

## PROGRAMIRANJE 1



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

# **PROGRAMIRANJE 1**

## **Zbirka zadataka**

**Beograd  
2016.**

Autori:

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

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

*Danijela Simić*, asistent na Matematičkom fakultetu u Beogradu

*Anđelka Zečević*, asistent na Matematičkom fakultetu u Beogradu

PROGRAMIRANJE 1

Zbirka zadataka

# Sadržaj

0.1	Višedimenzioni nizovi	v
0.2	Rešenja	xix

## 0.1 Višedimenzioni nizovi

**Zadatak 0.1.1** Program učitava dva cela broja  $m$  i  $n$ , a potom i elemente matrice realnih brojeva dimenzije  $m \times n$ . Napisati program koji najpre učitava, a zatim i ispisuje vrednosti učitane matrice. Pretpostaviti da je maksimalna dimenzija matrice  $50 \times 50$ . U slučaju greške ispisati odovarajuću poruku.

### Primer 1

```
INTERAKCIJA SA PROGRAMOM:
Uneti dimenzije matrice: 3 4
Uneti matricu celih brojeva:
1 2 3 4
5 6 7 8
9 10 11 12
Matrica je:
1 2 3 4
5 6 7 8
9 10 11 12
```

### Primer 2

```
INTERAKCIJA SA PROGRAMOM:
Uneti dimenzije matrice: 5 3
Uneti matricu celih brojeva:
1 1 2
5 0 2
7 8 9
1 2 4
0 1 1
Matrica je:
1 1 2
5 0 2
7 8 9
1 2 4
0 1 1
```

### Primer 3

```
INTERAKCIJA SA PROGRAMOM:
Uneti dimenzije matrice: 500 3
Neispravna dimenzija matrice.
```

[Rešenje 0.1.1]

**Zadatak 0.1.2** Program učitava dva cela broja  $m$  i  $n$ , a potom i elemente matrice realnih brojeva dimenzije  $m \times n$ . Napisati program koji najpre učitava, a zatim i ispisuje vrednosti učitane matrice. Pretpostaviti da je maksimalna dimenzija matrice  $50 \times 50$ . U slučaju greške ispisati odovarajuću poruku.

*Primer 1*

```
|| INTERAKCIJA SA PROGRAMOM:
|| Uneti dimenzije matrice: 3 4
|| Uneti matricu celih brojeva:
|| 1 2 3 4
|| 5 6 7 8
|| 9 10 11 12
|| Euklidska norma je: 25.495
```

*Primer 2*

```
|| INTERAKCIJA SA PROGRAMOM:
|| Uneti dimenzije matrice: 5 3
|| Uneti matricu celih brojeva:
|| 1 1 2
|| 5 0 2
|| 7 8 9
|| 1 2 4
|| 0 1 1
|| Euklidska norma je: 15.875
```

*Primer 3*

```
|| INTERAKCIJA SA PROGRAMOM:
|| Uneti dimenzije matrice: 500 3
|| Neispravna dimenzija matrice.
```

[Rešenje 0.1.2]

**Zadatak 0.1.3** Napisati funkcije za rad sa celobrojnim matricama:

- (a) `void ucitavanje(int mat[][MAX], int* n, int* m)` kojom se prvo učitava dimezija matrice (brojevi  $m$  i  $n$ ), a potom i vrednosti matrice celih brojeva  $mat$ ,
- (b) `void ispis(int mat[][MAX], int n, int m)` kojom se ispisuje matrica  $mat$  dimenzije  $m \times n$ .

Napisati program koji najpre učitava, a zatim i ispisuje vrednosti učitane matrice. Pretpostaviti da je maksimalna dimenzija matrice  $50 \times 50$ . U slučaju greške ispisati odovarajuću poruku.

### Primer 1

```
INTERAKCIJA SA PROGRAMOM:
Uneti dimenzije matrice: 3 4
Uneti matricu celih brojeva:
1 2 3 4
5 6 7 8
9 10 11 12
Matrica je:
1 2 3 4
5 6 7 8
9 10 11 12
```

### Primer 2

```
INTERAKCIJA SA PROGRAMOM:
Uneti dimenzije matrice: 5 3
Uneti matricu celih brojeva:
1 1 2
5 0 2
7 8 9
1 2 4
0 1 1
Matrica je:
1 1 2
5 0 2
7 8 9
1 2 4
0 1 1
```

### Primer 3

```
INTERAKCIJA SA PROGRAMOM:
Uneti dimenzije matrice: 500 3
Neispravna dimenzija matrice.
```

[Rešenje 0.1.3]

**Zadatak 0.1.4** Napisati funkciju `void transponovana(int a[][max], int m, int n, int b[][max])` koja određuje matricu  $b$  koja je dobijena transponovanjem matrice  $a$ . Napisati program koji učitava matricu celih brojeva<sup>1</sup> i ispisuje odgovarajuću transponovanu matricu. Pretpostaviti da je maksimalna dimenzija matrice  $50 \times 50$ . U slučaju greške ispisati odovarajuću poruku.

