

## PROGRAMIRANJE 1



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

# **PROGRAMIRANJE 1**

## **Zbirka zadataka**

**Beograd  
2016.**

Autori:

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

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

*Danijela Simić*, asistent na Matematičkom fakultetu u Beogradu

*Anđelka Zečević*, asistent na Matematičkom fakultetu u Beogradu

PROGRAMIRANJE 1

Zbirka zadataka

# Sadržaj

<b>1</b>	<b>Uvodni zadaci</b>	<b>1</b>
1.1	Naredba izraza . . . . .	1
<b>2</b>	<b>Kontrola toka</b>	<b>11</b>
2.1	Naredbe grananja . . . . .	11
2.2	Petlje . . . . .	22
2.3	Funkcije . . . . .	42
<b>3</b>	<b>Predstavljanje podataka</b>	<b>57</b>
3.1	Nizovi . . . . .	57
3.2	Pokazivači . . . . .	74
3.3	Niske . . . . .	80
3.4	Strukture . . . . .	90
<b>4</b>	<b>Ulaz i izlaz programa</b>	<b>99</b>
4.1	Datoteke . . . . .	99



# 1

## Uvodni zadaci

### 1.1 Naredba izraza

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

*Primer 1*

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Zdravo svima!
```

**Zadatak 1.1.2** Napisati program 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 ispisuje najpre unete vrednosti, a zatim i njihov zbir, razliku, proizvod, ceo deo pri deljenju prvog broja drugim brojem i ostatak pri deljenju prvog broja drugim brojem. NAPOMENA: *Pretpostaviti da je unos korektan, tj. da druga uneta vrednost nije 0.*

## 1 Uvodni zadaci

---

### Primer 1

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

### Primer 2

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

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

### Primer 1

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

### Primer 2

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

**Zadatak 1.1.5** Napisati program koji za unetu količinu jabuka u kilogramima i unetu cenu po kilogramu ispisuje ukupnu vrednost date količine jabuka. NAPOMENA: *Pretpostaviti da je cena jabuka pozitivan ceo broj i da je unos korektan.*

### Primer 1

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

### Primer 2

```
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. NAPOMENA: *Pretpostaviti da su cene svih artikala pozitivni celi brojevi, kao i da su unete vrednosti ispravne, tj. da se može vratiti kusur.*

### Primer 1

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

### Primer 2

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



**Zadatak 1.1.7** Napisati program koji za uneta vremena poletanja i sletanja aviona ispisuje dužinu trajanja leta. NAPOMENA: *Pretpostaviti da su poletanje i sletanje u istom danu kao i da su sve vrednosti ispravno unete.*

*Primer 1*

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

*Primer 2*

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

**Zadatak 1.1.8** Date su dve celobrojne promenljive. Napisati program koji razmenjuje njihove vrednosti.

*Primer 1*

```
INTERAKCIJA SA PROGRAMOM:
Unesi dve celobrojne vrednosti: 5 7
pre zamene: x=5, y=7
posle zamene: x=7, y=5
```

*Primer 2*

```
INTERAKCIJA SA PROGRAMOM:
Unesi dve celobrojne vrednosti: 237 -592
pre zamene: x=237, y=-592
posle zamene: x=-592, y=237
```

**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:
Unesi dve celobrojne vrednosti: 5 7
Nove vrednosti su: a=12, b=-2
```

*Primer 2*

```
INTERAKCIJA SA PROGRAMOM:
Unesi dve celobrojne vrednosti: 237 -592
Nove vrednosti su: a=-355, b=829
```

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

*Primer 1*

```
INTERAKCIJA SA PROGRAMOM:
Unesi trocifreni broj: 697
jedinica 7, desetica 9, stotina 6
```

*Primer 2*

```
INTERAKCIJA SA PROGRAMOM:
Unesi trocifreni broj: 504
jedinica 4, desetica 0, stotina 5
```

**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 dinar. NAPOMENA: *Pretpostaviti da je cena proizvoda pozitivan ceo broj.*

## 1 Uvodni zadaci

---

### Primer 1

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

### Primer 2

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

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

### Primer 1

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

### Primer 2

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

**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 ispisom cifara u obrnutom poretku
- (e) izračunava broj koji se dobija zamenom cifre jedinice i cifre stotine

NAPOMENA: *Pretpostaviti da je unos ispravan.*

*Primer 1*

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

*Primer 2*

```
INTERAKCIJA SA PROGRAMOM:
Unesite cetvorocifreni broj: 3570
Proizvod cifara: 0
Razlika sume krajnjih i srednjih: -9
Suma kvadrata cifara: 83
Broj u obrnutom poretku: 753
Broj sa zamenjenom cifrom jedinica i stotina: 3075
```

**Zadatak 1.1.14** Napisati program koji ispisuje broj koji se dobija izbacivanjem cifre desetica u unetom prirodnom broju.

*Primer 1*

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

*Primer 2*

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

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

*Primer 1*

```
INTERAKCIJA SA PROGRAMOM:
Unesite pozitivan prirodan broj: 12345
Unesite pozitivan dvocifreni broj: 67
Novi broj je 1267345
```

*Primer 2*

```
INTERAKCIJA SA PROGRAMOM:
Unesite pozitivan prirodan broj: 50000000
Unesite pozitivan dvocifreni broj: 12
Novi broj je 705044704
```

**Zadatak 1.1.16** Napisati program koji učitava realnu vrednost izraženu u inčima, konvertuje tu vrednost u centimetre i ispisuje je zaokruženu na dve decimale. **UPUTSTVO:** *Jedan inč ima 2.54 centimetra.*

## 1 Uvodni zadaci

---

### Primer 1

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

### Primer 2

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

**Zadatak 1.1.17** Napisati program koji učitava dužinu izraženu u miljama, konvertuje tu vrednost u kilometre i ispisuje je zaokruženu na dve decimalne. UPUTSTVO: *Jedna milja ima 1.609344 kilometara.*

### Primer 1

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

### Primer 2

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

**Zadatak 1.1.18** Napisati program koji učitava težinu izraženu u funtama, konvertuje tu vrednost u kilograme i ispisuje je zaokruženu na dve decimalne. UPUTSTVO: *Jedna funta ima 0.45359237 kilograma.*

### Primer 1

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

### Primer 2

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

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

### Primer 1

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

### Primer 2

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

**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 4 decimalne.

*Primer 1*

```

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

```

*Primer 2*

```

|| INTERAKCIJA SA PROGRAMOM:
|| Unesite brojeve: -1 0 0 1
|| -1.0000

```

*Primer 3*

```

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

```

*Primer 4*

```

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

```

**Zadatak 1.1.21** 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 duzinu poluprecnika kruga: 4.2
|| Obim: 26.39, povrsina: 55.42

```

*Primer 2*

```

|| INTERAKCIJA SA PROGRAMOM:
|| Unesite duzinu poluprecnika kruga: 14.932
|| Obim: 93.82, povrsina: 700.46

```

**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.*

*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.

## 1 Uvodni zadaci

---

NAPOMENA: *Pretpostaviti da je unos ispravan.*

### Primer 1

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

### Primer 2

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

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

### 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 unetih brojeva je 6.67
```

### Primer 2

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

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

## 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 je 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 je 152380.00

```

**Zadatak 1.1.28** Napisati program koji za unete pozitivne prirodne brojeve  $x$ ,  $p$  i  $c$  ispisuje broj koji se dobija ubacivanjem cifre  $c$  u broj  $x$  na poziciju  $p$ . NAPOMENA: Podrazumevati da je unos ispravan, tj. da je broj  $p$  manji od ukupnog broja cifara broja  $x$ . Numeracija cifara počinje od nule, odnosno cifra najmanje težine nalazi se na nultoj poziciji. UPUTSTVO: Koristiti funkciju `pow` iz `math.h` biblioteke.

## Primer 1

```

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

```

## Primer 1

```

INTERAKCIJA SA PROGRAMOM:
Unesite redom x, p i c: 12345 2 9
Rezultat je: 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.

## 1 Uvodni zadaci

---

### Primer 1

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

### Primer 2

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

**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 je 4
```

### Primer 2

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

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

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

zaokruženu na dve decimale.

### Primer 1

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

### Primer 1

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



## 2

# Kontrola toka

## 2.1 Naredbe grananja

**Zadatak 2.1.1** Napisati program koji za dva uneta cela broja ispisuje njihov minimum.

*Primer 1*

```
|| INTERAKCIJA SA PROGRAMOM:  
| Unesite dva cela broja: 5 18  
| Minimum je 5.
```

*Primer 2*

```
|| INTERAKCIJA SA PROGRAMOM:  
| Unesite dva cela broja: 43 -16  
| Minimum je -16.
```

**Zadatak 2.1.2** Napisati program koji za dva uneta cela broja ispisuje njihov maksimum.

*Primer 1*

```
|| INTERAKCIJA SA PROGRAMOM:  
| Unesite dva cela broja: 141 67  
| Maksimum je 141.
```

*Primer 2*

```
|| INTERAKCIJA SA PROGRAMOM:  
| Unesite dva cela broja: -893 -54  
| Maksimum je -54.
```

**Zadatak 2.1.3** Napisati program koji za uneti realan broj ispisuje njegovu apsolutnu vrednost zaokruženu na dve decimale.

*Primer 1*

```
|| INTERAKCIJA SA PROGRAMOM:  
| Unesite jedan realan broj: 7.42  
| Njegova apsolutna vrednost je: 7.42
```

*Primer 2*

```
|| INTERAKCIJA SA PROGRAMOM:  
| Unesite jedan realan broj: -562.428  
| Njegova apsolutna vrednost je: 562.43
```

### Primer 3

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesite jedan realan broj: 0  
|| Njegova apsolutna vrednost je: 0.00
```

### Primer 4

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesite jedan realan broj: 52  
|| Njegova apsolutna vrednost je: 52.00
```

**Zadatak 2.1.4** Napisati program koji za uneti ceo broj ispisuje njegovu recipročnu vrednost zaokruženu na četiri decimalne.

### Primer 1

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

### Primer 2

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

### Primer 3

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesite jedan ceo broj: 0  
|| Nedozvoljeno deljenje nulom.
```

### Primer 4

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

**Zadatak 2.1.5** Napisati program koji učitava tri cela broja i ispisuje zbir pozitivnih.

### Primer 1

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

### Primer 2

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesite tri cela broja: -15 81 0  
|| Suma unetih pozitivnih brojeva: 81
```

### Primer 3

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

### Primer 4

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesite tri cela broja: 16 2 576  
|| Suma unetih pozitivnih brojeva: 594
```

**Zadatak 2.1.6** U prodavnici je organizovana akcija da svaki kupac dobije najjeftiniji od tri artikla za jedan dinar. Napisati program koji za unete cene tri artikla izračunava ukupnu cenu, kao i koliko dinara se uštedi zahvaljujući popustu. NAPOMENA: *Pretpostaviti da su cene artikala pozitivni celi brojevi.*

### Primer 1

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

### Primer 2

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

### Primer 3

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

### Primer 4

```
INTERAKCIJA SA PROGRAMOM:
Unesite cene tri artikla: 247 133 126
Cena sa popustom: 381
Usteda: 125
```

**Zadatak 2.1.7** Napisati program koji za uneti četvorocifreni broj ispisuje njegovu najveću cifru.

### Primer 1

```
INTERAKCIJA SA PROGRAMOM:
Unesite broj: 6835
Najveca cifra je: 8
```

### Primer 2

```
INTERAKCIJA SA PROGRAMOM:
Unesite broj: 238
Greska: Niste uneli cetvorocifren broj!
```

### Primer 3

```
INTERAKCIJA SA PROGRAMOM:
Unesite broj: 7777
Najveca cifra je: 7
```

### Primer 4

```
INTERAKCIJA SA PROGRAMOM:
Unesite broj: -2002
Najveca cifra je: 2
```

**Zadatak 2.1.8** Napisati program koji za uneto vreme (broj sati iz intervala  $[0, 24)$  i broj minuta iz intervala  $[0, 60)$ ) ispisuje koliko je sati i minuta ostalo do ponoći.

### Primer 1

```
INTERAKCIJA SA PROGRAMOM:
Unesite vreme (broj sati u intervalu [0,24),
broj minuta u intervalu [0,60)): 18 19
Do ponoci je ostalo 5 sati i 41 minuta.
```

### Primer 2

```
INTERAKCIJA SA PROGRAMOM:
Unesite vreme (broj sati u intervalu [0,24),
broj minuta u intervalu [0,60)): 23 7
Do ponoci je ostalo 0 sati i 53 minuta.
```

### Primer 3

```
INTERAKCIJA SA PROGRAMOM:
Unesite vreme (broj sati u intervalu [0,24),
broj minuta u intervalu [0,60)): 24 20
Neispravan unos.
```

### Primer 4

```
INTERAKCIJA SA PROGRAMOM:
Unesite vreme (broj sati u intervalu [0,24),
broj minuta u intervalu [0,60)): 14 0
Do ponoci je ostalo 10 sati i 0 minuta.
```

**Zadatak 2.1.9** Napisati program koji za učitani karakter ispisuje uneti karakter i njegov ASCII kod. Ukoliko je uneti karakter malo (veliko) slovo, ispisati i odgovarajuće veliko (malo) slovo i njegov ASCII kod.

## 2 Kontrola toka

---

### Primer 1

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

### Primer 2

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

### Primer 3

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

### Primer 4

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

**Zadatak 2.1.10** Napisati program koji za unetih pet karaktera ispisuje koliko je među njima malih slova.

### Primer 1

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite karaktere: A u E f h
|| Broj malih slova: 3
```

### Primer 2

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite karaktere: k L M 9 o
|| Broj malih slova: 2
```

**Zadatak 2.1.11** Program učitava pet karaktera. Napisati koliko se puta pojavilo veliko ili malo slovo a.

### Primer 1

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite karaktere: aBcAe
|| 2
```

### Primer 2

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite karaktere: aa4A_
|| 3
```

### Primer 3

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite karaktere: aAaAa
|| 5
```

### Primer 4

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite karaktere: B6(vV
|| 0
```

**Zadatak 2.1.12** Program učitava pet karaktera. Ispisati koliko puta su se pojavile cifre.

### Primer 1

```
|| INTERAKCIJA SA PROGRAMOM:  
| Unesite karaktere: A1cA3  
| 2
```

### Primer 2

```
|| INTERAKCIJA SA PROGRAMOM:  
| Unesite karaktere: 2a45_  
| 2
```

### Primer 3

```
|| INTERAKCIJA SA PROGRAMOM:  
| Unesite karaktere: 43986  
| 5
```

### Primer 4

```
|| INTERAKCIJA SA PROGRAMOM:  
| Unesite karaktere: B6(vV  
| 1
```

**Zadatak 2.1.13** Napisati program koji za unetu godinu ispisuje da li je prestupna.

### Primer 1

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

### Primer 2

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

### Primer 3

```
|| INTERAKCIJA SA PROGRAMOM:  
| Unesite godinu: 2000  
| Godina je prestupna.
```

### Primer 4

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

**Zadatak 2.1.14** Broj je Armstrongov ako je jednak zbiru kubova svojih cifara. Napisati program koji za dati trocifren broj proverava da li je Armstrongov.

### Primer 1

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

### Primer 2

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

### Primer 3

```
|| INTERAKCIJA SA PROGRAMOM:  
| Unesite broj: 84  
| Greska: Niste uneli trocifren broj!
```

### Primer 4

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

**Zadatak 2.1.15** Napisati program koji ispisuje proizvod parnih cifara unetog četvorocifrenog broja.

### 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: -1234  
|| Proizvod parnih cifara: 8
```

### Primer 4

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesite broj: 288  
|| Broj nije cetvorocifren!
```

**Zadatak 2.1.16** Napisati program koji učitava četvorocifreni broj i ispisuje broj koji se dobija kada se unetom broju razmene najmanja i najveća cifra. NAPOMENA: *U slučaju da se najmanja ili najveća cifra pojavljuju na više pozicija, uzeti prvo pojavljivanje.*

### Primer 1

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

### Primer 2

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesite broj: 247  
|| Broj nije cetvorocifren!
```

### Primer 3

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesite broj: 1192  
|| 9112
```

### Primer 4

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

**Zadatak 2.1.17** Napisati program koji ispituje da li se tačke  $A(x_1, y_1)$  i  $B(x_2, y_2)$  nalaze u istom kvadrantu i ispisuje odgovor DA ili NE.

**Zadatak 2.1.18** Napisati program koji ispituje da li se tačke  $A(x_1, y_1)$ ,  $B(x_2, y_2)$  i  $C(x_3, y_3)$  nalaze na istoj pravoj i ispisuje odgovor DA ili NE.

**Zadatak 2.1.19** Napisati program za rad sa intervalima. Za dva intervala realne prave  $[a1, b1]$  i  $[a2, b2]$ , program treba da odredi:

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

### Primer 1

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

### Primer 2

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

**Zadatak 2.1.20** Napisati program koji za uneti ceo broj  $x$  ispisuje njegov znak, tj da li je broj jednak nuli, manji od nule ili veći od nule.

### Primer 1

```
INTERAKCIJA SA PROGRAMOM:
Unesite jedan ceo broj: 17
Broj je veci od nule.
```

### Primer 2

```
INTERAKCIJA SA PROGRAMOM:
Unesite jedan ceo broj: 0
Broj je jednak nuli.
```

### Primer 3

```
INTERAKCIJA SA PROGRAMOM:
Unesite jedan ceo broj: -586
Broj je manji od nule.
```

### Primer 4

```
INTERAKCIJA SA PROGRAMOM:
Unesite jedan ceo broj: 62
Broj je veci od nule.
```

**Zadatak 2.1.21** 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 2.1.22** Napisati program koji za uneti četvorocifreni broj proverava da li su njegove cifre uređene rastuće, opadajuće ili nisu uređene i štampa odgovarajuću poruku.

### 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: 6792
|| Cifre nisu uredjene.
```

### Primer 4

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite cetvorocifreni broj: 88
|| Uneti broj nije cetvorocifren.
```

**Zadatak 2.1.23** Napisati program koji učitava karakter i:

- a) ako je  $c$  malo slovo, ispisuje odgovarajuće veliko
- b) ako je  $c$  veliko slovo, ispisuje odgovarajuće malo
- c) ako je  $c$  cifra, ispisuje poruku *cifra*
- d) u ostalim slučajevima, ispisuje karakter  $c$  između dve zvezdice.

### Primer 1

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite karakter: K
|| k
```

### Primer 2

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite karakter: 8
|| cifra
```

### Primer 3

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite karakter: >
|| **
```

### Primer 4

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite karakter: o
|| 0
```

**Zadatak 2.1.24** U nizu 12345678910111213....9899 ispisani su redom brojevi od 1 do 99. Napisati program koji za uneti ceo broj  $k$  ( $1 \leq k \leq 189$ ) ispisuje cifru koja se nalazi na  $k$ -toj poziciji datog niza.

### Primer 1

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

### Primer 2

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

**Zadatak 2.1.25** Data je funkcija  $f(x) = 2 \cdot \cos(x) - x^3$ . Napisati program koji za učitanu vrednost realne promenljive  $x$  i vrednost celobrojne promenljive  $k$  koje može biti 1, 2 ili 3 izračunava vrednost funkcije  $F(k, x) = f(f(f(\dots f(x))))$  gde je funkcija  $f$  primenjena  $k$ -puta i ispisuje je zaokruženu na dve decimala. U slučaju neispravnog ulaza, odštampati odgovarajuću poruku o grešci.



### Primer 1

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

### Primer 2

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

### Primer 3

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

### Primer 4

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesite redom x i k: 1 3  
|| F(1, 3)=-8.74
```

**Zadatak 2.1.26** Napisati program koji za uneti redni broj dana u nedelji ispisuje ime odgovarajućeg dana. U slučaju pogrešnog unosa ispisati odgovarajuću poruku.

