

karaktera. U slučaju greške, na standardni izlaz za greške ispisati odgovarajuću poruku.

#### Primer 1

```
POKRETANJE: ./a.out -m

INTERAKCIJA SA PROGRAMOM:
Unesite recenice:
programiranje u C-u je zanimljivo
Volim programiranje!
Kada porastem bicu programer!
u slobodno vreme programiram

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

#### Primer 2

```
POKRETANJE: ./a.out -V

INTERAKCIJA SA PROGRAMOM:
Unesite recenice:
programiranje u C-u je zanimljivo
Volim programiranje!
Kada porastem bicu programer!
u slobodno vreme programiram

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

#### Primer 3

```
POKRETANJE: ./a.out -k

IZLAZ ZA GREŠKE:
Greska: neispravna opcija.
```

#### Primer 4

```
POKRETANJE: ./a.out

IZLAZ ZA GREŠKE:
Greska: neispravan poziv.
```

**Zadatak 3.2.30** Napisati program koji poredi dve datoteke i ispisuje redni broj linija u kojima se datoteke razlikuju. Imena datoteka se zadaju kao argumenti komandne linije. Pretpostaviti da je maksimalna dužina linije 200 karaktera. Linije brojati počevši od 1. U slučaju greške, na standardni izlaz za greške ispisati odgovarajuću poruku.

#### Primer 1

```
POKRETANJE:
./a.out ulaz.txt izlaz.txt

ULAZ.TXT
danas vezbamo
programiranje
ovo je primer kad su
datoteke iste

IZLAZ.TXT:
danas vezbamo
programiranje
ovo je primer kad su
datoteke iste

IZLAZ:
```

#### Primer 2

```
POKRETANJE:
./a.out u1.dat u2.dat

U1.DAT
danas vezbamo
analizu
ovo je primer kad
su datoteke razlicite

U2.DAT
danas vezbamo
programiranje
ovo je primer kad su
datoteke razlicite

IZLAZ:
2 3 4
```

#### Primer 3

```
POKRETANJE:
./a.out prva.dat druga.dat

PRVA.DAT
ovo je primer
kada su
datoteke
razlicite duzine

DRUGA.DAT
ovo je primer kada
su
datoteke
razlicite
duzine
i kada treba ispisati broj
tih redova

IZLAZ:
1 2 4 5 6 7
```

# Dodatak A

## Ispitni rokovi

### A.1 Modul Matematika

#### A.1.1 Praktični deo ispita, januar 2019.

**Zadatak A.1.1** Napisati program koji učitava četvorocifrene brojeve do unosa broja 0, a zatim ispisuje one brojeve kojima je cifra desetica najveća cifra u zapisu. Ukoliko nema takvih brojeva među unetima, ispisati broj 0. U slučaju greške, ispisati  $-1$  na standardni izlaz za greške.

##### Test 1

```
ULAZ:
9523 -8542 3232 -9999 -1121 1576 0
IZLAZ:
3232 -9999 -1121 1576
```

##### Test 2

```
ULAZ:
4596 1234 9631 -120 0
IZLAZ:
4596
IZLAZ ZA GREŠKE:
-1
```

##### Test 3

```
ULAZ:
9876 2258 -4579 4689 -5567 6630 1200 5204 0
IZLAZ:
0
```

**Zadatak A.1.2** Napisati program koji pomaže korisniku da "šifruje" svoju elektronsku adresu kako ne bi dobijao nepoželjne poruke. "Šifrovanje" adrese se vrši tako što se znak @ zameni sa [AT]. Elektronska adresa se učitava kao niska maksimalne dužine 100 karaktera sa standardnog ulaza, a šifrovana adresa

se ispisuje na standardni izlaz. U slučaju da elektronska adresa nije ispravno zadata ispisati  $-1$  na standardni izlaz za greške.

Test 1	Test 2	Test 3
<pre> ULAZ: korisnik@gmail.com IZLAZ: korisnik[AT]gmail.com </pre>	<pre> ULAZ: student@matf.bg.ac.rs IZLAZ: student[AT]matf.bg.ac.rs </pre>	<pre> ULAZ: pogresnaadresa@yahoo.com IZLAZ ZA GREŠKE: -1 </pre>

**Zadatak A.1.3** Definirati strukturu *Hemijski\_element* koja sadrži naziv elementa (nisku dužine najviše 20 karaktera), oznaku elementa (nisku dužine najviše 2 karaktera) i broj neutrona (ceo broj). Napisati program koji učitava podatke o hemijskim elementima do unosa reči **kraj**, a potom još jedan naziv elementa i na standardni izlaz ispisuje oznaku i broj neutrona tog elementa. Ukoliko element nije pronađen među učitanim podacima, ispisati -1.