### Primer 1

```
INTERAKCIJA SA PROGRAMOM:
Uneti dimenzije matrice: 3 4
Uneti matricu celih brojeva:
1 2 3 4
5 6 7 8
9 10 11 12
Transponovana matrica je:
1 5 9
2 6 10
3 7 11
4 8 12
```

### Primer 2

```
INTERAKCIJA SA PROGRAMOM:
Uneti dimenzije matrice: 5 3
Uneti matricu celih brojeva:
1 1 2
5 0 2
7 8 9
1 2 4
0 1 1
Transponovana matrica je:
1 5 7 1 0
1 0 8 2 1
2 2 9 4 1
```

---

<sup>1</sup>Pod pojmom *učitati matricu* uvek se podrazumeva da se prvo unose dimenzije matrice, a potom i sama matrica.

### Primer 3

```
INTERAKCIJA SA PROGRAMOM:
Uneti dimenzije matrice: 500 3
Neispravna dimenzija matrice.
```

[Rešenje 0.1.4]

**Zadatak 0.1.5** Napisati funkciju `void razmeni(int mat[][max], int m, int n, int k, int t)` u kojoj se razmenjuju elementi  $k$ -te i  $t$ -te vrste matrice  $mat$  dimezije  $m \times n$ . Učitati matricu celih brojeva, dva cela broja  $k$  i  $t$  i ispisati matricu dobijenu razmenjivanjem  $k$ -te i  $t$ -te vrste. Pretpostaviti da je maksimalna dimenzija matrice  $50 \times 50$ . U slučaju greške ispisati odovarajuću poruku.

### Primer 1

```
INTERAKCIJA SA PROGRAMOM:
Uneti dimenzije matrice: 3 4
Uneti matricu celih brojeva:
1 2 3 4
5 6 7 8
9 10 11 12
Uneti indkse vrsta: 0 2
9 10 11 12
5 6 7 8
1 2 3 4
```

### Primer 2

```
INTERAKCIJA SA PROGRAMOM:
Uneti dimenzije matrice: 5 3
Uneti matricu celih brojeva:
1 1 2
5 0 2
7 8 9
1 2 4
0 1 1
Uneti indkse vrsta: 1 3
1 1 2
1 2 4
7 8 9
5 0 2
0 1 1
```

### Primer 3

```
INTERAKCIJA SA PROGRAMOM:
Uneti dimenzije matrice: 5 3
Uneti matricu celih brojeva:
1 1 2
5 0 2
7 8 9
1 2 4
0 1 1
Uneti indkse vrsta: -1 50
Neispravni indkksi vrsta.
```

[Rešenje 0.1.5]

**Zadatak 0.1.6** Napisati program koji za učitanu matricu celih brojeva ispisuje indkse onih elemenata matrice koji su jednaki zbiru svih svojih susednih elemenata (pod susednim elementima podrazumevamo okolnih 8 polja matrice



ako postoje). Pretpostaviti da je maksimalna dimenzija matrice  $50 \times 50$ . U slučaju greške ispisati odovarajuću poruku.

*Primer 1*

```
INTERAKCIJA SA PROGRAMOM:
Uneti dimenzije matrice: 4 5
Uneti elemente matrice:
1 1 2 1 3
0 8 1 9 0
1 1 1 0 0
0 3 0 2 2
Indeksi elemenata koji su jednaki
zbiru suseda su:
1 1
3 1
3 4
```

*Primer 2*

```
INTERAKCIJA SA PROGRAMOM:
Uneti dimenzije matrice: 3 4
Uneti elemente matrice:
7 10 12 20
-1 -3 1 7
0 -47 2 0
Indeksi elemenata koji su jednaki
zbiru suseda su:
0 3
1 2
```

[Rešenje 0.1.6]

**Zadatak 0.1.7** Napisati funkciju koja formira niz  $b_0, b_1, \dots, b_n$  od matrice tako što je svaki element niza jednak srednjoj vrednosti elemenata vrste matrice. Napisati program koji za učitane matricu celih brojeva ispisuje dobijeni niz. Pretpostaviti da je maksimalna dimenzija matrice  $50 \times 50$ . U slučaju greške ispisati odovarajuću poruku.