### Primer 1

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesite broj: 4  
|| U pitanju je: cetvrtak
```

### Primer 2

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

### Primer 3

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesite broj: 8  
|| Greska: nedozvoljeni unos!
```

### Primer 4

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesite broj: 2  
|| U pitanju je: utorak
```

**Zadatak 2.1.27** Napisati program koji za uneti karakter ispituje da li je samoglasnik.

### Primer 1

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

### Primer 2

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

### Primer 3

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

### Primer 4

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

**Zadatak 2.1.28** Napisati program koji učitava dva cela broja i jedan od karaktera +, -, \*, / ili % i ispisuje vrednost izraza dobijenog primenom date operacije na date argumente. U slučaju pogrešnog unosa ispisati odgovarajuću poruku.

### Primer 1

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesite operator i dva cela broja: - 8 11  
|| Rezultat je: -3
```

### Primer 2

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesite operator i dva cela broja: / 14 0  
|| Greska: deljenje nulom nije dozvoljeno!
```

### Primer 3

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesite operator i dva cela broja: ? 5 7  
|| Greska: nepoznat operator!
```

### Primer 4

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesite operator i dva cela broja: / 19 5  
|| Rezultat je: 3
```

**Zadatak 2.1.29** Napisati program koji za uneti dan i mesec ispisuje godišnje doba kojem pripadaju. NAPOMENA: *Podrazumevati da je unos korektan.*

### Primer 1

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesite dan i mesec: 14 10  
|| jesen
```

### Primer 2

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesite dan i mesec: 2 8  
|| leto
```

### Primer 3

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesite dan i mesec: 27 2  
|| zima
```

### Primer 4

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesite dan i mesec: 19 5  
|| prolece
```

**Zadatak 2.1.30** Napisati program koji za unetu godinu i mesec ispisuje naziv meseca kao i koliko dana ima u tom mesecu te godine.

**Zadatak 2.1.31** Napisati program koji za uneti datum u formatu *dan.mesec.godina*. proverava da li je korektan.

### Primer 1

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

### Primer 2

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

**Zadatak 2.1.32** Napisati program koji za korektno unet datum u formatu *dan.mesec.godina*. ispisuje datum prethodnog dana.

### Primer 1

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

### Primer 2

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

**Zadatak 2.1.33** Napisati program koji za korektno unet datum u formatu *dan.mesec.godina.* ispisuje datum narednog dana.

### Primer 1

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

### Primer 2

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

**Zadatak 2.1.34** Korisnik unosi tri cela broja:  $P$ ,  $Q$  i  $R$ . Nakon toga unosi i dva karaktera, *op1* i *op2*. Ovi karakteri predstavljaju operacije nad unetim brojevima i imaju naredno značenje:

- karakter **k** predstavlja logičku konjukciju
- karakter **d** predstavlja logičku disjunkciju
- karakter **m** predstavlja relaciju manje
- karakter **v** predstavlja relaciju veće

Program treba da sračuna vrednost izraza  $P \text{ op1 } Q \text{ op2 } R$  i da ga ispiše.

### Primer 1

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite tri cela broja: 0 1 2
|| Unesite dva karaktera cela broja: k m
|| 1
```

### Primer 2

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite tri cela broja: -3 -1 0
|| Unesite dva karaktera cela broja: d k
|| 0
```

\* **Zadatak 2.1.35** Program učitava jedan karakter i osam realnih brojeva koji predstavljaju koordinate četiri tačke:  $A(x_1, y_1)$ ,  $B(x_2, y_2)$ ,  $C(x_3, y_3)$ ,  $D(x_4, y_4)$ . Na osnovu unetog karaktera ispisuje se odgovarajuća poruka na standardni izlaz:

- ukoliko je uneti karakter  $k$  - proverava da li su date tačke temena pravougaonika čije su stranice paralelne koordinatnim osama i u slučaju da jesu, ispisuje vrednost obima datog pravougaonika. Možemo podrazumevati da će korisnik koordinate tačaka unositi redom  $A, B, C, D$ , pri čemu  $ABCD$  opisuje pravougaonik čije su stranice  $AB, BC, CD, DA$ , a dijagonale  $AC$  i  $BD$ . Na primer, tačke  $(1, 1)$ ,  $(2, 1)$ ,  $(2, 2)$ ,  $(1, 2)$  čine pravougaonik čije su stranice paralelne koordinatnim osama i čiji je obim 4 a tačke  $(1, 1)$ ,  $(2, 2)$ ,  $(3, 3)$ ,  $(4, 4)$  ne čine pravougaonik.
- ukoliko je uneti karakter  $h$  - proverava da li su unete tačke kolinearne i ukoliko jesu, ispisuje jednačinu prave kojoj pripadaju. Na primer, tačke  $(1, 2)$ ,  $(2, 3)$ ,  $(3, 4)$ ,  $(4, 5)$  su kolinearne i pripadaju pravoj  $y = x + 1$ , tačke

$(1, 1), (1, 2), (1, 3), (1, 4)$  su kolinearne i pripadaju pravoj  $x = 1$ , a tačke  $(1, 1), (2, 1), (2, 2), (1, 2)$  nisu kolinearne.

- ukoliko je uneti karakter  $j$  - Kramerovim pravilom proverava da li je sistem jednačina  $x_1 * p + x_2 * q = x_4 - x_3, y_1 * p + y_2 * q = y_4 - y_3$  određen, neodređen ili nema rešenja, i u slučaju da je određen ispisuje rešenja.

**Zadatak 2.1.36** Polje šahovske table se definiše parom prirodnih brojeva ne većih od 8: prvi se odnosi na red, drugi na kolonu. Ako su dati takvi parovi, napisati program koji proverava:

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

## 2.2 Petlje

**Zadatak 2.2.1** Napisati program koji 5 puta ispisuje tekst *Mi volimo da programiramo*.

### *Primer 1*

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

**Zadatak 2.2.2** Nikola želi da obraduje baku i da joj kupi jedan poklon u radnji. On na raspolaganju ima  $m$  novaca. U radnji se nalazi  $n$  artikala i zanima ga koliko ima artikala u radnji čija cena je manja ili jednaka  $m$ . Napisati program koji pomaže Nikoli da brzo odrediti broj artikala. Program učitava realan pozitivan broj  $m$ , ceo pozitivan broj  $n$  i  $n$  realnih pozitivnih brojeva različitih od 0. Ispisati koliko artikala ima manju ili jednaku cenu od  $m$ . U slučaju greške ispisati odgovarajuću poruku.

### Primer 1

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

### Primer 2

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

### Primer 3

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

### Primer 4

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite broj m: 30
|| Unesite broj n: 4
|| Unesite n brojeva: 67 -100 23 98
|| Cena ne moze biti negativna.
```

**Zadatak 2.2.3** Napisati program koji učitava cele brojeve sve dok se ne unese nula. Nakon toga ispisati proizvod onih unetih brojeva koji su pozitivni.

### Primer 1

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

### Primer 2

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite brojeve:
|| -5 -200 -43 0
|| Nisu uneseni pozitivni
|| brojevi.
```

### Primer 2

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

**Zadatak 2.2.4** Napisati program koji za pozitivan ceo broj proverava i ispisuje da li se cifra 5 nalazi u njegovom zapisu.

### Primer 1

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite broj: 1857
|| Cifra 5 se nalazi u zapisu!
```

### Primer 2

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite broj: 84
|| Cifra 5 se ne nalazi u
|| zapisu!
```

### Primer 3

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite broj: -235515
|| Cifra 5 se nalazi u zapisu!
```

**Zadatak 2.2.5** Program učitava cele brojeve sve do unosa broja nula 0. Napisati program koji 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: 762 -12 800 2010 -356 899 -101
|| 0
|| Aritmeticka sredina: 571.7143
```

### Primer 3

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

**Zadatak 2.2.6** U prodavnici se nalaze artikala čije cene su realni pozitivni brojevi. Program unosi cene artikala sve do unosa broja nula 0. Napisati program koji izračunava i ispisuje prosečnu vrednost cena u radnji.

### Primer 1

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

### Primer 2

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesite cene: 6.32 -9  
|| Cena ne može biti negativana.
```

### Primer 3

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesite cene: 0  
|| Nisu unesene cene.
```

**Zadatak 2.2.7** Program učitava ceo pozitivan broj  $n$ , a potom  $n$  realnih brojeva. Odrediti koliko puta je prilikom unosa došlo do promene znaka. Ispisati dobijenu vrednost.

### Primer 1

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesite broj n: 10  
|| Unesite brojeve:  
|| 7.82 4.3 -1.2 56.8 -3.4 -72.1 8.9 11.2 -11.2  
|| -102.4  
|| Broj promena je 5.
```

### Primer 2

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

### Primer 3

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

### Primer 4

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesite broj n: 0  
|| Broj promena je 0.
```

**Zadatak 2.2.8** U prodavnici se nalazi  $n$  artikala čije cene su realni brojevi. Napisati program koji učitava  $n$ , a potom i cenu svakog od  $n$  artikala i određuje i ispisuje najmanju cenu.

*Primer 1*

```

INTERAKCIJA SA PROGRAMOM:
Unesite broj artikla: 6
Unesite artikle:
12 3.4 90 100.53 53.2 12.8
Minimalna cena je: 3.400000

```

*Primer 2*

```

INTERAKCIJA SA PROGRAMOM:
Unesite broj artikla: 3
Unesite artikle: 4 -8 92
Cena ne moze biti negativna.

```

*Primer 3*

```

INTERAKCIJA SA PROGRAMOM:
Unesite broj artikla: -9
Neispravan unos.

```

**Zadatak 2.2.9** Program učitava ceo pozitivan broj  $n$ , a zatim i  $n$  celih brojeva. Napisati program koji ispisuje broj sa najvećom cifrom desetica. Ukoliko ima više takvih, ispisati prvi.

*Primer 1*

```

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

```

*Primer 2*

```

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

```

*Primer 3*

```

INTERAKCIJA SA PROGRAMOM:
Unesite broj n: -12
Neispravan unos.

```

**Zadatak 2.2.10** Program učitava ceo pozitivan broj  $n$ , a zatim i  $n$  celih brojeva. Napisati program koji ispisuje broj sa najvećim brojem cifara. Ukoliko ima više takvih, ispisati prvi.

*Primer 1*

```

INTERAKCIJA SA PROGRAMOM:
Unesite broj n: 5
Unesite n brojeva: 18 365 25 1 78
Najviše cifara ima broj 365.

```

*Primer 2*

```

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

```

**Zadatak 2.2.11** Program učitava ceo pozitivan broj  $n$ , a zatim i  $n$  celih brojeva. Napisati program koji ispisuje broj sa najvećom vodećom cifrom. Vodeća cifra je cifra najveće težine u zapisu broja. Ukoliko ima više takvih, ispisati prvi.

*Primer 1*

```

INTERAKCIJA SA PROGRAMOM:
Unesite broj n: 5
Unesite n brojeva: 8 964 32 511 27
Broj sa najvećom vodećom cifrom je 964.

```

*Primer 1*

```

INTERAKCIJA SA PROGRAMOM:
Unesite broj n: 3
Unesite n brojeva: 41 669 8
Broj sa najvećom vodećom cifrom je 8.

```

**Zadatak 2.2.12** Vršna su merenja nadmorskih visina na određenom delu teritorije i naučnike zanima razlika između najveće i najmanje nadmorske visine. Napisati program koji učitava realne brojeve sve do unosa 0 koji označavaju nadmorske visine i ispisuje razliku najveće i najmanje nadmorske visine.

*Primer 1*

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

*Primer 2*

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

**Zadatak 2.2.13** Program učitava cele pozitivane brojeve  $n$  ( $n > 1$ ) i  $d$ , a zatim i  $n$  celih brojeva. Napisati program koji izračunava koliko ima parova uzastopnih brojeva među unetim brojevima koji se nalaze na rastojanju  $d$ . Rastojanje između brojeva je definisano sa  $d(x, y) = |y - x|$ . Ispisati rezultat.

*Primer 1*

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

*Primer 2*

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

**Zadatak 2.2.14** Napisati program koji uneti prirodan broj transformiše tako što svaku parnu cifru u zapisu broja uveća za jedan. Ispisati novodobijeni broj.

*Primer 1*

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

*Primer 2*

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

*Primer 3*

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

**Zadatak 2.2.15** Napisati program koji formira i ispisuje broj koji se dobija izbacivanjem svake druge cifre polaznog celog broja, počevši od krajnje desne cifre.

*Primer 1*

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

*Primer 2*

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



### Primer 3

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

### Primer 4

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite broj: -67123
|| -613
```

\* **Zadatak 2.2.16** Napisati program koji na osnovu unetog prirodnog broja formira i ispisuje broj koji se dobija izbacivanjem cifara koje su jednake zbiru svojih suseda.

### Primer 1

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

### Primer 2

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

### Primer 3

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

\* **Zadatak 2.2.17** Broj je *palindrom* ukoliko se isto čita i sa leve i sa desne strane. Napisati program koji učitava prirodan broj i proverava da li je učitani broj palindrom.

### Primer 1

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite broj: 25452
|| Broj je palindrom!
```

### Primer 2

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite broj: 895
|| Broj nije palindrom!
```

### Primer 3

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite broj: 5
|| Broj je palindrom!
```

**Zadatak 2.2.18** Fibonačijev niz počinje ciframa 1 i 1, a svaki član se dobija zbirom prethodna dva. Napisati program koji učitava ceo prirodan broj  $n$  i određuje i ispisuje  $n$ -ti član Fibonačijevog niza.

### Primer 1

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite ceo broj: 10
|| Trazeni broj je: 55
```

### Primer 2

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite ceo broj: -100
|| Neispravan unos. Pozicija u Fibonacijevom
|| nizu mora biti pozitivan broj koji nije 0!
```

### Primer 3

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite ceo broj: 78
|| Trazeni broj je: 375819880
```

### Primer 4

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite ceo broj: 20
|| Trazeni broj je: 6765
```

## 2 Kontrola toka

---

\* **Zadatak 2.2.19** Niz prirodnih brojeva formira se prema sledećem pravilu:

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

Napisati program koji za uneti početni član niza  $a_0$  (ceo pozitivan broj) štampa niz brojeva sve do onog člana niza koji je jednak 1.

### Primer 1

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite ceo broj: 56
|| 56 28 14 7 11 17 26 13 20 10
|| 5 8 4 2 1
```

### Primer 2

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite ceo broj: -48
|| Nekorektan unos. Broj mora biti pozitivan.
```

### Primer 3

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite ceo broj: 67
|| 67 101 152 76 38 19 29 44 22 11
|| 17 26 13 20 10 5 8 4 2 1
```

### Primer 4

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite ceo broj: 33
|| 33 50 25 38 19 29 44 22
|| 11 17 26 13 20 10 5 8 4 2 1
```

\* **Zadatak 2.2.20** 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 prirodan broj  $k$  ispisuje dimenzije papira  $A_k$  u milimetrima. Rezultat ispisati kao celobrojne vrednosti.

### Primer 1

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

### Primer 2

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite format papira: 3
|| 297 420
```

### Primer 3

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite format papira: 7
|| 74 105
```

### Primer 4

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

**Zadatak 2.2.21** 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 2.2.22** Napisati program koji učitava karaktere sve do kraja ulaza, a potom ispisuje broj velikih slova, broj malih slova, broj cifara, broj belina i zbir unetih cifara.

*Primer 1*

```
INTERAKCIJA SA PROGRAMOM:
Tekst sa brojevima: 124, -8900, 23...
velika: 1, mala: 15, cifre: 9, beline: 5
suma cifara: 29
```

*Primer 2*

```
INTERAKCIJA SA PROGRAMOM:
NEMA cifara!
velika: 4, mala: 6, cifre: 0, beline: 1
suma cifara: 0
```

**Zadatak 2.2.23** Program učitava ceo pozitivan broj  $n$ , a potom i  $n$  karaktera. Za svaki od samoglasnika ispisati koliko puta se pojavio među unetim karakterima. Ne praviti razliku između malih i velikih slova.

*Primer 1*

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

*Primer 2*

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

**Zadatak 2.2.24** Program učitava ceo broj  $n$ , a zatim i  $n$  karaktera. Napisati program koji proverava da li se od unetih karaktera može napisati reč *Zima*.

*Primer 1*

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

*Primer 2*

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

**Zadatak 2.2.25** Napisati program koji učitava ceo pozitivan broj  $n$  i ispisuje vrednost sume kubova brojeva od 1 do  $n$ , odnosno  $s = 1 + 2^3 + 3^3 + \dots + n^3$ . U slučaju greške pri unosu podataka ispisati odgovarajuću poruku.

### Primer 1

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite pozitivan ceo broj: 14
|| Suma kubova od 1 do 14: 11025
```

### Primer 2

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite pozitivan ceo broj 25
|| Suma kubova od 1 do 25: 105625
```

**Zadatak 2.2.26** Napisati program koji učitava ceo pozitivan broj  $n$  i ispisuje sumu kubova,  $s = 1 + 2^3 + 3^3 + \dots + k^3$ , za svaku vrednost  $k = 1, \dots, n$ . U slučaju greške pri unosu podataka ispisati odgovarajuću poruku.

### Primer 1

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite pozitivan ceo broj: 5
|| i=1, s=1
|| i=2, s=9
|| i=3, s=36
|| i=4, s=100
|| i=5, s=225
```