NAPOMENA: *Pretpostaviti da neće biti uneto više od 120 elemenata, kao i da su podaci o hemijskim elementima ispravno zadati.*

Test 1	Test 2	Test 3
<pre> ULAZ: kalcijum Ca 20 cink Zn 35 fosfor P 16 kraj fosfor IZLAZ: P 16 </pre>	<pre> ULAZ: nikl Ni 31 bor B 6 kripton Kr 48 natrijum Na 12 kraj hrom IZLAZ ZA GREŠKE: -1 </pre>	<pre> ULAZ: litijum Li 4 ugljenik C 6 aluminijum Al 14 srebro Ag 61 gvozdje Fe 40 brom Br 45 kraj ugljenik IZLAZ: C 6 </pre>

**Zadatak A.1.4** U datoteci *pesme.txt* dat je ceo broj  $n$  koji označava broj pesama, a potom i  $n$  redova sa podacima o pesmama. U svakom redu naveden je naziv pesme i njen žanr (niske bez belina, dužine najviše 30 karaktera). Napisati program koji učitava podatke iz datoteke, a zatim, u zavisnosti od opcije koja se zadaje kao argument komandne linije, obrađuje podatke na sledeći način:

- ukoliko je zadata opcija **-p**, učitava se sa standardnog ulaza jedan karakter i na standardni izlaz ispisuju svi nazivi pesama koji počinju zadatim karakterom;
- ukoliko je zadata opcija **-z**, učitava se sa standardnog ulaza niska koja predstavlja žanr pesme i na standardni izlaz ispisuju nazivi svih pesama odabranog žanra.



Prilikom odabira pesama za ispis, zanemariti veličinu slova. U slučaju greške, ispisati  $-1$  na standardni izlaz za greške.

#### Test 1

```
POKRETANJE: ./a.out -p
PESME.TXT
7
BohemianRhapsody rock
RollingInTheDeep pop
StairwayToHeaven rock
BeatIt pop
SoWhat jazz
MyFunnyValentine jazz
Smooth pop
ULAZ:
S
IZLAZ:
StairwayToHeaven
SoWhat
Smooth
```

#### Test 2

```
POKRETANJE: ./a.out -z
PESME.TXT
7
BohemianRhapsody rock
RollingInTheDeep pop
StairwayToHeaven rock
BeatIt pop
SoWhat jazz
MyFunnyValentine jazz
Smooth pop
ULAZ:
pop
IZLAZ:
RollingInTheDeep
BeatIt
Smooth
```

#### Test 3

```
POKRETANJE: ./a.out -x
IZLAZ ZA GREŠKE:
-1
```

#### Test 4

```
POKRETANJE: ./a.out -p -z
IZLAZ ZA GREŠKE:
-1
```

#### Test 5

```
POKRETANJE: ./a.out
IZLAZ ZA GREŠKE:
-1
```

### A.1.2 Praktični deo ispita, februar 2019.

**Zadatak A.1.5** Napisati program koji učitava pozitivan četvorocifren broj  $n$ , a zatim na standardni izlaz ispisuje zbir onih cifara broja  $n$  koje su po vrednosti veće od aritmetičke sredine svih cifara broja  $n$ . U slučaju greške, ispisati  $-1$  na standardni izlaz za greške.

#### Test 1

```
ULAZ:
1234
IZLAZ:
7
```

#### Test 2

```
ULAZ:
6745
IZLAZ:
13
```

#### Test 3

```
ULAZ:
100
IZLAZ ZA GREŠKE:
-1
```

#### Test 4

```
ULAZ:
-1234
IZLAZ ZA GREŠKE:
-1
```

**Zadatak A.1.6** Napisati program koji učitava nisku  $s$  parne dužine od najviše 20 karaktera i na standardni izlaz ispisuje nisku koja se dobija nadovezivanjem karaktera prve polovine niske  $s$  na drugu polovinu niske  $s$ . U slučaju greške, ispisati  $-1$  na standardni izlaz za greške.