*Primer 1*

```
INTERAKCIJA SA PROGRAMOM:
Uneti dimenzije matrice: 4 5
Uneti elemente matrice:
1 1 2 1 3
0 8 1 9 0
1 1 1 0 0
0 3 0 2 2
Dobijeni niz je:
1.6 3.6 0.6 1.4
```

*Primer 2*

```
INTERAKCIJA SA PROGRAMOM:
Uneti dimenzije matrice: 3 4
Uneti elemente matrice:
7 10 12 20
-1 -3 1 7
0 -47 2 0
Dobijeni niz je:
12.25 1 -11.25
```

[Rešenje 0.1.7]

**Zadatak 0.1.8** Relacija se može predstaviti kvadratnom matricom nula i jedinica na sledeći način: element  $i$  je u relaciji sa elementom  $j$  ukoliko se u preseku  $i$ -te vrste i  $j$ -te kolone nalazi broj 1, a nije u relaciji ukoliko se tu nalazi broj 0.

- (a) Napisati funkciju `int refleksivna(int a[][MAX], int n)` kojom se za relaciju zadatom matricom `a` dimenzije `n` ispituje da li je refleksivna.

- (b) Napisati funkciju `int simetricna(int a[][MAX], int n)` kojom se za relaciju zadatom matricom `a` ispituje da li je simetrična.
- (c) Napisati funkciju `int tranzitivna(int a[][MAX], int n)` kojom se za relaciju zadatom matricom `a` ispituje da li je tranzitivna.
- (d) Napisati funkciju `int ekvivalencija(int a[][MAX], int n)` kojom se za relaciju koja je zadata matricom `a` ispituje da li je relacija ekvivalencije. Koristiti prethodno napisane funkcije.

Program učitava ceo broj  $n$  i kvadratnu matricu dimenzije  $n \times n$ . Matrica može imati samo brojeve 0 i 1. Ispisati rezultat poziva funkcija. Pretpostaviti da je maksimalna dimenzija matrice  $50 \times 50$ . U slučaju greške ispisati odovarajuću poruku.

#### Primer 1

```
INTERAKCIJA SA PROGRAMOM:
Uneti dimenziju matrice: 4
Uneti elemente matrice:
1 0 0 0
0 1 1 0
0 0 1 0
0 0 0 0
Relacija nije refleksivna.
Relacija nije simetricna.
Relacija jeste tranzitivna.
Relacija nije ekvivalencija.
```

#### Primer 2

```
INTERAKCIJA SA PROGRAMOM:
Uneti dimenziju matrice: 4
Uneti elemente matrice:
1 1 0 0
1 1 1 0
0 0 1 0
0 0 0 1
Relacija jeste refleksivna.
Relacija jeste simetricna.
Relacija nije tranzitivna.
Relacija nije ekvivalencija.
```

[Rešenje 0.1.8]

**Zadatak 0.1.9** Data je kvadratna matrica dimenzije  $n \times n$ .

- (a) Napisati funkciju `float trag(float a[][MAX], int n)` koja računa trag matrice, odnosno zbir elemenata na glavnoj dijagonali matrice.
- (b) Napisati funkciju `float suma_sporredna(float a[][MAX], int n)` koja računa zbir elemenata na sporednoj dijagonali matrice.
- (c) Napisati funkciju `float suma_iznad(float a[][MAX], int n)` koja određuje sumu elemenata iznad glavne dijagonale.
- (d) Napisati funkciju `float suma_ispod(float a[][MAX], int n)` koja određuje sumu elemenata ispod sporedne dijagonale matrice.

Napisati program koji testira napisane funkcije. Napisati program koji za učitane matricu realnih brojeva ispisuje na tri decimale trag matrice, sumu na sporednoj dijagonali, sumu iznad glavne dijagonale i sumu elemenata ispod sporedne

dijagonale. Pretpostaviti da je maksimalna dimenzija matrice  $50 \times 50$ . U slučaju greške ispisati odovarajuću poruku.