### Primer 2

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite pozitivan ceo broj 8
|| i=1, s=1
|| i=2, s=9
|| i=3, s=36
|| i=4, s=100
|| i=5, s=225
|| i=6, s=441
|| i=7, s=784
|| i=8, s=1296
```

**Zadatak 2.2.27** Program učitava realan broj  $x$  i ceo pozitivan broj  $n$ . Napisati program koji izračunava i ispisuje sumu  $S = x + 2 \cdot x^2 + 3 \cdot x^3 + \dots + n \cdot x^n$ .

### Primer 1

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

### Primer 2

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

**Zadatak 2.2.28** Program učitava realan broj  $x$  i ceo pozitivan broj  $n$ . Napisati program koji izračunava i ispisuje sumu  $S = 1 + \frac{1}{x} + \frac{1}{x^2} + \dots + \frac{1}{x^n}$ .

### Primer 1

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

### Primer 2

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

\* **Zadatak 2.2.29** Napisati program koji učitava realane brojeve  $x$  i  $eps$  i sa zadatom tačnošću  $eps$  izračunava i ispisuje sumu  $S = 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \dots$ . Izračunati sumu u odnosu na tačnost  $eps$  znači uporediti poslednji član sume sa  $eps$  i ukoliko je taj poslednji član manji od  $eps$  prekinuti dalja izračunavanja. UPUTSTVO: *Prilikom računanja sume koristiti prethodni izračunati član sume u*

računanju sledećeg člana sume. Naime, ako je izračunat član sume  $\frac{x^n}{n!}$  na osnovu njega se lako može dobiti član  $\frac{x^{n+1}}{(n+1)!}$ . Nikako ne računati stepen i faktoriјel 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 2.2.30** Napisati program koji učitava realane brojeve  $x$  i  $eps$  i sa zatom tačnošću  $eps$  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 2.2.31** Napisati program koji učitava realan broj  $x$  i prirodan broj  $n$  izračunava sumu  $S = (1 + \cos(x)) \cdot (1 + \cos(x^2)) \cdot \dots \cdot (1 + \cos(x^n))$ . NAPOMENA: Voditi računa o efikasnosti rešenja.

### Primer 1

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

### Primer 2

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

**\* Zadatak 2.2.32** Napisati program koji učitava ceo prirodan broj  $n$  i ispisuje vrednost razlomka

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

### Primer 1

```
INTERAKCIJA SA PROGRAMOM:
Unesite prirodan broj: 4
Razlomak = 0.697674
```

### Primer 2

```
INTERAKCIJA SA PROGRAMOM:
Unesite prirodan broj: 20
Razlomak = 0.697775
```

### Primer 3

```
INTERAKCIJA SA PROGRAMOM:
Unesite prirodan broj: 0
Neispravan unos.
```

\* **Zadatak 2.2.33** Napisati program koji računa sumu

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

za unete cele brojeve  $x$  i  $n$ . NAPOMENA: *Voditi računa o efikasnosti rešenja i o mogućnosti prekoračenja.*

*Primer 1*

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite x i n: 5.6 8
|| S=0.735084
```

*Primer 2*

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite x i n: 14.32 11
|| S=17273.136719
```

*Primer 3*

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite prirodan broj: -6
|| Neispravan unos.
```

\* **Zadatak 2.2.34** Program učitava ceo pozitivan broj  $n$  veći od 0. Napisati program koji računa proizvod

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

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

*Primer 1*

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

*Primer 2*

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

*Primer 3*

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

*Primer 4*

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

\* **Zadatak 2.2.35** Program učitava ceo pozitivan neparan broj  $n$ . Napisati program koji za uneto  $n$  izračunava:

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

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

*Primer 1*

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

*Primer 2*

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

*Primer 3*

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite broj n: 20
|| Neispravan unos
```

*Primer 4*

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

**Zadatak 2.2.36** Program učitava realne brojeve  $x$  i  $a$  i ceo pozitivan broj  $n$  veći od 0. Napisati program koji izračunava:

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

U slučaju greške pri unosu podataka ispisati odgovarajuću poruku.

*Primer 1*

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

*Primer 2*

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

*Primer 3*

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

*Primer 4*

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

**Zadatak 2.2.37** Za unetu pozitivnu celobrojnu vrednost  $n$  napisati programe koji ispisuju odgovarajuće brojeve. Pretpostaviti da je unos korektan.

- (a) Napisati program koji za unetu pozitivnu celobrojnu vrednost  $n$  ispisuje tablicu množenja.

*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: 3
1 2 3
2 4 6
3 6 9
```

### Primer 4

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

- (b) Napisati program koji za uneto  $n$  ispisuje sve brojeve od 1 do  $n^2$  pri čemu se ispisuje 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) Napisati program koji za uneto  $n$  ispisuje tablicu brojeva tako da su u prvoj vrsti svi brojevi od 1 do  $n$ , a svaka naredna vrsta dobija se 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) Napisati program koji za uneto  $n$  iscrtava pravougli „trougao“ sačinjen od „koordinata“ svojih tačaka. „Koordinata“ tačke je oblika  $(i, j)$  pri čemu  $i, j = 0, \dots, n$ . 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: 3
|| (0,0) (0,1) (0,2)
|| (1,0) (1,1)
|| (2,0)
```

### Primer 4

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

**Zadatak 2.2.38** Napisati program koji za unet prirodan broj  $n$  zvezdicama iscrtava odgovarajuću sliku. Pretpostaviti da je unos korektan.

- (a) Slika sadrži 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 sadrži rub kvadrata dimenzije  $n$ .

### Primer 1

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

### Primer 2

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

- (c) Slika sadrži 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
|| ****
|| ** *
|| * **
|| ****
```

## 2 Kontrola toka

---

\* **Zadatak 2.2.39** Napisati program koji za uneti prirodan broj  $n$  zvezdicama iscrtava slovo  $X$  dimenzije  $n$ . Pretpostaviti da je unos korektan.

*Primer 1*

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

*Primer 2*

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

\* **Zadatak 2.2.40** Napisati program koji za uneti prirodan neparan broj  $n$  korišćenjem znaka  $+$  iscrtava veliko  $+$  dimenzije  $n$ . Pretpostaviti da je unet prirodan broj.

*Primer 1*

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

*Primer 2*

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

*Primer 3*

```
INTERAKCIJA SA PROGRAMOM:
Unesite broj n: 4
Pogresan unos.
```

**Zadatak 2.2.41** Napisati program koji učitava prirodan broj  $n$ , a potom iscrtava odgovarajuću sliku. Pretpostaviti da je unos korektan.

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

### Primer 1

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

### Primer 2

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

- (f) Slika sadrži 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 2.2.42** Napisati program koji učitava ceo broj  $n$ , a potom iscrtava odgovarajuću sliku.

- (a) Slika sadrži 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 sadrži 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 sadrži 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 sadrži 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 2.2.43** Napisati program koji za uneti prirodan broj  $n$  iscrtava strelice dimenzije  $n$ . Pretpostaviti da je unos korektan.

Primer 1

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

Primer 2

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

\* **Zadatak 2.2.44** Napisati program koji učitava ceo broj  $n$ , i iscrtava 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  $n$ . Pretpostaviti da je unos korektan.

Primer 1

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

\* **Zadatak 2.2.45** Program učitava prirodne brojeve  $m$  i  $n$ . Napisati program koji iscrtava jedan do drugog stranice  $n$  kvadrata čija je svaka strana sastavljena od  $m$  zvezdica razdvojenih prazninom. Podrazumevati da je unos korektan.

Primer 1

```
INTERAKCIJA SA PROGRAMOM:
Unesite broj n: 5 3
*****
*       *       *       *
*       *       *       *
*       *       *       *
*       *       *       *
*****
```

Primer 2

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

\* **Zadatak 2.2.46** Program učitava prirodan broj  $n$ . Napisati program koji

štampa romb sastavljen od minusa u pravougaoniku sastavljenom od zvezdica. Podrazumevati da je unos korektan.

Primer 1

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

Primer 2

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

**Zadatak 2.2.47** Napisati program koji učitava ceo broj  $n$  ( $n \geq 2$ ) i koji iscrtava sliku kuće sa krovom: kuća je kocka stranice  $n$ , a krov jednakostranični trougao stranice  $n$ . Pretpostaviti da je unos korektan.

Primer 1

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

Primer 1

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

\* **Zadatak 2.2.48** Program učitava ceo pozitivan broj  $n$ . Napisati program koji ispisuje brojeve od 1 do  $n$ , zatim od 2 do  $n - 1$ , 3 do  $n - 2$ , itd. Ispis se završava kada nije moguće ispisati ni jedan broj. Za neispravan unos, program ispisuje odgovarajuću poruku. Pretpostaviti da je unos korektan.

Primer 1

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

Primer 2

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

Primer 3

```
INTERAKCIJA SA PROGRAMOM:
Unesite broj n: 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 2.2.49** Napisati program koji učitava ceo pozitivan broj  $n$  i ispisuje sve brojeve od 1 do  $n$ , zatim svaki drugi broj od 1 do  $n$ , zatim svaki treći broj od 1 do  $n$  itd., završavajući sa svakim  $n$ -tim (tj. samo sa 1). U slučaju greške pri unosu podataka odštampati ogovarajuću poruku.

*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
```

*Primer 4*

```
|| INTERAKCIJA SA PROGRAMOM:  
| Unesite broj n: -23  
| Neispravan unos.
```

## 2.3 Funkcije

**Zadatak 2.3.1** Napisati funkciju `int kvadrat(int x)` koja računa kvadrat datog broja. Napisati program koji učitava ceo broj i ispuje rezultat poziva funkcije.

*Primer 1*

```
|| INTERAKCIJA SA PROGRAMOM:  
| Unesite broj: 15  
| Kvadrat broja 15 je 225
```

*Primer 2*

```
|| INTERAKCIJA SA PROGRAMOM:  
| Unesite broj: -89  
| kvadrat broja -89 je 7921
```

**Zadatak 2.3.2** Napisati funkciju `int kub(int x)` koja računa kub datog broja. Napisati program koji učitava ceo broj i ispuje rezultat poziva funkcije.

*Primer 1*

```
|| INTERAKCIJA SA PROGRAMOM:  
| Unesite broj: 15  
| Kub broja 15 je 3375
```

*Primer 2*

```
|| INTERAKCIJA SA PROGRAMOM:  
| Unesite broj: -89  
| Kub broja -89 je -704969
```



**Zadatak 2.3.3** Napisati funkciju `unsigned int apsolutna_vrednost(int x)` koja izračunava apsolutnu vrednost broja  $x$ . Napisati program koji učitava jedan ceo broj i ispisuje rezultat poziva funkcije.

*Primer 1*

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

*Primer 2*

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

**Zadatak 2.3.4** Napisati funkciju `int min(int x, int y, int z)` koja izračunava minimum tri broja. Napisati program koji učitava tri cela broja i ispisuje rezultat poziva funkcije.

*Primer 1*

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

*Primer 2*

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

**Zadatak 2.3.5** Napisati funkciju `float razlomljeni_deo(float x)` koja izračunava razlomljeni deo broja  $x$ . Napisati program koji učitava jedan realan broj i ispisuje rezultat poziva funkcije.

*Primer 1*

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

*Primer 2*

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

**Zadatak 2.3.6** Napisati funkciju `float stepen(float x, int n)` koja računa vrednost  $n$ -tog stepena realnog broja  $x$ . Napisati program koji učitava realan broj  $x$  i ceo broj  $n$  i ispisuje rezultat rada funkcije.

*Primer 1*

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite jedan realan i jedan ceo broj:
|| 4.5 3
|| 4.5000003=91.125000
```

*Primer 2*

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite jedan realan i jedan ceo broj:
|| -33.2 5
|| -33.2000015=-40335776.000000
```

**Zadatak 2.3.7** 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 cela pozitivna broja i ispisuje rezultat poziva funkcije. NAPOMENA:

## 2 Kontrola toka

---

*Pretpostaviti da je unos korektan.*

### Primer 1

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesite dva broja: 81 3  
|| Jeste: 81 = 34
```

### Primer 2

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

**Zadatak 2.3.8** Napisati funkciju `int faktorijel(int n)` koja računa faktorijel broja  $n$ . Napisati i program koji učitava dva cela broja  $x$  i  $y$  iz intervala  $[0, 12]$  i ispisuje vrednost zbira  $x! + y!$ .

### Primer 1

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

### Primer 2

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

### Primer 3

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

**Zadatak 2.3.9** 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 rezultat poziva funkcije.

### Primer 1

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

### Primer 2

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

**Zadatak 2.3.10** 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 broj i ispisuje rezultat rada funkcije zaokružen na dve decimale.

### Primer 1

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesi jedan pozitivan ceo broj: 10  
|| Zbir reciprocnih je 2.93
```

### Primer 2

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesi jedan pozitivan ceo broj: 100  
|| Zbir reciprocnih je 5.19
```

**Zadatak 2.3.11** Napisati funkciju `void ispis(float x, float y, unsigned 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 odgovarajuće vrednosti i testira rad ove funkcije.

### Primer 1

```
INTERAKCIJA SA PROGRAMOM:
Unesite dva realna broja: 7 32
Unesite jedan prirodan broj: 10
0.6570 -0.3457 -0.0108 0.3659 -0.6731
0.8922 -0.9945 0.9666 -0.8122
```

### Primer 2

```
INTERAKCIJA SA PROGRAMOM:
Unesite dva realna broja: 20.5 -8.32
Unesite jedan prirodan broj: 5
-0.8934 -0.8979 -0.1920 0.6658 0.9968
```

**Zadatak 2.3.12** Napisati funkciju `float aritmeticka_sredina(int n)` koja računa aritmetičku sredinu cifara datog broja. Napisati i program koji učitava ceo broj i ispisuje rezultat na tri decimale.

### Primer 1

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

### Primer 2

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

### Primer 3

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

**Zadatak 2.3.13** 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 1 ako se cifra nalazi u broju, a 0 inače. Napisati program koji učitava dva cela broja i ispisuje rezultat poziva funkcije.

### Primer 1

```
INTERAKCIJA SA PROGRAMOM:
Unesite broj i cifru: 17890 7
Cifra se nalazi u broju.
```

### Primer 2

```
INTERAKCIJA SA PROGRAMOM:
Unesite broj i cifru: 1982 6
Cifra se ne nalazi u broju.
```

### Primer 3

```
INTERAKCIJA SA PROGRAMOM:
Unesite broj i cifru: 17890 26
Neispravan unos.
```

### Primer 4

```
INTERAKCIJA SA PROGRAMOM:
Unesite broj i cifru: -1982 9
Cifra se nalazi u broju.
```

**Zadatak 2.3.14** Napisati funkciju `int broj_neparnih_cifara(int x)` koja određuje broj neparnih cifre u zapisu datog celog broja. Testirati rad ove funkcije u programu koji učitava cele brojeve dok se ne unese nula i ispisuje broj neparnih cifara svakog unetog broja.

### Primer 1

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

### Primer 2

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

**Zadatak 2.3.15** Napisati program za ispitivanje svojstava cifara datog celog broja.

- (a) Napisati funkciju `sve_parne_cifre` koja ispituje da li se dati ceo broj sastoji isključivo iz parnih cifara. Funkcija treba da vrati 1 ako su sve cifre broja parne i 0 u suprotnom.
- (b) Napisati funkciju `sve_cifre_jednake` koja ispituje da li su sve cifre datog celog broja jednake. Funkcija treba da vrati 1 ako su sve cifre broja jednake i 0 u suprotnom.

Napisati program koji učitava ceo broj i ispisuje da li su sve cifre parne i da li su sve cifre jednake.

### 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
Sve cifre broja nisu parne.
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
Sve cifre broja nisu parne.
Cifre broja nisu jednake.
```

**Zadatak 2.3.16** Napisati funkciju `int zapis(int x, int y)` koja proverava da li se brojevi  $x$  i  $y$  zapisuju pomoću istih cifara. Funkcija treba da vrati vrednost 1 ako je uslov ispunjen, a 0 ako nije. Napisati i program koji učitava dva cela broja i ispisuje rezultat primene funkcije.

### Primer 1

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

### Primer 2

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

### Primer 3

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

**Zadatak 2.3.17** Napisati funkciju `int rastuce(int n)` koja ispituje da li su cifre datog celog broja u rastućem poretku. Funkcija treba da vrati vrednost 1 ako cifre ispunjavaju uslov, odnosno 0 ako ne ispunjavaju uslov. Napisati i program koji učitava ceo broj i ispisuje poruku da li su cifre unetog broja u rastućem poretku.

### Primer 1

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

### Primer 2

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

### Primer 3

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

**Zadatak 2.3.18** Napisati funkciju `int par_nepar(int n)` koja ispituje da li su cifre datog celog broja naizmenično parne i neparne. Funkcija treba da vrati vrednost 1 ako cifre ispunjavaju uslov, odnosno 0 ako ne ispunjavaju uslov. Napisati i program koji učitava ceo broj i testira rad funkcije.

### Primer 1

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

### Primer 2

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

### Primer 3

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

**Zadatak 2.3.19** Napisati funkciju `int ukloni_stotine(int n)` koja modifikuje zadati broj tako što iz njegovog zapisa uklanja cifru stotina (ako postoji). Napisati program koji za brojeve koji se unose sve do pojave broja 0 ispisuje odgovarajuće brojeve kojima je uklonjena cifra stotine.

*Primer 1*

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

*Primer 2*

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

**Zadatak 2.3.20** Napisati funkciju `int rotacija(int n)` koja rotira cifre zadatog broja za jednu poziciju u levo. Napisati program koji za brojeve koji se unose sve do pojave broja 0 ispisuje odgovarajuće rotirane brojeve.

*Primer 1*

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

*Primer 2*

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

**Zadatak 2.3.21** Napisati funkciju `int zbir_delilaca(int n)` koja izračunava zbir delilaca broja  $n$ . Napisati program koji učitava ceo broj  $k$  i ispisuje zbir delilaca svakog broja od 1 do  $k$ .

*Primer 1*

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

*Primer 2*

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

**Zadatak 2.3.22** Broj je prost ako je deljiv samo sa 1 i sa samim sobom. Napisati funkciju `int prost (int x)` koja ispituje da li je dati ceo broj prost. Funkcija treba da vrati 1 ako je broj prost i 0 u suprotnom. Napisati program koji za uneti ceo broj  $n$  ispisuje prvih  $n$  prostih brojeva.

*Primer 1*

```

|| INTERAKCIJA SA PROGRAMOM:
|| Unesite broj: 17
|| Broj je prost!

```

*Primer 2*

```

|| INTERAKCIJA SA PROGRAMOM:
|| Unesite broj: 24
|| Broj nije prost!

```

*Primer 3*

```

|| INTERAKCIJA SA PROGRAMOM:
|| Unesite broj: -11
|| Broj nije prost!

```

**Zadatak 2.3.23** Napisati funkciju `void prosti_brojevi(int m)` koja ispisuje sve proste brojeve manje od broja  $m$ . Napisati program koji učitava ceo broj veći od 1 i ispisati rezultat poziva funkcije. U slučaju pogrešnog unosa, ispisati poruku o grešci.