Test 1	Test 2	Test 3	Test 4
<pre> ULAZ:   Beograde IZLAZ:   radeBeog           </pre>	<pre> ULAZ:   matematika IZLAZ:   atikamatem           </pre>	<pre> ULAZ:   1234 IZLAZ:   3412           </pre>	<pre> ULAZ:   abc1234 IZLAZ ZA GREŠKE:   -1           </pre>

**Zadatak A.1.7** Napisati program koji čita sadržaj datoteke *ulaz.txt* i ispisuje na standardni izlaz sve niske datoteke koje predstavljaju cele brojeve. U slučaju greške, ispisati -1 na standardni izlaz za greške.

Test 1	Test 2	Test 3
<pre> POKRETANJE: ./a.out ULAZ.TXT   123 ab1 2ab -23 IZLAZ:   123 -23           </pre>	<pre> POKRETANJE: ./a.out ULAZ.TXT   145as 25gf 265 478 65 -96 IZLAZ:   265 478 65 -96           </pre>	<pre> POKRETANJE: ./a.out ULAZ.TXT   Oude nema brojeva IZLAZ:           </pre>
<pre> Test 4  POKRETANJE: ./a.out ULAZ.TXT NE POSTOJI! IZLAZ ZA GREŠKE:   -1           </pre>		

**Zadatak A.1.8** Napisati program koji sa standardnog ulaza učitava podatke o osvajačima takmičenja. Za svako takmičenje se redom zadaju godina takmičenja (pozitivan ceo broj) i ime osvajača (niska od najviše 30 karaktera bez belina). Program treba da ispiše:

- ako je navedena opcija -y kao prvi argument komandne linije, ime osvajača takmičenja za godinu koja se navodi kao drugi argument
- ako je navedena opcija -w kao prvi argument komandne linije, sve godine u kojima je takmičar čije se ime navodi kao drugi argument komande linije osvajao takmičenje.

U slučaju greške, ispisati -1 na standardni izlaz za greške.

NAPOMENA: Podrazumevati da su ulazni podaci o takmičenjima ispravni. Broj osvajača nije unapred poznat.

*Test 1*

```

POKRETANJE: ./a.out -y 2016
ULAZ:
2011 ManUtd
2012 ManCity
2013 ManUtd
2014 ManCity
2015 Chelsea
2016 Leicester
2017 Chelsea
2018 ManCity
IZLAZ:
Leicester

```

*Test 2*

```

POKRETANJE: ./a.out -w RealMadrid
ULAZ:
2011 Barcelona
2012 Chelsea
2013 BayernMunich
2014 RealMadrid
2015 Barcelona
2016 RealMadrid
2017 RealMadrid
2018 RealMadrid
IZLAZ:
2014 2016 2017 2018

```

*Test 3*

```

POKRETANJE: ./a.out -s 2001
IZLAZ ZA GREŠKE:
-1

```

*Test 4*

```

POKRETANJE: ./a.out -x
IZLAZ ZA GREŠKE:
-1

```

*Test 5*

```

POKRETANJE: ./a.out -s 2012 2000
IZLAZ ZA GREŠKE:
-1

```

*Test 6*

```

POKRETANJE: ./a.out -y 2005 -w RealMadrid
IZLAZ ZA GREŠKE:
-1

```

## A.2 Modul Informatika

### A.2.1 Praktični deo ispita, januar 2019.

**Zadatak A.2.1** Napisati program koji učitava cele trocifrene brojeve sve do kraja ulaza i na standardni izlaz ispisuje one čije su cifre uređene strogo rastuće (cifre se čitaju sa leva na desno). U slučaju greške, ispisati  $-1$  i prekinuti izvršavanje programa.