#### Primer 1

```
INTERAKCIJA SA PROGRAMOM:
Uneti dimenziju matrice: 4
Uneti elemente matrice:
6 12.08 -1 20.5
8 90 -33.4 19.02
7.02 5 -20 14.5
8.8 -1 3 -22.8
Trag je 53.200.
Suma na sporednoj dijagonali je 0.900.
Suma iznad glavne dijagonale je 31.700.
Suma ispod sporedne dijagonale je -1.820.
```

#### Primer 2

```
INTERAKCIJA SA PROGRAMOM:
Uneti dimenziju matrice: 5
Uneti elemente matrice:
1 2 3 5 5
7 8 9 0 1
6 4 3 2 2
8 9 1 3 4
0 3 1 8 6
Trag je 21.000.
Suma na sporednoj dijagonali je 17.000.
Suma iznad glavne dijagonale je 33.000.
Suma ispod sporedne dijagonale je 24.000.
```

[Rešenje 0.1.9]

**Zadatak 0.1.10** Kvadratna matrica je donje trougaona ako se u gornjem trouglu (iznad glavne dijagonale, ne uključujući dijagonalu) nalaze sve nule. Napisati program koji za učitane kvadratnu matricu proverava da li je ona donje trougaona i ispisuje odgovarajuću poruku. Pretpostaviti da je maksimalna dimenzija matrice  $100 \times 100$ . U slučaju greške ispisati odovarajuću poruku.

#### Primer 1

```
INTERAKCIJA SA PROGRAMOM:
Uneti dimenziju matrice: 5
Uneti elemente matrice:
-1 0 0 0 0
2 10 0 0 0
0 1 5 0 0
7 8 20 14 0
-23 8 5 1 11
Matrica jeste donje trougaona.
```

#### Primer 2

```
INTERAKCIJA SA PROGRAMOM:
Uneti dimenziju matrice: 3
Uneti elemente matrice:
2 -2 1
1 2 2
2 1 -2
Matrica nije donje trougaona.
```

#### Primer 3

```
INTERAKCIJA SA PROGRAMOM:
Uneti dimenziju matrice: 200
Neispravna dimenzija matrice.
```

#### Primer 4

```
INTERAKCIJA SA PROGRAMOM:
Uneti dimenziju matrice: 4
Uneti elemente matrice:
2 0 0 0
7 80 0 0
-9 4 4 0
14 23 -8 1
Matrica jeste donje trougaona.
```

[Rešenje 0.1.10]

**Zadatak 0.1.11** Napisati program koji za učitano kvadratnu matricu ispisuje redni broj kolone koja ima najveći zbir elemenata. Pretpostaviti da je maksimalna dimenzija matrice  $50 \times 50$ . U slučaju greške ispisati odovarajuću poruku.

*Primer 1*

```
|| INTERAKCIJA SA PROGRAMOM:
|| Uneti dimenziju matrice: 3
|| Uneti elemente matrice:
|| 1 2 3
|| 7 3 4
|| 5 3 1
|| Indeks kolone je: 0
```

*Primer 2*

```
|| INTERAKCIJA SA PROGRAMOM:
|| Uneti dimenziju matrice: 4
|| Uneti elemente matrice:
|| 7 8 9 10
|| 7 6 11 4
|| 3 1 2 -2
|| 8 3 9 9
|| Indeks kolone je: 2
```

[Rešenje 0.1.11]

**Zadatak 0.1.12** Napisati program koji za učitano matricu realnih brojeva izračunava i ispisuje na dve decimale razliku između zbira elemenata gornjeg trougla i zbira elemenata donjeg trougla matrice. Gornji trougao čine svi elementi matrice koji su iznad glavne i sporedne dijagonale (ne računajući dijagonale), a donji trougao čine svi elementi ispod glavne i sporedne dijagonale (ne računajući dijagonale). Pretpostaviti da je maksimalna dimenzija matrice  $50 \times 50$ . U slučaju greške ispisati odovarajuću poruku.

*Primer 1*

```
|| INTERAKCIJA SA PROGRAMOM:
|| Uneti dimenziju matrice: 3
|| Uneti elemente matrice:
|| 2 3.2 4
|| 7 8.8 1
|| 2.3 1 1
|| Razlika je: 2.20
```

*Primer 2*

```
|| INTERAKCIJA SA PROGRAMOM:
|| Uneti dimenziju matrice: 5
|| Uneti elemente matrice:
|| 2.3 1 12 8 -20
|| 4 -8.2 7 14.5 19
|| 1 -2.5 9 11 33
|| 3 4.3 -5.7 2 8
|| 9 56 1.08 7 5.5 19.01
|| Razlika je:-30.38
```

[Rešenje 0.1.12]

**Zadatak 0.1.13** Napisati program koji za učitano matricu dimenzije  $m \times n$  i uneta dva broja  $p$  i  $k$  ( $p \leq m$ ,  $k \leq n$ ) ispisuje sume svih podmatrica dimenzije  $p \times k$  unete matrice. Pretpostaviti da je maksimalna dimenzija matrice  $50 \times 50$ . U slučaju greške ispisati odovarajuću poruku. NAPOMENA: *Nije bitan redosled kojim se ispisuju sume.*