*Primer 1*

```

|| INTERAKCIJA SA PROGRAMOM:
|| Unesite broj: 15
|| 2 3 5 7 11 13

```

*Primer 2*

```

|| INTERAKCIJA SA PROGRAMOM:
|| Unesite broj: 9
|| 2 3 5 7

```

*Primer 3*

```

|| INTERAKCIJA SA PROGRAMOM:
|| Unesite broj: 1
|| Greska: pogresan unos!

```

**Zadatak 2.3.24** 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 vrši dok je razlika sabiraka u redu po apsolutnoj vrednosti manja od  $\varepsilon$ . Napisati program koji učitava dva realna broja  $x$  i  $eps$  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 2.3.25** 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 se dati niz završava jedinicom. Napisati funkciju `int srecan(int x)` koja vraća 1 ako je broj srećan, a 0 u suprotnom. Napisati program koji za uneti prirodan

## 2 Kontrola toka

---

broj  $n$  ispisuje sve srećne brojeve od 1 do  $n$ . NAPOMENA: *Pretpostaviti da je unos korektan.*

### Primer 1

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

### Primer 2

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesite broj: 0  
|| Nema srećnih brojeva.
```

**Zadatak 2.3.26** 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 1 ako je broj Armstrongov, odnosno 0 ako nije. Napisati program koji za učitani ceo broj proverava da li je Armstrongov.

### Primer 1

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

### Primer 2

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

### Primer 3

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

**Zadatak 2.3.27** Napisati funkciju `int prebrojavanje(float x)` koja prebrojava koliko puta se broj  $x$  pojavljuje u nizu brojeva koji se unose sve do pojave broja 0. Napisati program koji učitava vrednost broja  $x$  i testira rad napisane funkcije.

### Primer 1

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

### Primer 2

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

**Zadatak 2.3.28** Napisati funkciju `long unsigned fibonacci(int n)` koja računa  $n$ -ti element Fibonačijevog niza. Napisati i program koji učitava ceo broj  $n$  ( $0 \leq n \leq 50$ ) i ispisuje traženi Fibonačijev broj.



### Primer 1

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

### Primer 2

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

**Zadatak 2.3.29** Napisati funkciju `int konverzija (int c)` koja prebacuje veliko slovo u ekvivalentno malo i obrnuto. Napisati program koji testira ovu funkciju na karakterima koji se unose do pojave znaka EOF.

### Primer 1

```
INTERAKCIJA SA PROGRAMOM:
Unesite karaktere: ZDRAVO
zdravo
```

### Primer 2

```
INTERAKCIJA SA PROGRAMOM:
Unesite karaktere: Dobro jutro, R2D2!
dOBRO JUTRO, r2d2!
```

**Zadatak 2.3.30** Napisati funkciju `char sifra(char c, int k)` koja za dati karakter  $c$  određuje šifru na sledeći način: ukoliko je  $c$  slovo, šifra je karakter koji se nalazi  $k$  pozicija ispred njega u 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 karakter po karakter do kraja ulaza i ispisuje šifrovani tekst.

### Primer 1

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

**Zadatak 2.3.31** Napisati funkciju `int prestupna(int godina)` koja za zadatu godinu proverava da li je prestupna. Funkcija treba da vrati 1 ako je godina prestupna ili 0 ako nije. Napisati program koji učitava dva cela broja  $g1$  i  $g2$  i ispisuje sve godine iz intervala  $[g1, g2]$  koje su prestupne.

### Primer 1

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

### Primer 2

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

### Primer 3

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

### Primer 4

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

**Zadatak 2.3.32** Napisati funkciju `int broj_dana(int mesec, int godina)` koja za dati mesec i godinu vraća broj dana u datom mesecu. Napisati program koji učitava dva cela broja (mesec i godinu) i ispisuje broj dana u datom mesecu. U slučaju nekorektnog unosa ispisati odgovarajuću poruku o grešci.

### Primer 1

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesite mesec i godinu: 8  
|| 1998  
|| Broj dana je: 31
```

### Primer 2

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesite mesec i godinu: 2  
|| 2004  
|| Broj dana je: 29
```

### Primer 3

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesite mesec i godinu: 24  
|| 2004  
|| Neispravan unos.
```

**Zadatak 2.3.33** Napisati funkciju `int ispravan(int dan, int mesec, int godina)` koja za dati datum proverava da li je ispravan. Napisati program koji učitava tri cela broja (dan, mesec, godinu) i ispisuje da li je datum ispravan ili ne.

### Primer 1

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesite datum: 24.8.1998.  
|| Datum je ispravan.
```

### Primer 2

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesite datum: 31.4.1789.  
|| Datum nije ispravan.
```

### Primer 3

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesite datum: 29.2.2004.  
|| Datum je ispravan.
```

### Primer 4

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesite datum: 29.14.2004.  
|| Datum nije ispravan.
```

**Zadatak 2.3.34** Napisati funkciju `void sledeci_dan(int dan, int mesec, int godina)` koja za dati datum određuje datum sledećeg dana. Napisati program koji učitava tri cela broja i ispisuje datum sledećeg dana.

### 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.  
| Datum nije ispravan.
```

**Zadatak 2.3.35** 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 i ispisuje koliko dana je proteklo od Nove godine.

### 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.  
| Datum nije ispravan.
```

**Zadatak 2.3.36** 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 i ispisuje broj dana do kraja godine.

### 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 od Nove godine je: 0
```

### Primer 3

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

### Primer 4

```
|| INTERAKCIJA SA PROGRAMOM:  
| Unesite datum: 31.4.2004.  
| Datum nije ispravan.
```

**Zadatak 2.3.37** Napisati funkciju `int broj_dana_između(int dan1, int mesec1, int godina1, int dan2, int mesec2, int godina2)` koja određuje broj dana između dva datuma. Napisati program koji učitava šest celih brojeva, koji označavaju dva datuma, i na standardni izlaz ispisuje broj dana između ta dva datuma. U slučaju pogrešnog unosa ispisati poruku o grešci.

## 2 Kontrola toka

---

### 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: 104319
```

### Primer 3

```
INTERAKCIJA SA PROGRAMOM:
Unesite prvi datum: 24.8.1998.
Unesite drugi datum: 12.10.2010.
Broj dana izmedju dva datuma je: 4432
```

### Primer 4

```
INTERAKCIJA SA PROGRAMOM:
Unesite prvi datum: 24.8.1998.
Unesite drugi datum: 31.4.2004.
Ulaz nije ispravan.
```

**Zadatak 2.3.38** Napisati funkciju `void romb(int n)` koja iscrtava romb čija je stranica dužine  $n$ . Napisati program koji učitava ceo pozitivan broj i ispisuje rezultat poziva funkcije.

### Primer 1

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

### Primer 2

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

### Primer 3

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

**Zadatak 2.3.39** 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 prikazuje rezultat poziva funkcije. U slučaju pogrešnog unosa, ispisati poruku o grešci.

### Primer 1

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

### Primer 2

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

### Primer 3

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

**Zadatak 2.3.40** 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 ispisuje rezultat poziva funkcije. U slučaju pogrešnog unosa, ispisati poruku o grešci.

### Primer 1

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

### Primer 2

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

### Primer 3

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



# 3

## Predstavljanje podataka

### 3.1 Nizovi

**Zadatak 3.1.1** Ako su  $a = (a_1, \dots, a_n)$  i  $b = (b_1, \dots, b_n)$  vektori dimenzije  $n$ , njihov skalarni proizvod je  $a \cdot b = a_1 \cdot b_1 + \dots + a_n \cdot b_n$ . Napisati program koji računa skalarni proizvod dva vektora. Vektori se zadaju kao celobrojni nizovi sa najviše 100 elemenata. Program učitava dimenziju i elemente nizova, a na izlaz ispisuje vrednost skalarnog proizvoda.

#### Primer 1

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

#### Primer 2

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

#### Primer 3

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

**Zadatak 3.1.2** Napisati program koji za učitani niz ispisuje:

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

### 3 Predstavljanje podataka

---

(b) parne elemente niza.

Pretpostaviti da je dimenzija niza broj koji nije veći od 100.

#### Primer 1

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

#### Primer 2

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

#### Primer 3

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

**Zadatak 3.1.3** Napisati program koji za učitani ceo broj, ispisuje broj pojavljivanja svake od cifara u zapisu tog broja. UPUTSTVO: Za evidenciju broja pojavljivanja svake cifre pojedinačno, koristiti niz.

#### Primer 1

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

#### Primer 2

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

**Zadatak 3.1.4** Napisati program koji za dva cela broja  $x$  i  $y$  koja se učitavaju sa standardnog ulaza proverava da li se 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!
```

**Zadatak 3.1.5** Napisati program koji učitava karaktere sa standardnog ulaza sve do kraja ulaza i izračunava koliko se puta u unetom tekstu pojavila svaka od cifara, svako malo slovo i svako veliko slovo. Ispisati broj pojavljivanja samo za karaktere koji su se u unetom tekstu pojavili barem jednom. UPUTSTVO: Za evidenciju broja pojavljivanja cifara, malih i velikih slova koristiti pojedinačne nizove.

*Primer 1*

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

**Zadatak 3.1.6** Sa standardnog ulaza se unosi jedna linija teksta. Napisati program koji izračunava i ispisuje koliko puta se pojavilo svako od slova engleskog alfabeta u unetom tekstu. Ne praviti razliku između malih i velikih slova.

*Primer 1*

```
INTERAKCIJA SA PROGRAMOM:
haHJjKl
a:1 b:0 c:0 d:0 e:0 f:0 g:0 h:2 i:0 j:2 k:1 l:1 m:0
n:0 o:0 p:0 q:0 r:0 s:0 t:0 u:0 v:0 w:0 x:0 y:0 z:0
```

*Primer 2*

```
INTERAKCIJA SA PROGRAMOM:
DanaS j3 _j_utRo laBU78d
a:1 b:0 c:0 d:0 e:0 f:0 g:0 h:2 i:0 j:2 k:1 l:1 m:0
n:0 o:0 p:0 q:0 r:0 s:0 t:0 u:0 v:0 w:0 x:0 y:0 z:0
```

### 3 Predstavljanje podataka

---

#### Primer 3

```
INTERAKCIJA SA PROGRAMOM:
Sao Paolo 1998 _JuZna Amerika90
a:5 b:0 c:0 d:2 e:1 f:0 g:0 h:0 i:1 j:1 k:1 l:1 m:1
n:1 o:3 p:1 q:0 r:1 s:1 t:0 u:1 v:0 w:0 x:0 y:0 z:0
```

**Zadatak 3.1.7** Napisati program koji za dva učitana niza  $a$  i  $b$  dimenzije  $n$  formira i na izlaz ispisuje niz  $c$  koji se dobija naizmeničnim raspoređivanjem elemenata nizova  $a$  i  $b$ , tj.  $c = [a_0, b_0, a_1, b_1, \dots, a_{n-1}, b_{n-1}]$ . Pretpostaviti da dimenzija učitanih nizova nije veća od 100.

#### Primer 1

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

#### Primer 2

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

**Zadatak 3.1.8** Sa standardnog ulaza se učitava ceo broj  $n$  (manji od 100) i elementi dva niza  $a$  i  $b$  dimenzije  $n$ . Napisati program koji formira i ispisuje niz  $c$  čiju prvu polovinu čine elementi niza  $b$ , a drugu polovinu elementi niza  $a$ .

#### Primer 1

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

#### Primer 2

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

#### Primer 3

```
INTERAKCIJA SA PROGRAMOM:
Unesite dimenziju niza: 145
Greska: Nedozvoljena vrednost!
```

**Zadatak 3.1.9** Napisati program koji sa standardnog ulaza učitava 10 celih brojeva i razdvaja ih na parne i neparne tako što parne brojeve upisuje na početak niza, a neparne brojeve na kraj niza. Ispisati niz dobijen na ovaj način. NAPOMENA: *Nije dozvoljeno koristiti pomoćne nizove.*

#### Primer 1

```
INTERAKCIJA SA PROGRAMOM:
Unesite 10 brojeva:
-2 8 11 53 59 20 17 -8 3 14
Rezultujući niz:
-2 8 20 -8 14 3 17 59 53 11
```

#### Primer 2

```
INTERAKCIJA SA PROGRAMOM:
Unesite 10 brojeva:
9 142 -9 -278 -69 33 34 28 -6 14
Rezultujući niz:
142 -278 34 28 -6 14 33 -69 -9 9
```

**Zadatak 3.1.10** Napisati program koji učitava dimenziju  $n$  celobrojnog niza  $a$  i njegove elemente, i iz niza  $a$  izbacuje sve elemente koji nisu deljivi svojom poslednjom cifrom. Izuzetak su elementi čija je poslednja cifra 0 koje treba zadržati. Program treba da ispiše izmenjeni niz na standardni izlaz. Niz  $a$  sadrži najviše 100 elemenata.

#### Primer 1

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

#### Primer 2

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

**Zadatak 3.1.11** Napisati program koji u nizu dužine  $n$  (broj manji od 100) čiji se elementi učitavaju sa ulaza eliminiše sve brojeve koji nisu deljivi svojim indeksom. Niz reorganizovati tako da nema *rupa* koje su nastale eliminacijom elemenata i ispisati na standardni izlaz. NAPOMENA: *Nulti element niza treba zadržati jer nije dozvoljeno deljenje nulom.*

#### Primer 1

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

#### Zadatak 3.1.12

Napisati funkcije za rad sa nizovima celih brojeva.

- Napisati funkciju `void ucitaj(int a[], int n)` koja učitava elemente niza  $a$  dimenzije  $n$ .
- Napisati funkciju `void stampaj(int a[], int n)` koja štampa elemente niza  $a$  dimenzije  $n$ .

### 3 Predstavljanje podataka

---

- (c) Napisati funkciju `int suma(int a[], int n)` koja računa i vraća sumu elemenata niza  $a$  dimenzije  $n$ .
- (d) Napisati funkciju `int prosek(int a[], int n)` koja računa i vraća prosečnu vrednost (aritmetičku sredinu) elemenata niza  $a$  dimenzije  $n$ .
- (e) Napisati funkciju `int minimum(int a[], int n)` koja izračunava i vraća minimum elemenata niza  $a$  dimenzije  $n$ .
- (f) Napisati funkciju `int pozicija_maksimuma(int a[], int n)` koja izračunava i vraća poziciju maksimalnog elementa u nizu  $a$  dimenzije  $n$ . U slučaju više pojavljivanja maksimalnog elementa, vratiti najmanju poziciju.

Napisati program koji testira rad zadatih funkcija. Sa standardnog ulaza učitati dimenziju niza (broj ne veći od 100).

#### *Primer 1*

```
INTERAKCIJA SA PROGRAMOM:
Unesite dimenziju niza:
5
2 5 -2 8 11
Ucitani niz: 2 5 -2 8 11
Suma elemenata niza: 24
Prosečna vrednost elemenata niza: 4.80
Minimalni element niza: -2
Indeks maksimalnog elementa niza: 4
```

#### *Primer 2*

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

#### **Zadatak 3.1.13**

Napisati funkcije za rad sa nizovima celih brojeva.

- (a) Napisati funkciju koja proverava da li niz sadrži zadatu vrednost  $m$ . Povratna vrednost funkcije je 1 ako je vrednost sadržana u nizu ili 0 ako nije.
- (b) Napisati funkciju koja vraća vrednost prve pozicije na kojoj se nalazi element koji ima vrednost  $m$  ili  $-1$  ukoliko element nije u nizu.
- (c) Napisati funkciju koja vraća vrednost poslednje pozicije na kojoj se nalazi element koji ima vrednost  $m$  ili  $-1$  ukoliko element nije u nizu.
- (d) Napisati funkciju koja proverava da li elementi niza čine palindrom.
- (e) Napisati funkciju koja proverava da li su elementi niza uređeni neopadajuće.

Napisati i program koji testira rad napisanih funkcija za uneti broj  $m$  i niz čija dimenzija nije veća od 100.

*Primer 1*

```

INTERAKCIJA SA PROGRAMOM:
Unesite dimenziju niza:
7
8 11 -2 14 -2 11 8
Ucitani niz: 8 11 -2 14 -2 11 8
Unesite jedan ceo broj:
11
Niz sadrzi element cija je vrednost 11.
Niz sadrzi element cija je vrednost 11.
Indeks njegovog prvog pojavljivanja u nizu je 1.
Niz sadrzi element cija je vrednost 11.
Indeks njegovog poslednjeg pojavljivanja u nizu je 5.
Elementi niza cine palindrom.
Niz nije sortiran neopadajuće.

```

*Primer 2*

```

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

```

**Zadatak 3.1.14** Napisati funkcije za rad sa nizovima celih brojeva.

- Napisati funkciju koja sve vrednosti niza uvećava za zadatu vrednost  $m$ .
- Napisati funkciju koja obrće elemente niza.
- Napisati funkciju koja rotira niz ciklično za jedno mesto u levo.
- Napisati funkciju koja rotira niz ciklično za  $k$  mesta u levo.

Napisati i program koji testira rad napisanih funkcija za uneti broj  $m$  i niz čija dimenzija nije veća od 100.

*Primer 1*

```

INTERAKCIJA SA PROGRAMOM:
Unesite dimenziju niza:
6
7 -3 11 783 26 -19
Unesite jedan ceo broj:
2
Elementi niza nakon uvecanja za 2:
9 -1 13 785 28 -17
Elementi niza nakon obrtanja:
-17 28 785 13 -1 9
Elementi niza nakon rotiranja za 1 mesto ulevo:
28 785 13 -1 9 -17
Unesite jedan pozitivan ceo broj:
3
Elementi niza nakon rotiranja za 3 mesto ulevo:
-1 9 -17 28 785 13

```

*Primer 2*

```

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

```

### 3 Predstavljanje podataka

---

**Zadatak 3.1.15** Napisati program koji transformiše uneti niz tako što kvadrira sve negativne elemente niza. Pretpostaviti da je dimenzija niza broj koji nije veći od 100.

#### Primer 1

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

#### Primer 2

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

#### Primer 3

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

**Zadatak 3.1.16** Sa standardnog ulaza se učitava dimenzija niza, elementi niza i jedan ceo broj  $k$ . Napisati program koji štampa indekse elemenata koji su deljivi sa  $k$ . Pretpostaviti da je dimenzija niza broj koji nije veći od 100.

#### Primer 1

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

#### Primer 2

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

#### Primer 3

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

**Zadatak 3.1.17** Napisati program koji učitava dimenziju i elemente niza i štampa niz u kojem su najveći i najmanji element niza razmenili mesta. Ukoliko se najmanji ili najveći element više puta pojavljuju u nizu, uzeti u obzir njihova prva pojavljivanja. Pretpostaviti da je dimenzija niza broj koji nije veći od 100.

*Primer 1*