*Test 1*

```

ULAZ:
-532 236 100 -555 546
IZLAZ:
236

```

*Test 2*

```

ULAZ:
123 -123 321 -321
IZLAZ:
123 -123

```

*Test 3*

```

ULAZ:
258 695 -1234
IZLAZ:
258 -1

```

*Test 4*

```

ULAZ:
14
IZLAZ:
-1

```

**Zadatak A.2.2** Napisati program koji sa standardnog ulaza učitava reč s maksimalne dužine 20 karaktera (bez belina), a zatim karakter koji predstavlja način modifikacije učitane niske:

- ukoliko je učitani karakter  $m$ , sve karaktere reči  $s$  koji su mala slova, pretvoriti u odgovarajuća velika
- ukoliko je učitani karakter  $v$ , sve karaktere reči  $s$  koji su velika slova, pretvoriti u odgovarajuća mala
- ukoliko je učitani karakter  $o$ , ne menjati karaktere reči  $s$

Na standardni izlaz ispisati nisku nakon modifikacije. U slučaju greške, ispisati  $-1$  na standardni izlaz i prekinuti izvršavanje programa.

Test 1	Test 2	Test 3	Test 4
<pre> ULAZ:   sreca m IZLAZ:   SRECA </pre>	<pre> ULAZ:   IspiT v IZLAZ:   ispit </pre>	<pre> ULAZ:   Rec o IZLAZ:   Rec </pre>	<pre> ULAZ:   PROgram x IZLAZ:   -1 </pre>

**Zadatak A.2.3** Napisati program za praćenje rezultata automobilske trke. Na takmičenju učestvuje  $n$  ( $n \geq 3$ ) takmičara u  $m$  ( $m \geq 2$ ) trka. Program prvo učitava broj takmičara i trka, a zatim za svakog od  $n$  takmičara vreme u sekundama u svakoj od  $m$  trka. Pretpostaviti da neće biti više od 100 takmičara i 100 trka. Vremena čuvati u matrici dimenzije  $n \times m$  tako da element  $(i, j)$  predstavlja vreme koje je takmičar  $i$  postigao u  $j$ -toj trci. Na standardni izlaz ispisati redne brojeve takmičara (brojeći ih od 0) koji su pobedili u trkama (bili najbrži), redom za svaku trku. Pretpostaviti da neće biti više takmičara sa istim prolaznim vremenom po trci. U slučaju greške, ispisati  $-1$  na standardni izlaz i prekinuti izvršavanje programa.

Test 1	Test 2	Test 3	Test 4
<pre> ULAZ:   3 3   192.9 87.8 109.102   181.2 92.1 102.4   151.1 87.9 118.9 IZLAZ:   2 0 1 </pre>	<pre> ULAZ:   3 4   51.3 184.94 121.7 99.51   50.9 182.71 119.2 99.2   51.2 192.11 122.9 100.1 IZLAZ:   1 1 1 1 </pre>	<pre> ULAZ:   4 3   113.5 145.2 -14.5 IZLAZ:   -1 </pre>	<pre> ULAZ:   4 -3 IZLAZ:   -1 </pre>

**Zadatak A.2.4** Definirati strukturu sa nazivom *Kutija* koja sadrži dužinu, širinu i visinu kutije u centimetrima (pozitivni celi brojevi). Napisati program koji učitava pozitivan ceo broj  $n$  ( $n \leq 100$ ), a zatim i podatke o  $n$  kutija. Nakon toga, program treba da ispiše zapreminu kutije u koju se može smestiti svaka od preostalih  $n - 1$  kutija pojedinačno. Pretpostaviti da neće biti više takvih kutija, a ukoliko takva kutija ne postoji, ispisati 0. U slučaju greške, ispisati  $-1$  na standardni izlaz i prekinuti izvršavanje programa.

NAPOMENA: Da bi jedna kutija (sa celobrojnim dimenzijama) stala u drugu, svaka od dimenzija te kutije (dužina, širina i visina redom) mora biti manja barem 1 centimetar od odgovarajućih dimenzija druge kutije. Prilikom smeštanja jedne kutije u drugu nema obrtanja kutije.

Test 1	Test 2	Test 3	Test 4
<pre> ULAZ: 4 15 2 9 185 27 12 16 21 10 120 12 3 IZLAZ: 59940 </pre>	<pre> ULAZ: 3 9 18 2 21 5 3 3 15 5 IZLAZ: 0 </pre>	<pre> ULAZ: -3 IZLAZ: -1 </pre>	<pre> ULAZ: 3 1 2 3 8 9 -5 IZLAZ: -1 </pre>

## A.2.2 Praktični deo ispita, februar 2019.

**Zadatak A.2.5** Napisati program koji učitava cele trocifrene brojeve sve do kraja ulaza i na standardni izlaz ispisuje one brojeve čija je cifra desetica jednaka aritmetičkoj sredini cifara stotina i jedinica. U slučaju greške, ispisati  $-1$  na standardni izlaz i prekinuti izvršavanje programa.