```

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

```

*Primer 2*

```

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

```

*Primer 3*

```

INTERAKCIJA SA PROGRAMOM:
Unesite dimenziju niza: 145
Greska: Nedozvoljena vrednost!

```

**Zadatak 3.1.18** Napisati funkciju `int min_max(int a[], int n)` koja pronalazi indekse najmanjeg i najvećeg elementa u nizu  $a$  dimenzije  $n$  koristeći samo jedan prolaz kroz niz. Funkcija kao povratnu vrednost vraća manji od ta dva indeksa. Napisati program koji testira ovu funkciju za učitane nizove celih brojeva maksimalne dužine 100 elemenata.

*Primer 1*

```

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

```

*Primer 2*

```

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

```

*Primer 3*

```

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

```

**Zadatak 3.1.19** Napisati program koji učitane karaktere (najviše njih 100, učitavaju se sve do pojave karaktera `*`) ispisuje u redosledu suprotnom od redosleda čitanja.

### 3 Predstavljanje podataka

---

#### Primer 1

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

#### Primer 2

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

#### Primer 3

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

#### Zadatak 3.1.20

Sa standardnog ulaza se unosi broj elemenata niza  $a$  i njegovi elementi. Napisati program koji od datog niza formira niz  $b$  u koji ulaze elementi niza  $a$  koji se pojavljuju tačno tri puta. Pretpostaviti da će uneti niz imati najviše 100 elemenata. Pretpostaviti da je dimenzija niza broj koji nije veći od 100.

#### Primer 1

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

#### Primer 2

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

#### Primer 3

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

**Zadatak 3.1.21** Napisati funkciju `int sadrzi_bar_dva(int a[], int na, int b[], int nb)` koja proverava da li niz  $a$  dužine  $na$  sadrži barem dva broja koja se pojavljuju u nizu  $b$  dužine  $nb$ . Napisati i program koji učitava redom dimenzije i elemente nizova  $a$  i  $b$  i ispisuje da li uneti nizovi ispunjavaju traženo svojstvo. Pretpostaviti da će uneti nizovi imati najviše 100 elemenata.



*Primer 1*

```

INTERAKCIJA SA PROGRAMOM:
Unesite broj elemenata niza a:
5
Unesite elemente niza a:
5 8 7 -2 6
Unesite broj elemenata niza b:
6
Unesite elemente niza b:
11 -11 7 -7 6
Svojstvo je ispunjeno.

```

**Zadatak 3.1.22** Sa standardnog ulaza se, redom, učitavaju dimenzije i elementi dva niza,  $a$  i  $b$ . Napisati program koji određuje i ispisuje njihovu uniju, presek i razliku (redosled prikaza elemenata nije bitan). Pretpostaviti da će uneti nizovi imati najviše 100 elemenata.

*Primer 1*

```

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

```

*Primer 2*

```

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

```

*Primer 3*

```

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

```

**Zadatak 3.1.23** Napisati program koji za učitani niz formira i ispisuje niz koji se dobija izbacivanjem svih neparnih elemenata niza. Zadatak rešiti na dva načina: korišćenjem pomoćnog niza i transformacijom polaznog niza. Pretpostaviti da će uneti niz imati najviše 100 elemenata.

*Primer 1*

```

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

```

*Primer 2*

```

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

```

### 3 Predstavljanje podataka

---

#### Primer 3

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

#### Primer 4

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

**Zadatak 3.1.24** Napisati program koji za učitani niz ispisuje niz koji se dobija izbacivanjem svih elemenata koji su prosti brojevi. Zadatak rešiti na dva načina: korišćenjem pomoćnog niza i transformacijom polaznog niza. Pretpostaviti da će uneti niz imati najviše 100 elemenata. NAPOMENA: *Broj 1 nije prost.*

#### Primer 1

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

#### Primer 2

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

#### Primer 3

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

#### Primer 4

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

#### Primer 5

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

**Zadatak 3.1.25** Napisati funkciju `int prebrojavanje(int a[], int n)` koja izračunava broj elemenata celobrojnog niza  $a$  dužine  $n$  koji su manji od poslednjeg elementa niza. Napisati i program koji testira rad funkcije. Pretpostaviti da dužina niza neće biti veća od 100.

*Primer 1*

```

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

```

*Primer 2*

```

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

```

*Primer 3*

```

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

```

**Zadatak 3.1.26** Napisati funkciju `int prebrojavanje(int a[], int n)` koja izračunava broj parnih elemenata niza celih brojeva  $a$  dužine  $n$  koji prethode maksimalnom elementu niza. Napisati i program koji testira rad funkcije. Pretpostaviti da dužina niza neće biti veća od 100.

*Primer 1*

```

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

```

*Primer 2*

```

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

```

*Primer 3*

```

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

```

**Zadatak 3.1.27** Napisati funkciju `int cifre(char s[], int n)` koja izračunava broj cifara u nizu karaktera  $a$  dužine  $n$ . Napisati program koji za karaktere koji se unose u zasebnim redovima ispisuje broj unetih cifara. Pretpostaviti da dužina niza neće biti veća od 100.

### 3 Predstavljanje podataka

---

#### Primer 1

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

#### Primer 2

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

#### Primer 3

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

**Zadatak 3.1.28** Napisati funkciju `int zbir(int a[], int n, int i, int j)` koja računa zbir elemenata niza celih brojeva  $a$  dužine  $n$  od pozicije  $i$  do pozicije  $j$ . Napisati i program koji testira rad funkcije. Pretpostaviti da dužina niza neće biti veća od 100.

#### Primer 1

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

#### Primer 2

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

#### Primer 3

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

**Zadatak 3.1.29** Napisati funkciju `float zbir_pozitivnih(float a[], int n, int k)` koja izračunava zbir prvih  $k$  pozitivnih elemenata realnog niza  $a$  dužine  $n$ . Napisati i program koji testira rad funkcije. Pretpostaviti da dužina niza neće biti veća od 100.

*Primer 1*

```

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

```

*Primer 2*

```

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

```

*Primer 3*

```

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

```

**Zadatak 3.1.30** Napisati funkciju `void kvadriranje(float a[], int n)` koja kvadrira elemente realnog niza  $a$  dužine  $n$  koji se nalaze na parnim pozicijama. Napisati program koji transformiše na ovaj način uneti niz. Pretpostaviti da dužina niza neće biti veća od 100.

*Primer 1*

```

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

```

*Primer 2*

```

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

```

*Primer 3*

```

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

```

**Zadatak 3.1.31** Napisati funkciju `int blizu_3(int a[], int n)` koja pronalazi i vraća indeks elementa niza koji je po vrednosti najbliži aritmetičkoj sredini onih elemenata niza koji su deljivi brojem tri. Napisati program koji testira rad funkcije. Pretpostaviti da dužina niza neće biti veća od 100.

### 3 Predstavljanje podataka

---

#### Primer 1

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

#### Primer 2

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

**Zadatak 3.1.32** Napisati funkcije za rad sa nizovima celih brojeva.

- (a) Napisati funkciju koja izbacuje poslednji element niza.
- (b) Napisati funkciju koja izbacuje prvi element niza. Zadatak rešiti na dva načina: čuvanjem redosleda elemenata i premeštanjem poslednjeg elementa niza na upražnjenu poziciju.
- (c) Napisati funkciju koja izbacuje element sa date pozicije  $k$ .
- (d) Napisati funkciju koja izbacuje sva pojavljivanja datog elementa  $x$  iz niza.
- (e) Napisati funkciju koja ubacuje dati element  $x$  na kraj niza.
- (f) Napisati funkciju koja ubacuje dati element  $x$  na početak niza.
- (g) Napisati funkciju koja ubacuje dati element  $x$  na datu poziciju  $k$ .

Napisati program koji testira rad zadatih funkcija. Sa standardnog ulaza učitati dimenziju niza (broj ne veći od 100).

#### Primer 1

```
INTERAKCIJA SA PROGRAMOM:
Unesite dimenziju niza:
8
2 5 -2 16 33 19 8 11
Niz posle izbacivanja poslednjeg elementa: 2 5 -2 16 33 5 8
Niz nakon izbacivanja prvog elementa: 5 -2 16 33 5 8
Unesite poziciju elementa za izbacivanje:
3
Niz nakon izbacivanja 3. elementa: 5 -2 16 5 8
Unesite element cije pojavljivanje treba izbaciti:
5
Niz nakon izbacivanja elementa 5: -2 16 8
Unesite element koji treba ubaciti u niz:
19
Niz nakon ubacivanja elementa 19 na kraj: -2 16 8 19
Niz nakon ubacivanja elementa 19 na pocetak: 19 -2 16 8 19
Unesite poziciju na koju treba ubaciti element:
2
Niz nakon ubacivanja elementa 19 na poziciju 2: 19 -2 19 16 8 19
```

**Zadatak 3.1.33** Napisati funkcije za rad sa nizovima celih brojeva.

- Napisati funkciju koja određuje dužinu najduže serije jednakih uzastopnih elemenata u datom nizu brojeva.
- Napisati funkciju koja određuje dužinu najvećeg neopadajućeg podniza datog niza celih brojeva.
- Napisati funkciju koja određuje da li se jedan niz javlja kao podniz uzastopnih elemenata drugog niza.
- Napisati funkciju koja određuje da li se jedan niz javlja kao podniz elemenata drugog niza (elementi ne moraju da budu uzastopni, ali je redosled pojavljivanja isti).
- Napisati funkciju koja izbacuje višestruka pojavljivanja elemenata iz datog niza brojeva. Zadatak rešiti na dva načina: zadržavanjem prvog pojavljivanje elementa i zadržavanjem poslednjeg pojavljivanje elementa.

**Zadatak 3.1.34** Napisati funkciju koja iz zadatog niza izbacuje sve elemente koji su deljivi svojim indeksom. Niz reorganizovati tako da nema *rupa* koje su nastale izbacivanjem elemenata. Povratna vrednost funkcije je nova dimenzija niza. Napisati program koji za učitni niz (dimenzije manje od 100) ispisuje niz dobijen nakon poziva funkcije. NAPOMENA: *Element na nultoj poziciji niza zadržati jer nije dozvoljeno deljenje nulom.*

*Primer 1*

```
INTERAKCIJA SA PROGRAMOM:
Unesite broj elemenata niza: 10
Unesite elemente niza:
4 2 1 6 7 10 8 2 16 27
4 1 7 8 2 16
```

**Zadatak 3.1.35** Za celobrojni niz  $a$  dimenzije  $n$  kažemo da je *permutacija* ako sadrži sve brojeve od 1 do  $n$ .

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

### 3 Predstavljanje podataka

---

Napisati program koji sa standardnog ulaza učitava dimenziju niza (broj manji od 100) i elemente niza i ispisuje da li je uneti niz permutacija ili ne.

#### Primer 1

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

#### Primer 2

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

## 3.2 Pokazivači

**Zadatak 3.2.1** Napisati funkciju koja uređuje svoja dva celobrojna argumenta tako da se u prvom nalazi manji, a u drugom veći. Napisati program koji učitava dva cela broja i ispisuje rezultat poziva funkcije.

#### Primer 1

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesite vrednosti promenljivih x i y: 2 5  
|| Uredjene promenljive: x=2, y=5
```

#### Primer 2

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesite vrednosti promenljivih x i y: 11 -4  
|| Uredjene promenljive: x=-4, y=11
```

**Zadatak 3.2.2** Napisati funkciju koja za boju datu u *rgb* formatu računa *cmy* format po formulama:

$$c = 1 - (r/255)$$

$$m = 1 - (g/255)$$

$$y = 1 - (b/255)$$

Napisati program koji učitava tri cela broja broja (*rgb* format) i ispisuje rezultat poziva funkcije (*cmy* format). NAPOMENA: Vrednosti boja u *rgb* formatu su u opsegu  $[0, 255]$ .

#### Primer 1

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesite boju u rgb formatu: 56 111 24  
|| c=0.78, m=0.56, y=0.91
```

#### Primer 2

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesite boju u rgb formatu: 156 -90 5  
|| Nekorektan unos.
```

#### Primer 3

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesite boju u rgb formatu: 9 0 237  
|| c=0.96, m=1.00, y=0.07
```

#### Primer 4

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesite boju u rgb formatu: 300 11 27  
|| Nekorektan unos.
```



**Zadatak 3.2.3** Napisati funkciju koja za dve prave date svojim koeficijentima pravca i slobodnim članovima određuje njihovu tačku preseka. Funkcija treba da vrati 1 ako se prave seku i 0 ako nemaju tačku preseka (ako su paralelne). Napisati program koji učitava podatke o pravama, poziva napisanu funkciju i 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.
```

**Zadatak 3.2.4** Napisati funkciju koja za dva cela broja izračunava njihov količnik i ostatak pri deljenju. Funkcija treba da vrati vrednost 1 ukoliko je uspešno izračunala vrednosti i 0 ukoliko deljenje nije moguće. Napisati program koji testira rad ove funkcije.

**Zadatak 3.2.5** Napisati funkciju koja za dinarsku vrednost cene artikla izračunava odgovarajuću cenu u evrima i u dolarima. Napisati program koji testira rad ove funkcije.

**Zadatak 3.2.6** Napisati funkciju koja za dužinu trajanja filma kaja je data u sekundama, određuje ukupno trajanje filma u satima, minutama i sekundama. Napisati program koji testira rad ove funkcije.

**Zadatak 3.2.7** Napisati funkciju `void modifikacija(char* s, char* t, int* br_modifikacija)` koja na osnovu niske *s* formira nisku *t* tako što svako malo slovo zamenjuje velikim. Broj izvršenih modifikacija se čuva u okviru argumenta *br\_modifikacija*. Pretpostaviti da niska *s* neće biti duža od 20 karaktera. Napisati program koji testira rad napisane funkcije.

*Primer 1*

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

*Primer 2*

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

*Primer 3*

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

*Primer 4*

```
INTERAKCIJA SA PROGRAMOM:
Unesite nisku: 1234
Modifikovana niska je: 1234
Broj modifikacija je: 0
```

### 3 Predstavljanje podataka

---

**Zadatak 3.2.8** Napisati funkciju `void interpunkcija(int* br_tacaka, int* br_zareza)` koja prebrojava tačke i zareze u tekstu koji se unosi sa standardnog ulaza. Napisati program koji testira napisanu funkciju.

#### Primer 1

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

#### Primer 2

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

#### Primer 3

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

**Zadatak 3.2.9** Napisati funkciju `void par_nepar(int a[], int n, int parni[], int* pn, int neparni[], int* nn)` koja razbija niz  $a$  na niz parnih i niz neparnih brojeva. Pokazivači  $pn$  i  $nn$  redom treba da sadrže broj elemenata niza parnih tj. niza neparnih elemenata. Pretpostaviti da dužina niza  $a$  neće biti veća od 50. Napisati program koji testira napisanu funkciju.

#### Primer 1

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

#### Primer 2

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

#### Primer 3

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

#### Primer 4

```
INTERAKCIJA SA PROGRAMOM:
Unesite broj elemenata niza: 1
Unesite elemente niza:
0
Niz parnih brojeva: 0
Niz neparnih brojeva:
```

**Zadatak 3.2.10** Napisati funkciju `void min_max(float a[], int n, float* min, float* max)` koja izračunava minimalni i maksimalni element niza  $a$  dužine  $n$ . Napisati program koji učitava niz realnih brojeva maksimalne dužine 50 i ispisuje vrednosti minimuma i maksimuma zaokruženu na tri decimale.

*Primer 1*

```

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

```

*Primer 2*

```

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

```

*Primer 3*

```

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

```

*Primer 4*

```

INTERAKCIJA SA PROGRAMOM:
Unesite broj elemenata niza: 3
Unesite elemente niza:
7.82 18.989 7.82
Minimum: 7.820
Maksimum: 18.989

```

**Zadatak 3.2.11** Napisati program koji ispisuje broj navedenih argumenata komandne linije, a zatim i same argumenate i njihove redne brojeve.

*Primer 1*

```

POKRETANJE: ./a.out abcde 123 -5 3.7
INTERAKCIJA SA PROGRAMOM:
Broj argumenata je 5:
0: ./a.out
1: abcde
2: 123
3: -5
4: 3.7

```

*Primer 2*

```

POKRETANJE: ./a.out
INTERAKCIJA SA PROGRAMOM:
Broj argumenata je 1:
0: ./a.out

```

**Zadatak 3.2.12** Napisati program koji ispisuje zbir numeričkih argumenata komandne linije. UPUTSTVO: *Koristiti funkciju atoi.*

*Primer 1*

```

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

```

*Primer 2*

```

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

```

*Primer 3*

```

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

```

*Primer 4*

```

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

```

**Zadatak 3.2.13** Napisati program koji ispisuje argumente komandne linije koji počinju slovom z.

### 3 Predstavljanje podataka

---

#### Primer 1

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

#### Primer 2

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

#### Primer 3

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

#### Primer 4

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

**Zadatak 3.2.14** Napisati program koji ispisuje broj argumenata komandne linije koji sadrže slovo z.

#### Primer 1

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

#### Primer 2

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

#### Primer 3

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

#### Primer 4

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

**Zadatak 3.2.15** Napisati program koji na osnovu broja  $n$  koji se zadaje kao argument komandne linije ispisuje cele brojeve iz intervala  $[-n, n]$ .

#### Primer 1

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

#### Primer 2

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

#### Primer 3

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

#### Primer 4

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

**Zadatak 3.2.16** Napisati program koji proverava da li se među zadatim argumentima komandne linije nalaze barem dva ista.

*Primer 1*

```

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

```

*Primer 2*

```

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

```

*Primer 3*

```

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

```

*Primer 4*

```

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

```

**Zadatak 3.2.17** Napisati funkciju koja za dva data stringa određuje koliko se uzastopnih karaktera prvog stringa nalazi u drugom stringu počev od početka. Napisati program koji testira napisanu funkciju za dva stringa koji se unose kao argumenti komandne linije.