Test 1	Test 2	Test 3	Test 4
<pre> ULAZ: 543 236 100 -555 546 IZLAZ: 543 -555 </pre>	<pre> ULAZ: 402 -402 103 -103 IZLAZ: </pre>	<pre> ULAZ: -1234 IZLAZ: -1 </pre>	<pre> ULAZ: 14 IZLAZ: -1 </pre>

**Zadatak A.2.6** Sa standardnog ulaza se učitava niska  $s$  maksimalne dužine 30 karaktera. Napisati program koji na standardni izlaz ispisuje dužinu najduže podniske niske  $s$  čiji su karakteri uređeni strogo rastuće po ASCII kodovima čitajući sa leva na desno.

Test 1	Test 2	Test 3	Test 4
<pre> ULAZ: stolica IZLAZ: 2 </pre>	<pre> ULAZ: a12bcABc IZLAZ: 4 </pre>	<pre> ULAZ: PPPPPPPP IZLAZ: 1 </pre>	<pre> ULAZ: abcdefw IZLAZ: 7 </pre>

**Zadatak A.2.7** Sa standardnog ulaza se učitava neparan prirodan broj  $n$  ( $n \leq 101$ ), a zatim  $n^2$  celih brojeva koje treba sačuvati u odgovarajućoj kvadratnoj matrici. Proveriti da li je suma elemenata na glavnoj dijagonali matrice

## A Ispitni rokovi

neparna, i ako jeste, na standardni izlaz ispisati vrednost maksimalnog elementa glavne dijagonale. Ako to nije slučaj, ispisati vrednost minimalnog elementa glavne dijagonale. U slučaju greške, ispisati  $-1$  na standardni izlaz i prekinuti izvršavanje programa.

Test 1	Test 2	Test 3	Test 4
ULAZ: 3 15 6 7 2 -4 -2 3 2 6 IZLAZ: 15	ULAZ: 5 12 6 7 1 2 2 -4 -2 2 0 3 2 6 10 7 3 2 6 12 5 12 6 7 1 2 IZLAZ: -4	ULAZ: 4 IZLAZ: -1	ULAZ: -7 IZLAZ: -1

**Zadatak A.2.8** Definirati strukturu sa nazivom *Student* koja sadrži podatke o studentu: indeks studenta (pozitivan ceo broj), broj poena ostvaren na ispitu (nenegativan realan broj dvostruke tačnosti iz intervala  $[0, 100]$ ) i oznaku učionice u kojoj je student polagao ispit (niska iz skupa "704", "718", "rlab" i "bim"). Napisati program koji sa standardnog ulaza učitava prirodan broj  $n$ , a zatim podatke o  $n$  studenata koji su polagali ispit iz Programiranja 1, redom, indeks, broj poena i oznaku učionice. Nakon podataka o studentima se učitava oznaka učionice za koju treba ispisati broj studenata iz te učionice koji su položili ispit. Oznaka učionice se zadaje kao niska od najviše 10 karaktera. Pretpostaviti da su podaci o studentima ispravni i da neće biti više od 100 studenata. U slučaju greške ispisati  $-1$  na standardni izlaz i prekinuti izvršavanje programa. Student je položio ispit ako je na istom ostvario bar 51 poen.

Test 1	Test 2	Test 3	Test 4
ULAZ: 9 20180001 98 704 20180002 33 704 20180003 7 718 20180005 61.8 rlab 20180006 50.5 bim 20180007 55.6 718 20180008 51 704 20180009 30 rlab 20180010 40.4 rlab 704 IZLAZ: 2	ULAZ: 7 20180003 73 718 20180005 60.8 rlab 20180006 40.5 bim 20180007 45.6 718 20180008 19.9 704 20180009 31.4 rlab 20180010 49.4 rlab rlab IZLAZ: 1	ULAZ: 4 20180001 98 704 20180002 33 704 20180003 73.5 718 20180005 60.8 rlab bim IZLAZ: 0	ULAZ: -4 IZLAZ: -1