*Primer 1*

```

|| POKRETANJE: ./a.out aladin bal
|| INTERAKCIJA SA PROGRAMOM:
||   3

```

*Primer 2*

```

|| POKRETANJE: ./a.out aladin lad
|| INTERAKCIJA SA PROGRAMOM:
||   4

```

*Primer 3*

```

|| POKRETANJE: ./a.out Aladin ala
|| INTERAKCIJA SA PROGRAMOM:
||   0

```

*Primer 4*

```

|| POKRETANJE: ./a.out aladin
|| INTERAKCIJA SA PROGRAMOM:
||   Nekorektan poziv
||   Program treba pozvati sa ./a.out arg1 arg2

```

**Zadatak 3.2.18** Napisati program koji ispisuje sve opcije koje su navedene u komandnoj liniji.

*Primer 1*

```

|| POKRETANJE: ./a.out -abc input.txt -d -Fg output
|| INTERAKCIJA SA PROGRAMOM:
||   a b c d F g

```

*Primer 2*

```

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

```

*Primer 3*

```

|| POKRETANJE: ./a.out ulaz.txt
|| INTERAKCIJA SA PROGRAMOM:

```

*Primer 4*

```

|| POKRETANJE: ./a.out file.txt -x -yZ -g output
|| INTERAKCIJA SA PROGRAMOM:
||   x yZ g

```

**Zadatak 3.2.19** Napisati funkciju `void sifruj(char s[], char c, int k)` koja šifruje string `s` na sledeći način: svako malo i veliko slovo stringa `s` konvertuje u slovo koje je u abecedi od njega udaljeno `k` pozicija, i to `k` pozicija ulevo,

### 3 Predstavljanje podataka

---

ako je karakter c jednak karakteru 'L' ili udesno ako je karakter c jednak karakteru 'D'. Šifrovanje treba da bude kružno. Ako string s sadrži karakter koji nije alfanumerički, ostaviti ga nešifriranog. Napisati program koji testira napisanu funkciju za string i prirodan broj koji se unose kao argumenti komandne linije dok se pravac šifrovanja unosi kao opcija -p koja može imati vrednosti 'L' ili 'D'. Ukoliko opcija -p nije navedena, podrazumevani pravac je udesno. NAPOMENA: *Možemo podrazumevati da string sadrži najviše 30 karaktera.*

#### Primer 1

```
POKRETANJE: ./a.out abcd 2
INTERAKCIJA SA PROGRAMOM:
cdef
```

#### Primer 2

```
POKRETANJE: ./a.out abcd 2 -p D
INTERAKCIJA SA PROGRAMOM:
cdef
```

#### Primer 3

```
POKRETANJE: ./a.out abcd 2 -p L
INTERAKCIJA SA PROGRAMOM:
yzab
```

#### Primer 4

```
POKRETANJE: ./a.out abcd -3 -p L
INTERAKCIJA SA PROGRAMOM:
Nekorektan unos
```

#### Primer 5

```
POKRETANJE: ./a.out abcd 3 -p X
INTERAKCIJA SA PROGRAMOM:
Nekorektan unos
```

#### Primer 6

```
POKRETANJE: ./a.out ab12cd 2 -p D
INTERAKCIJA SA PROGRAMOM:
Nekorektan unos
```

**Zadatak 3.2.20** Parametri komandne linije su  $n, a$  i  $b$  ( $a < b$ ). Treba popuniti prvih  $n$  elemenata niza  $A$  celim slučajnim brojevima koji su između  $a$  i  $b$ . Ištampati niz  $A$  na standardni izlaz. Maksimalan broj elemenata niza  $A$  je 200. Ukoliko nisu zadati svi argumenti komandne linije ili ne zadovoljavaju potrebna svojstva ispisati poruku o grešci.

## 3.3 Niske

**Zadatak 3.3.1** Napisati funkciju koja konvertuje dati string tako sto mala slova menja u velika a velika u mala. Napisati potom glavni program koji ucitava string, poziva napisanu funkciju i ispisuje konvertovani string. Mozemo pretpostaviti da string ne sadrzi vise od 10 karaktera.

**Zadatak 3.3.2** Napisati funkciju skрати koja uklanja beline sa kraja datog stringa. Napisati program koji testira napisanu funkciju na stringu "rep belina".

**Zadatak 3.3.3** Napisati program koji ucitava string src i formira string dst

trostrukim nadovezivanjem stringa `src`. Program treba da ispise string `dst`. Na primer, za uneti string "dan", string `dst` treba da bude "dandandan". Pretpostaviti da string `src` nije duzi od 30 karaktera.

**Zadatak 3.3.4** Napisati funkciju `int učitaj_liniju(char s[], int n)` koja učitava liniju maksimalne dužine `n` u string `s` i vraća dužinu učitane linije. Linija može da sadrži blanko znakove ali ne može da sadrži `\n` ili EOF.

Napisati potom glavni program koji učitava linije do EOF i ispisuje najdužu liniju i njenu dužinu. Ukoliko ima više linija maksimalne dužine, ispisati prvu. Možemo pretpostaviti da svaka linija sadrži najviše 80 karaktera, zajedno sa `\n`.

**Zadatak 3.3.5** Napisati program koji pretvara nisku u ceo broj. Npr. za ulaz "-1238" se generise rezultat -1238 Pogledati funkcije `atoi` i `atof` koje postoje u biblioteci `stdlib.h`

**Zadatak 3.3.6** Napisati program koji pretvara zadatu broj u nisku. Npr. za broj -453 treba generisati nisku "-453"

**Zadatak 3.3.7** Napisati program koji učitava dva stringa i ispituje najpre da li su jednaki. Ako jesu, program treba da izda odgovarajuću poruku, a ako nisu, treba da ispita da li je drugi podstring prvog. Ukoliko jeste, program treba da ispise počev od kog indeksa prvog stringa počinje drugi string. U suprotnom, ispisati odgovarajuću poruku. Možemo pretpostaviti da stringovi ne sadrže više od 20 karaktera.

**Zadatak 3.3.8** Napisati program koji za uneti string `s` i karakter `c` utvrđuje da li se `c` pojavljuje u stringu `s` i ukoliko se pojavljuje, ispisuje indeks prvog pojavljivanja `a` u suprotnom ispisuje odgovarajuću poruku. Možemo pretpostaviti da string ima najviše 20 karaktera.

### **Zadatak 3.3.9**

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

### 3 Predstavljanje podataka

---

#### Primer 1

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

#### Primer 2

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

#### Primer 3

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

**Zadatak 3.3.10** Napisati funkciju *void kopiraj\_n(char t[], char s[], int n)* koja kopira najviše  $n$  karaktera niske  $s$  u nisku  $t$ . Napisati i program koji učitava nisku maksimalne dužine 20 karaktera i jedan ceo broj i testira rad napisane funkcije.

#### Primer 1

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

#### Primer 2

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

#### Primer 3

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

**Zadatak 3.3.11** Napisati funkciju *void dupliranje(char t[], char s[])* koja na osnovu niske  $s$  formira nisku  $t$  tako što duplira svaki karakter niske  $s$ . Napisati i program koji učitava nisku maksimalne dužine 20 karaktera i testira rad napisane funkcije.

#### Primer 1

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

#### Primer 2

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

#### Primer 3

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



**Zadatak 3.3.12** Napisati funkciju *int heksa\_broj(char s[])* koja proverava da li je niskom *s* zadat korektan heksadekadni broj. Heksadekadni broj je korektno zadat ako počinje prefiksom *0x* ili *0X* i ako sadrži samo cifre i mala ili velika slova *A, B, C, D, E* i *F*. Funkcija treba da vrati vrednost 1 ako je niska korektan heksadekadni broj, odnosno 0 ako nije. Napisati i program koji učitava nisku maksimalne dužine 7 karaktera i ispisuje rezultat rada funkcije.

*Primer 1*

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

*Primer 2*

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

*Primer 3*

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

**Zadatak 3.3.13** Napisati funkciju *int heksa\_broj(char s[])* koja izračunava dekadnu vrednost heksadekadnog broja zadatog niskom *s*. Napisati i program koji učitava nisku maksimalne dužine 7 karaktera i ispisuje rezultat rada funkcije. Pretpostaviti da je uneta niska korektan heksadekadni broj.

*Primer 1*

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

*Primer 2*

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite nisku: 0Xff2
|| 4082
```

*Primer 3*

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

**Zadatak 3.3.14** Napisati funkciju *int podniska(char s[], char t[])* koja proverava da li je niska *t* podniska niske *s*. Napisati i program koji učitava dve niske maksimalne dužine 10 karaktera i testira rad napisane funkcije.

*Primer 1*

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

*Primer 2*

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

### 3 Predstavljanje podataka

---

#### Primer 3

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

**Zadatak 3.3.15** Napisati funkciju *void modifikacija(char \* s)* koja modifikuje nisku *s* tako što svaki drugi karakter zameni zvezdicom. Pretpostaviti da niska *s* neće biti duža od 20 karaktera. Napisati i program koji testira rad napisane funkcije.

#### Primer 1

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

#### Primer 2

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

#### Primer 3

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

**Zadatak 3.3.16** Napisati funkciju *int strspn\_klon(char \* t, char \* s)* koja izračunava dužinu prefiksa niske *t* sastavljenog od karaktera niske *s*. Napisati zatim i program koji učitava dve niske maksimalne dužine 20 karaktera i ispisuje rezultat poziva napisane funkcije.

#### Primer 1

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

#### Primer 2

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesite nisku t: aaiioo124  
|| Unesite nisku s: aeiou  
|| 6
```

#### Primer 3

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

**Zadatak 3.3.17** Napisati implementaciju funkcije *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*. Učitati potom jednu nisku maksimalne dužine 20 karaktera i jedan dodatni karakter i testirati rad napisane

funkcije.

#### Primer 1

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

#### Primer 2

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

### Zadatak 3.3.18

- Napisati funkciju

```
int prepis(char a[][21], int na, char b[][21])
```

koja iz niza reči **a** dužine **na** prepisuje u niz **b** reči koje su zapisane samo malim ili samo velikim slovima. Informaciju o dužini niza **b** (broj reči koje zadovoljavaju prethodni uslov) vratiti kao povratnu vrednost funkcije.

- Napisati program koji sa standardnog ulaza učitava prvo broj reči (strogo veći od nule, manji od 50), a zatim i same reči razdvojene blanko znakom (smatrati da reči koje se unose sa ulaza neće biti duže od 20 karaktera - ovaj uslov ne proveravati). Za slučaj kada je broj reči izvan traženog opsega ispisati -1 i prekinuti izvršavanje programa. Korišćenjem prethodno definisane funkcije **prepis**, odrediti sve reči koje su zapisane samo malim ili samo velikim slovima. Rezultat ispisati na standardni izlaz. Napomena: Ukoliko se pri rešavanju zadatka ne bude koristila funkcija **prepis**, zadatak neće biti pregledan i nosiće nula poena.

#### Primer 1

```
INTERAKCIJA SA PROGRAMOM:
3 abc ABC aBc
abc ABC
```

#### Primer 2

```
INTERAKCIJA SA PROGRAMOM:
2 mmB RGa
```

#### Primer 3

```
INTERAKCIJA SA PROGRAMOM:
-3
-1
```

#### Primer 4

```
INTERAKCIJA SA PROGRAMOM:
4 2abc AVF$ abc AV4
abc
```

**Zadatak 3.3.19** Napisati funkciju `void min_razlika(char s[], char s1[], char s2[])` koja u datotoj nisci s pronalazi dve reči koje imaju minimalnu razliku između svojih

### 3 Predstavljanje podataka

---

samoglasnika. ( Reč je niz karaktera između dve praznine; razmak između samoglasnika reči *dan*as i *jut*ro je 2, a razmak između *sut*rk i *mno*zenje je 5). Testirati pozivom u main-u. Maksimalna dužina niske je 20 karaktera.

**Zadatak 3.3.20** Napisati funkciju `int pp(char s[], char t[])` koja određuje poziciju poslednjeg karaktera niske *s* sadržanog u okviru niske *t*, zane-marujući pri tom razliku između velikih i malih slova, ili -1 ako takvog karaktera nema. Testirati pozivom u main-u. Maksimalna dužina niske je 20 karaktera.

#### Primer 1

```
|| INTERAKCIJA SA PROGRAMOM:
|| a4BA3Bc A3b
|| 5
```

**Zadatak 3.3.21** Napisati funkciju `int f1(char s[])` koja prihvata tu nisku i proverava da li niska sadrži veliko slovo. Funkcija vraća 1 ako sadrži veliko slovo, inače 0. Testirati pozivom u main-u. Maksimalna dužina niske je 20 karaktera.

**Zadatak 3.3.22** Napisati funkciju `void ukloniSlova(char s[])` koja iz niske *s* uklanja sva mala i velika slova. Testirati pozivom u main-u. Maksimalna dužina niske je 20 karaktera.

**Zadatak 3.3.23** Napisati funkciju `unsigned btoi(char* s, unsigned char b)` koja određuje vrednost zapisa datog neoznačenog broja *s* u datoj osnovi *b*. Napisati funkciju `void itob(unsigned n, unsigned char b, char* s)` koja datu vrednost *n* zapisuje u datoj osnovi *b* i smešta rezultat u nisku *s*. Napisati zatim program koji čita liniju po liniju sa standardnog ulaza i obrađuje ih sve dok ne naiđe na praznu liniju. Svaka linija sadrži jedan dekadni, oktalni ili heksadekadni broj (zapisan kako se zapisuju konstante u programskom jeziku C). Program za svaki uneti broj ispisuje njegov binarni zapis. Pretpostaviti da će svi uneti brojevi biti u opsegu tipa `unsigned`.

#### Primer 1

```
|| INTERAKCIJA SA PROGRAMOM:
|| 0x49 0x1ABC
|| 1001001 1101010111100
```

#### Primer 2

```
|| INTERAKCIJA SA PROGRAMOM:
|| 012 435 0x64FE
|| 1010 110110011 110010011111110
```

#### Primer 3

```
|| INTERAKCIJA SA PROGRAMOM:
|| 123 0777
|| 1111011 111111111
```

#### Primer 4

```
|| INTERAKCIJA SA PROGRAMOM:
|| 981
|| 1111010101
```

**Zadatak 3.3.24** Implementirati funkciju `int str_str(char s[], char t[])` koja proverava da li niska `s` sadrzi nisku `t`. Zatim napisati program koji sa standardnog ulaza učitava pet redova (svaki red ima najviše 100 karaktera) i koji ispisuje sve redne brojeve linija koje sadrže nisku `program` (linije se numerišu od broja 1). Ukoliko ne postoji red sa niskom `program` ispisati -1.

*Primer 1*

```
INTERAKCIJA SA PROGRAMOM:
novi red*program
c prog. jezik
c? programskih jezik
Programski odbor
<b>program</b>
1 3 5
```

**Zadatak 3.3.25** Napisati funkciju `void sifrat(char* rec, char* kljuc)` koja šifruje `rec` na sledeći način: za svako slovo reči `rec` i odgovarajuće slovo ključa određuje koliki je (alfabetски) razmak između njih i označimo taj broj sa `k`. Potom to slovo `reci` zamenjuje `k`-tim slovom alfabeta. Podrazumeva se da je ključ duži od `reci`.

*Primer 1*

```
INTERAKCIJA SA PROGRAMOM:
bac
dfge
bed
```

**Zadatak 3.3.26** Napisati funkciju `void obrni(char rec[], int k)` koja rotira reč za `k` mesta ulevo.

*Primer 1*

```
INTERAKCIJA SA PROGRAMOM:
sveska
2
eskasv
```

**Zadatak 3.3.27** Napisati sledeće funkcije:

```
int poredjenje(char* s1, char* s2);
// vraca 1 ako su s1 i s2 iste niske, 0 u suprotnom
```

```
void uVelikaSlova(char* s);
// pretvara sva slova niske s u velika, ostale znakove ne menja
```

Napisati program koji sa standardnog ulaza učitava dve reči (dužine najviše 20 znakova) i, koristeći ove dve funkcije, ispisuje da li su one jednake ako se sva slova pretvore u velika slova.

#### Primer 1

INTERAKCIJA SA PROGRAMOM:

```
isPit2010  
IsPit2010  
jesu jednake
```

**Zadatak 3.3.28** Napisati program kojim se sadržaj unetog stirnga šifrira tako što se svako slovo zamenjuje sledećim ASCII slovom, a znakovi 'z' i 'Z' zamenjuju redom sa 'a' i 'A'. Uneta reč nije duža od 20 karaktera.

**Zadatak 3.3.29** Napisati funkciju `void sifruj(char rec[], char sifra[])` koja na osnovu date reči formira šifru koja se dobija tako što se svako slovo u reči zameni sa naredna tri slova koja su mu susedna u abecedi. Na primer, reč "tamo" treba da bude zamenjena sa "uvwbcnoppqr" a reč "zec" sa "abcfghdef". Napisati program koji šifrue unetu reč sa standardnog ulaza i štampa dobijeni rezultat na standardni izlaz. Za reč pretpostaviti da nije duža od 20 karaktera. Unos reči ostvariti koristeći specifikator "

**Zadatak 3.3.30** Sa ulaza se unosi reč koja nije duža od 20 znakova. Napisati program koji formira i štampa rezultujuću reč koja se dobija tako što se uneta reč kopira 4 puta pri čemu se između svakog kopiranja umeće crtica. Na primer ako je uneta reč **ana**, formirana reč treba da bude **ana-ana-ana-ana**. Zadatak uraditi:

- (a) pisanjem odgovarajuće funkcije koja vrši nadovezivanje reči,
- (b) koristeći postojeću funkciju iz biblioteke `string.h` (`strcat`).

Napomena: voditi računa da se za rezultujuću reč odvoji odgovarajuća količina memorije.

**Zadatak 3.3.31** Sa ulaza se unosi reč koja nije duža od 20 znakova. Napisati funkciju koja svako pojavljivanje znaka koji se zadaje kao prvi argument funkcije udvaja a svako po- javljivanje znaka koji se zadaje kao drugi argument funkcije izbacuje. Napisati program koji poziva ovu funkciju za reč unetu sa standardnog ulaza i za znakove koji se takođe zadaju sa standardnog ulaza. Na primer, ako se unese reč **ana** i znakovi **a** i **n**, tada funkcija treba da izmeni reč tako da ona postane **aaaa**, ako se unese reč **abrakadabra** i znakovi **a** i **b**, tada funkcija treba reč da izmeni tako da ona postane **aaraakaadkkraa**.

Napomena: voditi računa da novonastala izmenjena reč može imati veći broj karaktera i u skladu sa tim rezervisati odgovarajuću količinu memorije. Dopušteno je koristiti pomoćan niz.

**Zadatak 3.3.32** Napisati funkciju `void ukloni(char *s);` koja iz niske uklanja sva slova iza kojih neposredno sledi slovo koje je u abecedi nakon njih (veličina slova se zanemaruje). Testirati funkciju u programu koji učitava liniju teksta (najviše 100 karaktera).

*Primer 1*

```
|| INTERAKCIJA SA PROGRAMOM:
||   zdRaVo svIma
||   zRVo vma
```

*Primer 2*

```
|| INTERAKCIJA SA PROGRAMOM:
||   12345AbcD
||   12345D
```

*Primer 3*

```
|| INTERAKCIJA SA PROGRAMOM:
||   JeD1aN D52Va.
||   JeD1N D52Va.
```

*Primer 4*

```
|| INTERAKCIJA SA PROGRAMOM:
||   abcd efg
||   d g
```

**Zadatak 3.3.33**

- Napisati C funkciju `int procitaj_recenicu(char *s, int max_len)`, koja sa standardnog ulaza čita rečenicu i smešta je u nisku `s`. Čitanje rečenice se zaustavlja ako se pročita simbol `.` ili je već učitano `max_len-1` karaktera. Funkcija treba da vrati broj pročitanih karaktera.
- Napisati C funkciju `void prebroj(char *s, int *broj_malih, int *broj_velikih)`, koja za zadatu nisku `s` računa broj malih i velikih slova koji se u njoj pojavljuju.
- Napisati glavni program koji sa standardnog ulaza čita rečenice i na standardni izlaz ispisuje onu kod koje je razlika broja malih i velikih slova najveća.

**Zadatak 3.3.34**

- Uvesti tip podataka `Sifra` kojim se opisuje način šifrovanja alfanumeričkih karaktera. Svaka šifra se opisuje celobrojnom vrednošću `b` koja određuje broj pozicija pomeranja, kao i karakterom `'L'` ili `'D'` koji određuje smer pomeranja (levo ili desno).
- Napisati funkciju `void sifruj(char rec[], Sifra s)` koja transformiše zadatu reč `rec` po šifri `s`. Reč se šifruje tako što se svako slovo zamenjuje slovom za `b` mesta levo ili desno od njega u abecedi, i to ciklično, a isto tako i za cifre.  
Npr: za `b=2`, i `smer='D'` : a se menja sa c, b sa d,..., x sa z, y sa a, z sa b, 1 sa 3, .. 8 sa 0, 9 sa 1

- c) Sa standardnog ulaza se zadaje način šifrovanja i to u obliku 2 D 5 L (šifra može biti i duža). Potom se učitava  $n$  i  $n$  reči sa standardnog ulaza (maksimalna dužina reči je 20 karaktera). Ispisati reči na standardni izlaz nakon primenjenih svih zadatih načina šifrovanja.

**Zadatak 3.3.35** Implementirati funkciju `int strspn(char* s, char* t)` koja izračunava dužinu početnog dela niske `s` sastavljenog isključivo od karaktera sadržanih u niski `t`.

Napisati i program koji sa standardnog ulaza učitava dve niske (dužine najviše 100 karaktera, svaku u zasebnom redu) i ispisuje rezultat poziva funkcije `strspn` na standardni izlaz.

Na primer, za učitane podatke "734a.bf62", "0123456789") program ispisuje vrednost 3.

## 3.4 Strukture

**Zadatak 3.4.1** Data je struktura koja opisuje koordinate tacke u ravni:

```
typedef struct point
{
    float x;
    float y;
} POINT;
```

U glavnom programu date su dve tacke: tacka A sa fiksiranim koordinatama (1,2) i tacka B cije koordinate zadaje korisnik. Napisati funkcije:

- za racunanje rastojanja izmedju dve date tacke
- za odredjivanje tacke koja se nalazi na sredini duzi odredjene dvema datim tackama.

Testirati napisane funkcije u glavnom programu.

**Zadatak 3.4.2** Data je struktura

```
typedef struct Student
{
    char ime[MAX];
    char prezime[MAX];
    char smer;
    float prosek;
} STUDENT;
```



- (a) Napisati funkciju koja ucitava sa standardnog ulaza podatke o studentu. Mozemo pretpostaviti da ime i prezime studenta ne sadrže više od 30 karaktera.
- (b) Napisati funkciju koja ispisuje podatke o studentu na standardni izlaz.
- (c) Ucitati niz od  $n$  studenata i :
  - (a) ispisati imena i prezimena onih koji su na smeru R
  - (b) ispisati podatke za studenta sa najvećim prosekom; ako ima više takvih studenata, ispisati
    - (a) sve njih
    - (b) prvog
    - (c) poslednjeg

#### Zadatak 3.4.3

Napisati program koji ucitava reci sa standardnog ulaza dok korisnik ne zada EOF i ispisuje ih na standardni izlaz svaku u posebnom redu, poravnatu udesno u odnosu na poslednji karakter najduze reci. Koristiti

```
strukturu typedef struct rec
{
    char s[21];
    int duzina;
}REC;
```

Na primer, ako su unesene sledece reci:

```
Danas imamo ispit iz programiranja1.
Nadam se da nece biti tesko!
```

onda ispis izgleda ovako:

```
        Danas
        imamo
        ispit
        iz
programiranja1.
        Nadam
        se
        da
        nece
        biti
        tesko!
```

### 3 Predstavljanje podataka

---

Program realizovati kroz sledece funkcije:

- (a) Funkciju za učitavanje jedne reci u strukturu REC.
- (b) Funkciju za učitavanje niza struktura koja vraća dimenziju niza
- (c) Funkciju koja određuje maksimalnu dužinu reci u datom nizu
- (d) Funkciju koja ispisuje reci u traženom formatu

Mozemo pretpostaviti da nijedna rec ne sadrži više od 30 karaktera i da neće biti uneto više od 1000 reci.

**Zadatak 3.4.4** Napisati program koji izracunava prosečnu cenu jedne potrosacke korpe. Potrosacka korpa se sastoji od broja kupljenih artikala i niza kupljenih artikala. Svaki artikal određen je svojim nazivom, kolicinom i cenom. Program treba da ucita broj potrosaca  $n$  (najviše 100), zatim podatke za  $n$  potrosackih korpi i da na osnovu učitanih podataka izracuna prosečnu cenu potrosacke korpe. Učitavanje se vrši sa standardnog ulaza pri čemu se prvo zadaje broj artikala, a zatim za svaki artikal naziv, kolicina i cena. Mozemo pretpostaviti da nijedan potrosac neće kupiti više od 20 artikala, kao i da naziv svakog artikla sadrži maksimalno 30 karaktera.

**Zadatak 3.4.5** Struktura IZRAZ opisuje numericki izraz nad celim brojevima koji se sastoji od dva celobrojna operanda, numericke operacije nad celim brojevima i vrednosti izraza:

```
typedef struct izraz { char o; int x; int y; IZRAZ;
```

- (a) Napisati funkciju koja ispituje da li je dati izraz korektno zadat i vraća 1 ako jeste a 0 u suprotnom. Podrazumevamo da je izraz korektno zadat ako operacija odgovara +, -, \* ili / i u slučaju deljenja drugi operand je različit od 0.
- (b) Napisati funkciju koja za dati izraz određuje vrednost izraza.
- (c) Napisati funkciju koja ucitava dati izraz. Funkcija treba da ucita sa standardnog ulaza operaciju i dva operanda u polja o, x i y strukture IZRAZ. Funkcija vraća 1 ako je učitavanje bilo uspesno, tj. ako je izraz bio korektno zadat ili 0 u suprotnom.
- (d) Napisati funkciju koja stampa dati izraz infiksno, u obliku  $x \ o \ y = vr$ . Na primer, za izraz  $+ \ 4 \ 17$  ispis treba da bude  $4+17=21$

Napisati glavni program koji ucitava prirodan broj  $n < 1000$  a zatim  $n$  izraza u notaciji

+ 4 17

- 8 -16

Program treba da ispise maksimalnu vrednost medju unetim izrazima i da ispise one izraze cija je vrednost manja od polovine maksimalne vrednosti.

**Zadatak 3.4.6** Definisati strukturu kojom se predstavlja kompleksan broj. Napisati funkcije koje izracunavaju zbir, razliku, proizvod i količnik dva kompleksna broja, a zati i program koji učitava dva kompleksna broja i ispisuje vrednost zbira, razlike, proizvoda i količnika.

#### Primer 1

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

**Zadatak 3.4.7** Definisati strukturu *Lopta* sa poljima *poluprecnik* (ceo broj u centimetrima) i *boja* (enumeracioni tip koji uključuje plavu, žutu, crvenu i zelenu boju). Zatim učitati informacije o  $n$  lopti ( $0 < n < 50$ ) i ispisati ukupnu zapreminu, kao i broj crvenih lopti. *Napomena: probati sa testiranjem zadataka pomoću preusmeravanja.*

#### Primer 1

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

**Zadatak 3.4.8** Zimi su prehlade česte i treba unositi više vitamina C. Struktura *Vocka* sadrži ime vočke (nisku maksimalne dužine 20 karaktera) i količinu vitamina C u miligramima (realan broj). Napisati program koji sa standardnog ulaza učitava podatke o vočkama sve do unosa reči KRAJ i ispisuje ime vočke sa najviše vitamina C. Pretpostaviti da broj vočki neće biti veći od 50. *Napomena: probati sa testiranjem zadataka pomoću preusmeravanja.*

### 3 Predstavljanje podataka

---

#### Primer 1

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

**Zadatak 3.4.9** Deda Mraz planira kupovinu poklona za studente koji su vredno učili C u toku godine. Na njegovoj listi se nalazi ime i prezime studenta (niske dužina do 50 karaktera) i njegova želja (niska maksimalne dužine 100 karaktera). Napisati program koji će služiti Deda Mrazu kao podsetnik: na osnovu liste koju je napravio, Deda Mraz može da unese ime i prezime studenta i da proveri njegovu želju. Ako ima više studenata sa istim imenom i prezimenom ispisati sve želje. *Napomena: probati sa testiranjem zadataka pomoću preusmeravanja.*

#### Primer 1

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

**Zadatak 3.4.10** Definisati strukturu *Grad* u kojoj se nalazi ime grada (niske dužine 20 karaktera) i prosečna temperatura u toku decembra (realan broj). Napisati program koji učitava imena  $n$  ( $0 < n < 50$ ) gradova i njihove prosečne temperature, a zatim ispisuje one gradove koji imaju idealnu temperaturu za klijanje: od 3 do 8 stepeni. *Napomena: probati sa testiranjem zadataka pomoću preusmeravanja.*

*Primer 1*

```

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

```

*Primer 2*

```

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

```

**Zadatak 3.4.11** Definisati strukturu *ParReci* koja sadrži reč na srpskom jeziku i odgovarajući prevod na engleski jezik. Zatim sa standardnog ulaza sve do kraja ulaza učitavati parove reči i, posebno, za rečenicu koja se zadaje sa ulaza ispisati prevod - ako je reč u rečenici nepoznata umesto nje ispisati odgovarajući broj zvezdica. Reči neće biti duže od 50 karaktera, ukupan broj parova reči neće biti veći od 100, a ukupna dužina rečenice neće biti veća od 100 karaktera. *Napomena: probati sa testiranjem zadataka pomoću preusmeravanja.*

*Primer 1*

```

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

```

**Zadatak 3.4.12** Napisati funkcije koje izračunavaju zbir, razliku i proizvod dva razlomka, *razlomak zbir(razlomak a, razlomak b)* itd. Unosi se broj *n* a potom i *n* razlomaka sa standardnog ulaza (najviše 100). Ispisati njihov zbir, razliku i proizvod na standardni izlaz.

**Zadatak 3.4.13** Napraviti strukturu *VOCE* koja sadrži ime (ne duže od 20 karaktera) i cenu (tipa *float*). Sa standardnog ulaza unosi se broj voćki (ne vići od 200), a potom uneti niz voća i pozvati funkciju koja izračunava prosečnu cenu voća. Potom ispisati imena onih voćki čija je cena veća od prosečne.

### 3 Predstavljanje podataka

---

**Zadatak 3.4.14** Sa standardnog ulaza učitava se  $n$  ( $0 < n \leq 200$ ), a potom i spisak (dužine  $n$ ) engleskih reči i njihov prevod na srpski jezik. Potom se učitava jedna reč sa standardnog ulaza. Na standardni izlaz ispisati odgovarajući prevod date reči ili podatak o tome da se reč ne nalazi na spisku.

```
apple jabuka
pineapple ananas
orange narandza
pear kruska
grape grozdje
```

i reč *orange* program treba da ispiše *narandza* a za reč *cherry* program treba da ispiše poruku *Rec se ne nalazi u recniku*. U programu se mogu koristiti funkcije iz zaglavlja *string.h*.

**Zadatak 3.4.15** Napisati program koji sa standardnog ulaza čitava najpre broj artikala (ceo broj manji od 20) a zatim podatke o artiklima. Artikli su voćke koje imaju po dva podatka: naziv voćke i cenu (naziv voćke je karakterska niska dužine do 20 karaktera). Program potom traži od korisnika da unese neku cenu i štampa na standardni izlaz sve voćke koje imaju zadatu cenu.

Primer rada programa:

```
4
jabuka 30
kruska 40
ananas 60
limun 40
```

```
Unesite cenu: 40
Voce te cene je: kruska limun
```

**Zadatak 3.4.16** Definisati strukturu koja opisuje dete atributima ime deteta (ne vece od 20 karaktera) , pol deteta (m ili z) i ocena. Ocenu je svako dete dalo radu obdaništa. Maksimalan broj dece je 100. Napisati program koji:

a) Sa standardnog ulaza se unosi  $n$ , a potom podaci o  $n$  dece. Koristiti strukturu:

```
typedef struct
{
    char ime[20];
    char pol;
```

```
    int ocena;  
} DETE;
```

- b) ispisati na standardni izlaz statistiku: koliko ima dečaka, a koliko devojčica i prosečnu ocenu. Potom ispisuje imena dece brojnijeg pola.

### Zadatak 3.4.17

- Definisati tip podataka **TACKA** pogodan za predstavljanje tačke Dekartovske ravni (čije su  $x$  i  $y$  koordinate podaci tipa **double**).
- Definisati funkciju **double rastojanje(TACKA a, TACKA b)** koja izračunava rastojanje između dve tačke.
- Definisati funkciju **unsigned ucitaj\_poligon(TACKA\* tacke, unsigned n)** koja učitava **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.
- Definisati funkciju **double obim(TACKA\* poligon, unsigned n)** koja izračunava obim poligona sa **n** tačaka u zadatom nizu (napomena: ne zaboraviti stranicu koja spaja poslednje i prvo teme).
- Definisati funkciju **double maksimalna\_stranica(TACKA\* poligon, unsigned n)** koja izračunava dužinu najduže stranice poligona sa **n** tačaka u zadatom nizu (napomena: ne zaboraviti stranicu koja spaja poslednje i prvo teme).
- Definisati funkciju **main** u kojoj se sa standardnog ulaza učitava celobrojna nenegativna vrednost **N** ( $0 < N \leq 100$ ).

Inače, poziva se funkcija **ucitaj\_poligon**. Ukoliko je uspešno učitano **m** tačka (**N** ne mora da bude jednako **m**), onda se poziva funkcija **obim** za **m** učitanih tačaka i ispisuje njen rezultat na standardni izlaz (ukoliko ova funkcija nije implementirana — ispisati na standardni izlaz simbol **?**). Posle toga se poziva funkcija **maksimalna\_stranica** za **m** učitanih tačaka i ispisuje njen rezultat na standardni izlaz (ukoliko ova funkcija nije implementirana — ispisati na standardni izlaz simbol **?**).





## 4

# Ulaz i izlaz programa

## 4.1 Datoteke

**Zadatak 4.1.1** Napisati program koji prepisuje sadržaj datoteke `ulaz.txt` u datoteku `izlaz.txt` karakter po karakter.

**Zadatak 4.1.2** Napisati program koji u datoteci cije se ime navodi kao prvi argument komandne linije određuje liniju maksimalne dužine i ispisuje je na standardni izlaz. Ukoliko ima više takvih linija, ispisati onu koja je leksikografski prva. Možemo pretpostaviti da datoteka ne sadrži linije duže od 80 karaktera.

**Zadatak 4.1.3** U datoteci cije se ime zadaje kao prvi argument komandne linije nalazi se prirodan broj  $n$  a zatim  $i$   $n$  celih brojeva. Napisati program koji prebrojava koliko  $k$ -tocifrenih brojeva postoji u datoteci, pri čemu se prirodan broj  $k$  zadaje kao drugi argument komandne linije.

**Zadatak 4.1.4** U datoteci cije se ime navodi kao prvi argument komandne linije navedena je rec  $r$  i niz linija. Napisati program koji u datoteku cije se ime navodi kao drugi argument komandne linije upisuje sve linije u kojima se rec  $r$  pojavljuje bar  $n$  puta, gde je  $n$  prirodan broj koji se unosi sa standardnog ulaza. Ispis treba da bude u formatu `broj_pojavljivanja : linija`.

**Zadatak 4.1.5** Program se pokreće tako što se navedu nazivi dve datoteke (ulazna i izlazna) i opcije. U datoteci cije se ime navodi kao prvi argument komandne linije nalaze se podaci o razlomcima: u prvom redu se nalazi broj razlomaka, a u svakom sledećem redu brojilac i imenilac jednog razlomka. Potrebno je kreirati strukturu koja opisuje razlomak i učitati niz razlomaka iz datoteke, a potom:

## 4 Ulaz i izlaz programa

---

a) ukoliko je navedena opcija x, upisati u datoteku cije je ime drugi argument komandne linije recipročni razlomak za svaki razlomak iz niza (npr. za  $2/3$  treba upisati  $3/2$ )

b) ukoliko je navedena opcija y, upisati u datoteku cije je ime drugi argument komandne linije realnu vrednost recipročnog razlomka svakog razlomka iz niza (npr. za  $2/3$  treba upisati 1.5)

Mozemo pretpostaviti da se u datoteci sa podacima o razlomcima nalazi najviše 100 razlomaka.

**Zadatak 4.1.6** Za svaki automobil poznati su marka, model i cena. Iz datoteke cije se ime zadaje sa standardnog ulaza učitava se broj automobila a potom i podaci za svaki automobil. Program treba da:

a) izracuna prosečnu cenu po marki kola

b) za maksimalnu cenu koju je kupac spreman da plati, a koja se zadaje kao argument komandne linije, da ispise automobile u tom cenovnom rangi zajedno sa prosečnom cenom odgovarajuće marke

Mozemo pretpostaviti da se model i marka sastoje od jedne reci i da svaka od njih sadrži najviše 30 karaktera kao i da se u datoteci nalaze podaci za najviše 100 automobila.

**Zadatak 4.1.7** Napisati program koji prebrojava mala slova u datoteci *test.txt*.

### Primer 1

```
TEST.TXT
Abcd EFGH+ijKLMN

IZLAZ:
Broj malih slova je: 5
```

### Primer 2

```
TEST.TXT
PrograMiranje

IZLAZ:
Broj malih slova je: 11
```

**Zadatak 4.1.8** Napisati program koji prepisuje svaki treći karakter datoteke *ulaz.txt* u datoteku *izlaz.txt*.

### Primer 1

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

**Zadatak 4.1.9** Kao argumenti komandne linije se zadaju ime datoteke i ceo broj  $k$ . Napisati program koji na standardni izlaz ispisuje sve linije zadate datoteke čija je dužina veća od  $k$ . Može se pretpostaviti da dužina linije neće biti veća od 80 karaktera.

*Primer 1*

```

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

IZLAZ:
Teme koje su obradjivane:
Funkcije
Strukture

```

*Primer 2*

```

POKRETANJE: ./a.out test.txt

IZLAZ:
Greska: Pogresan broj argumenata!

```

**Zadatak 4.1.10** Napisati program koji prebrojava koliko se linija datoteke *ulaz.txt* završava niskom *s* koja se učitava sa standardnog ulaza. Može se pretpostaviti da dužina linije neće biti veća od 80 karaktera, kao i da dužina niske *s* neće biti veća od 20 karaktera.

*Primer 1*

```

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

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

```

*Primer 2*

```

ULAZ.TXT
abcde abcde
abcde
abcde abcde AB

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

```

**Zadatak 4.1.11** Napisati program koji pronalazi maksimum brojeva zapsanih u datoteci *brojevi.txt*.

*Primer 1*

```

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

IZLAZ:
Najveci broj je: 54.96

```

**Zadatak 4.1.12** U datoteci *studenti.txt* se nalaze informacije o studentima: prvo broj studenata, a zatim u pojedinačnim linijama korisničko ime i pet poslednjih ocena koje je student dobio. Napisati program koji pronalazi studenta koji je ostvario najbolji uspeh i ispisuje njegove podatke. Pretpostaviti da broj

## 4 Ulaz i izlaz programa

---

studenata neće biti veći od 100.

### 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:
korisnicko ime: mr15007, prosek ocena: 10.00
```

**Zadatak 4.1.13** U datoteci *tacke.txt* se nalazi prvo broj tačaka, a zatim u pojedinačnim linijama  $x$  i  $y$  koordinate tačke. Napisati program koji u datoteku *rastojanja.txt* upisuje rastojanje svake od pročitanih tačaka od koordinatnog početka, a na standardni izlaz koordinate tačke koja je najudaljenija. Koristiti strukturu *Tacka* sa poljima  $x$  i  $y$ , kao i funkciju kojom se računa rastojanje. Pretpostaviti da broj tačaka u datoteci neće biti veći od 50.

### Primer 1

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

RASTOJANJA.TXT
11.18
5.29
11.31
4.00

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

### Primer 1

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

IZLAZ:
Greska: Nedozvoljen broj tacaka!
```

**Zadatak 4.1.14** Napisati program koji za reč  $s$  maksimalne dužine 20 karaktera koja se zadaje sa standardnog ulaza u datoteku *rotacije.txt* upisuje sve rotacije reči  $s$ .

*Primer 1*

```

INTERAKCIJA SA PROGRAMOM:
  Unesite rec:  abcde

ROTACIJE.TXT
  abcde
  bcdea
  cdeab
  deabc
  eabcd

```

**Zadatak 4.1.15** Napisati program koji linije koji se učitavaju sa standardnog ulaza sve do kraja ulaza prepisuje u datoteku *izlaz.txt* i to, ako je prilikom pokretanja zadata opcija *-v* ili *-V* samo one linije koje počinju velikim slovom, ako je zadata opcija *-m* ili *-M* samo one linije koje počinju malim slovom, a ako je opcija izostavljena sve linije. Pretpostaviti da linije neće biti duže od 80 karaktera.

*Primer 1*

```

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

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

```

*Primer 2*

```

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

IZLAZ.TXT
  Volim programiranje!
  Kada porastem bicu programer!

```

*Primer 3*

```

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

```

**Zadatak 4.1.16** Sa standardnog ulaza učitavaju se imena dve tekstualne datoteke i jedan karakter. Napisati program koji prepisuje datoteku čije se ime navodi kao prvo u datoteku čije ime se navodi kao drugo. Ukoliko je učitani karakter u program prilikom prepisivanja treba da zamenjuje sva mala slova velikim, a ukoliko je učitani karakter 1 sva velika slova se zamenjuju malim. U slučaju greske ispisati -1. Greška može biti neuspešno otvaranje datoteke ili pogrešno zadat karakter. Maksimalna dužina naziva datoteka je 20 karaktera.

## 4 Ulaz i izlaz programa

---

### Primer 1

```
INTERAKCIJA SA PROGRAMOM:
  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:
  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:
  primer.c prazna.txt V
PRIMER.C
  #include <stdio.h>
  int main()
  {
  }
PRAZNA.TXT

IZLAZ:
  -1
```

**Zadatak 4.1.17** Sastaviti program koji sa standardnog ulaza prima ime datoteke koju treba otvoriti. Ispisati (na standardnom izlazu) koja cifra (meu svim ciframa koje se pojavljuju u datoteci) ima najveći broj pojavljivanja. U slučaju greške pri otvaranju datoteke ispisati -1. Ukoliko nema cifara u datoteci ispisati -1. Maksimalna dužina naziva datoteka je 20 karaktera.

### Primer 1

```
INTERAKCIJA SA PROGRAMOM:
  ulaz.txt
ULAZ.TXT
  danas je lep dan
  i Ja zelim
  da postanem programer
IZLAZ:
  -1
```

### Primer 2

```
INTERAKCIJA SA PROGRAMOM:
  prva.dat druga.dat l
PRVA.DAT
  Cena soka je 30
  Cena vina je 150
  Cena limunade je 200
  Cena sendvica je 120
IZLAZ:
  0
```

*Primer 3*

```

INTERAKCIJA SA PROGRAMOM:
  primer.c
PRIMER.C
  #include <stdio.h>
  int main()
  {
  }
PRAZNA.TXT

IZLAZ:
  -1

```

**Zadatak 4.1.18** Prvi red datoteke `matrice.txt` sadrži 2 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).

U slučaju greške prilikom otvaranja datoteke ispisati -1. Pretpostaviti da je sadržaj datoteke ispravan.

*Primer 1*

```

MATRICE.TXT
  1 2 3 4
  7 2 15 -3
  -1 3 1 3
IZLAZ:
  (1, 0, 7)
  (1, 2, 15)

```

**Zadatak 4.1.19** Napisati program koji za dve datoteke čija su imena data kao prvi i drugi na standardnom ulazu, radi sledeće: za cifru u prvoj datoteci, u drugu datoteku se upisuje 0, za slovo se upisuje 1, a za sve ostale karaktere se upisuje 2. Maksimalna dužina naziva datoteka je 20 karaktera.

## 4 Ulaz i izlaz programa

---

### Primer 2

```
INTERAKCIJA SA PROGRAMOM:
```

```
prva.dat druga.dat
```

```
PRVA.DAT
```

```
Cena soka je 30
```

```
Cena vina je 150
```

```
Cena limunade je 200
```

```
Cena sendvica je 120
```

```
DRUGA.DAT
```

```
111121111211200211112111121120002111121111111211200021111211111112112000
```

**Zadatak 4.1.20** Ako je data tekstualna datoteka `plain.txt` napraviti tekstualnu datoteku `sifra.txt` tako što se svako slovo zamenjuje svojim prethodnikom (ciklično) suprotne velicine 'b' sa 'A', 'B' sa 'a', 'a' sa 'Z', 'A' sa 'z', itd. Podrazumevati da se na sistemu koristi tabela karaktera ASCII.

**Zadatak 4.1.21** Sa standardnog ulaza se učitava ime tekstualne datoteke i prirodan broj `k`. Podrazumeva se da zadata datoteka sadrži samo slova i beline i da je svaka reč iz datoteke dužine najviše 100. Program treba da učitava reči iz datoteke, da svaku reč rotira za `k` mesta i da tako dobijenu reč upiše u datoteku čije je ime `rotirano.txt`. Maksimalna dužina naziva datoteka je 20 karaktera.

**Zadatak 4.1.22** Napisati program koji u datoteku `izlaz.txt` prepisuje sve reči iz datoteke `ulaz.txt` čiji je zbir ascii kodova slova strogo veći od 1000. Reči su odvojene prazninama i nisu duže od 200 karaktera.

### Primer 1

```
ULAZ.TXT
```

```
Sa standardnog ulaza unosi se neoznaceni broj. Formirati novi broj koji se dobija izbacivanjem svake druge cifre iz polaznog broja.
```

```
IZLAZ.TXT
```

```
standardnog izbacivanjem
```

### Primer 2

```
ULAZ.TXT
```

```
i sada jedan kratak primer
```

```
p1: 1234567890
```

```
p2: ABCDEFGHIJ
```

```
p3: abcdefghij
```

```
IZLAZ.TXT
```

```
abcdefghijkl
```

### Primer 3

```
ULAZ.TXT
```

```
konstruisanje test-primera sa dugackim recima kao prestolonaslednik brojevima1234567890
```

```
IZLAZ.TXT
```

```
konstruisanje test-primera prestolonaslednik brojevima1234567890
```

### Primer 4

```
ULAZ.TXT
```

```
ima jos dugackih reci: predskazanje,
```

```
potom
```

```
nelogicnosti, zanemarivati, odugovlaciti, a ima
```

```
i i malih reci koje su kratke
```

```
predosecaj
```

```
IZLAZ.TXT
```

```
predskazanje, nelogicnosti,
```

```
zanemarivati, odugovlaciti,
```

```
predosecaj
```



**Zadatak 4.1.23** U datoteci `razno.txt` nalazi se tekst. U datoteku `palindromi.txt` prepisati sve reči iz datoteke `razno.txt` koje su palindromi. Reč je palindrom ako se čita isto sa leve i desne strane. Za reč smatramo niz karaktera koji se nalazi između belina i koji nije duži od 200 karaktera. Dozvoljeno je korišćenje specifikatora za čitanje reči. Maksimalan broj reči nije poznat. U slučaju greške ispisati -1 i prekinuti izvršavanje programa.

*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\datoteka{Oko kapAk radar}
```

**Zadatak 4.1.24** U datoteci čije se ime navodi na standardnom ulazu programa nalazi se broj  $n$ , a zatim i  $n$  reči (dužine najviše 50 karaktera). Napisati program koji učitava ovaj niz i

(a) ispisuje ga [3],

(b) iz njega uklanja sve duplikate i u datoteku `rez.txt` ispisuje transformisani niz [4]

U slučaju greške ispisati -1. Maksimalna dužina naziva datoteka je 20 karaktera.

*Primer 1*

```
INTERAKCIJA SA PROGRAMOM:
dat1.txt
DAT1.TXT
12 jha14 hahaha deda mraz deda
mraz deda deda jase konj konj konj
IZLAZ:
jha14 hahaha deda mraz deda mraz deda
deda jase konj konj konj
REZ.TXT:
jha14 hahaha deda mraz jase konj
```

*Primer 2*

```
INTERAKCIJA SA PROGRAMOM:
dat2.txt
DAT2.TXT
14
so secer supa so ljuto secer kiselo slatko
ljuto
paprika, ljuta paprika, ljuto dete
IZLAZ:
so secer supa so ljuto secer kiselo slatko
ljuto paprika, ljuta paprika, ljuto dete
REZ.TXT:
so secer supa ljuto kiselo slatko
paprika, ljuta dete
```

**Zadatak 4.1.25** U datoteci čije se ime navodi na standardnom ulazu programa nalazi se broj  $n$ , a zatim i  $n$  reči (dužine najviše 50 karaktera). Napisati program koji učitava ovaj niz i

(a) ispisuje ga, [3]

## 4 Ulaz i izlaz programa

(b) u datoteku `rez.txt` upisuje sve reči koje sadrže prvu reč i podvlaku. [4]

U slučaju greške ispisati -1. Maksimalna dužina naziva datoteka je 20 karaktera.

### Primer 1

```
INTERAKCIJA SA PROGRAMOM:
  dat1.txt
DAT1.TXT
  7 rec Opet _rec Reci rec_enica
  DVa recica_
IZLAZ:
  rec Opet _rec Reci rec_enica
  DVa recica_
REZ.TXT:
  _rec rec_enica recica_
```

### Primer 2

```
INTERAKCIJA SA PROGRAMOM:
  dat2.txt
DAT2.TXT
  11 Sunce sija iznad grada
  Sunce_Moje Jedan Dva Su_nce Sve Sunce123_123
  suncanica.
IZLAZ:
  Sunce sija iznad grada
  Sunce_Moje Jedan Dva Su_nce Sve Sunce123_123
  suncanica.
REZ.TXT:
  Sunce_Moje Sunce123_123
```

**Zadatak 4.1.26** Imena dve datoteke se zadaje na standardnom ulazu. U prvoj datoteci navedena je rec `r` i niz linija. Napisati program koji u drugu datoteku upisuje sve linije u kojima se reč `r` pojavljuje bar `n` puta, gde je `n` prirodan broj koji se unosi sa standardnog ulaza. Ispis treba da bude u formatu `broj_pojavljivanja: linija`. Linije brojati počevši od 1. Maksimalna dužina naziva datoteka je 20 karaktera.

**Zadatak 4.1.27** Napisati program koji poredi dva fajla i ispisuje redni broj linija u kojima se fajlovi razlikuju. Imena fajlova se zadaju kao argumenti komandne linije. U slučaju neuspešnog otvaranja datoteka ispisati poruku o grešci. Pretpostaviti da je maksimalna dužina reda u datoteci 200 karaktera. Ukoliko nisu zadati potrebni argumenti komadne linije ispisati poruku o grešci. Linije brojati počevši od 1.

### 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 primer1.dat primer2.dat
PRIMER1.DAT
  danas vezbamo
  analizu
  ovo je primer kad
  su datoteke razlicite
PRIEMR2.DAT
  danas vezbamo
  programiranje
  ovo je primer kad su
  datoteke razlicite
IZLAZ:
  2 3 4
```

*Primer 3*

```

POKRETANJE: ./a.out prva.dat
IZLAZ:
greska

```

*Primer 2*

```

POKRETANJE: ./a.out prva.dat druga.dat
PRVA.DAT
ovo je primer
kada su
datoteke
razlicite duzine
DRUGA.DAT
kada su
programiranje
datoteke
razlicite
duzine
i kada treba ispisati broj
tih redova
IZLAZ:
1 4 5 6 7

```

**Zadatak 4.1.28** Definirati strukturu

```

typedef struct{
    unsigned int a, b;
    char ime[5];
}_pravougaonik;

```

kojom se opisuje pravougaonik dužinama svojih stranica i imenom. Napisati program koji iz datoteke čije ime se zadaje kao argument komandne linije učitava pravougaonike (nepoznato koliko), a zatim ispisuje imena onih pravougaonika koji su kvadrati i vrednost najveće površine među pravougaonicima koji nisu kvadrati. U slučaju unosa nekorektnih dužina stranica pravougaonika ili neko-rektne vrednosti broja *n*, ispisati -1 i odmah prekinuti izvršavanje programa. Maksimalan broj pravougaonika je 200.

*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
```

### Primer 4

```
|| POKRETANJE: ./a.out primerx.dat
|| PRIMERX.DAT
|| 9 7 p
|| IZLAZ:
|| 63
```

### Primer 5

```
|| POKRETANJE: ./a.out prazna.dat
|| PRAZNA.DAT
|| IZLAZ:
```

**Zadatak 4.1.29** Ime datoteke dato je kao argument komandne linije. U datoteci se nalaze otvorene i zatvorene zagrade i još nekakav tekst. Proveriti da li su zagrade pravilno uparene. Npr. `ab( cd) ..` odgovor je `jesu`, a `..)ba()` odgovor je `nisu`. Ukoliko nisu zadati svi argumenti komandne linije ispisati poruku o grešci.

### Primer 1

```
|| POKRETANJE: ./a.out
|| zagrade.txt
|| ZAGRADE.TXT
|| ab( cd) ..
|| ((3+4)*5+1)*9
|| IZLAZ:
|| jesu
```

### Primer 2

```
|| POKRETANJE: ./a.out
|| primer2.dat
|| PRIMER2.DAT
|| (7+8
|| nisu(
|| uparene
|| IZLAZ:
|| nisu
```

### Primer 3

```
|| POKRETANJE: ./a.out
|| primer3.dat
|| PRIMER3.DAT
|| )) 7 + 6 ((
|| IZLAZ:
|| nisu
```

### Primer 4

```
|| POKRETANJE: ./a.out
|| IZLAZ:
|| greska
```

**Zadatak 4.1.30** Napraviti strukturu `STUDENT` koja sadrži:

- `ime` (u polju se čuva ime i prezime studenta, napr. "Marko Markovic", maksimalna dužina polja je 100 karaktera),
- `oc` (sadrži najviše 10 ocena studenta)
- `br_ocena` (ukupan broj ocena za studenata)
- `pr_oc` (prosečna ocena)

U datoteci se nalaze podaci o studentima. Za svakog studenta unosi se ime i prezime razdvojeno razmakom (uputstvo: može se korisiti `strcat` da spoji ime i prezime koji se mogu pročitati sa specifikatorom `%s`), a potom ocene koje se završavaju sa 0. Pronaći studenta koji ima najveći prosek i ispisati sve njegove podatke (prosek ispisati na 2 decimale). Maksimalan broj studenta je 100. Ime datoteke se zadaje kao argument komandne linije.

*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 7
|| 0
||
|| IZLAZ:
|| Jelena Jankovic 10 10 10 0
|| 10.00

```

*Primer 2*

```

|| POKRETANJE: ./a.out
|| IZLAZ:
|| greska

```

**Zadatak 4.1.31**

- Napisati C funkciju `int unesiSkup(char *s, FILE* f)` kojom se unosi skup elemenata iz datoteke F. Skup se predstavlja kao niz karaktera, pri čemu su dozvoljeni elementi skupa mala i velika slova abecede, kao i cifre. Unos se prekida kada se naiđe na znak za novi red ili nedozvoljeni karakter za skup (maksimalan broj elemenata skupa je 1000). Funkcija vraća broj elemenata skupa koji su uspesno učitani.
- Napisati funkciju `void prebroj(char *s, int *br_slova, int *br_cifara)` kojom se određuje broj slovnih elemenata skupa (velikih ili malih slova) kao i broj cifara u skupu.
- Napisati glavni program gde se unose podaci o skupu elemenata. Ime datoteke se zadaje kao argument komandne linije. Na standardni izlaz ispisati informacije o broju slova i cifara (koristiti funkcije pod a) i b)).

*Primer 1*

```

|| POKRETANJE: ./a.out skup.txt
|| SKUP.TXT
|| abc56ighj9012hjFGHH
|| IZLAZ:
|| broj slova: 13
|| broj cifara: 6

```

*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
IZLAZ:
greska
```

#### Zadatak 4.1.32 Definisati strukturu

```
typedef struct{
    int x;
    int y;
    int z;
} vektor;
```

kojom se opisuje trodimenzioni vektor. U datoteci `vektori.txt` nalazi se nepoznati broj vektora (maksimalno ih može biti 200). Učitati ih u niz i ispisuje na standardnom izlazu koordinate vektora sa najvećom dužinom. Dužina vektora se izračunava po formuli:

$$|v| = \sqrt{x^2 + y^2 + z^2}$$

U slučaju greške ispisati -1 i prekinuti izvršavanje programa.

#### Primer 1

```
VEKTORI.TXT
2
4 -1 7
3 1 2
IZLAZ:
4 -1 7
```

#### Primer 2

```
VEKTORI.TXT
67
IZLAZ:
-1
```

#### Primer 3

```
VEKTORI.TXT
3
0 0 0
0 1 0
1 0 0
IZLAZ:
0 1 0
```

#### Primer 4

```
VEKTORI.TXT
4
3 0 1
4 5 2
1 0 0
2 -1 2
IZLAZ:
4 5 2
```

**Zadatak 4.1.33** Prvi red datoteke `ulaz.txt` sadrži 2 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 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.