### Primer 1

```
INTERAKCIJA SA PROGRAMOM:
Uneti dimenzije matrice: 3 4
Uneti matricu celih brojeva:
1 2 3 4
5 6 7 8
9 10 11 12
Uneti dva cela broja: 3 3
Sume podmatrica su: 54 63
```

### Primer 2

```
INTERAKCIJA SA PROGRAMOM:
Uneti dimenzije matrice: 3 4
Uneti matricu celih brojeva:
1 2 3 4
5 6 7 8
9 10 11 12
Uneti dva cela broja: 2 3
Sume podmatrica su: 24 30 48 54
```

### Primer 3

```
INTERAKCIJA SA PROGRAMOM:
Uneti dimenzije matrice: -3 200
Neispravna dimenzija matrice.
```

### Primer 4

```
INTERAKCIJA SA PROGRAMOM:
Uneti dimenzije matrice: 5 3
Uneti matricu celih brojeva:
1 1 2
5 0 2
7 8 9
1 2 4
0 1 1
Uneti dva cela broja: 2 2
Sume podmatrica su: 7 5 20 19 18 23 4 8
```

[Rešenje 0.1.13]

**Zadatak 0.1.14** Napisati program koji učitava vrednosti matrice i ispituje da li su elementi matrice po kolonama, vrstama i dijagonalama (glavnoj i sporednoj) sortirani strogo rastuće. Ispisati odgovarajuće poruke. Pretpostaviti da je maksimalna dimenzija matrice  $50 \times 50$ . U slučaju greške ispisati odovarajuću poruku.

### Primer 1

```
INTERAKCIJA SA PROGRAMOM:
Unesi dimenziju matrice: 3
Unesi elemente matrice:
1 2 3
4 5 6
7 8 9
Elementi su sortirani po kolonama.
Elementi su sortirani po vrstama.
Elementi su sortirani po dijagonalama.
```

### Primer 2

```
INTERAKCIJA SA PROGRAMOM:
Unesi dimenziju matrice: 2
Unesi elemente matrice:
6 9
4 10
Elementi nisu sortirani po kolonama.
Elementi su sortirani po vrstama.
Elementi nisu sortirani po dijagonalama.
```

### Primer 3

```
INTERAKCIJA SA PROGRAMOM:
Unesi dimenziju matrice: 4
Unesi elemente matrice:
5 5 7 9
6 10 11 13
8 12 14 15
13 15 16 20
Elementi su sortirani po kolonama.
Elementi nisu sortirani po vrstama.
Elementi su sortirani po dijagonalama.
```

### Primer 4

```
INTERAKCIJA SA PROGRAMOM:
Unesi dimenziju matrice: 1
Unesi elemente matrice:
5
Elementi su sortirani po kolonama.
Elementi su sortirani po vrstama.
Elementi su sortirani po dijagonalama.
```

[Rešenje 0.1.14]

**Zadatak 0.1.15** Napisati program koji učitava vrednosti matrice i ispituje da li su zbrovi elemenata kolona matrice uredjeni u strogo rastućem poretku i ispisuje odgovarajuću poruku. Pretpostaviti da je maksimalna dimenzija matrice koja se unosi  $10 \times 10$ . U slučaju greške ispisati odovarajuću poruku.

### Primer 1

```
INTERAKCIJA SA PROGRAMOM:
Uneti dimenziju matrice: 4
Uneti elemente matrice:
1 0 0 0
0 0 1 0
0 0 0 1
0 1 0 0
Sume nisu uredjenje strogo rastuce.
```

### Primer 2

```
INTERAKCIJA SA PROGRAMOM:
Uneti dimenziju matrice: 3
Uneti elemente matrice:
1 2 3
4 5 6
7 8 9
Sume jesu uredjenje strogo rastuce.
```

### Primer 3

```
INTERAKCIJA SA PROGRAMOM:
Uneti dimenziju matrice: 3
Uneti elemente matrice:
2 -2 1
1 2 2
2 1 -2
Sume nisu uredjenje strogo rastuce.
```

### Primer 4

```
INTERAKCIJA SA PROGRAMOM:
Uneti dimenziju matrice: 5
Uneti elemente matrice:
-1 0 3 0 20
0 0 0 10 0
0 0 -1 0 0
0 1 0 0 0
0 0 0 0 -1
Sume jesu uredjenje strogo rastuce.
```

[Rešenje 0.1.15]

**Zadatak 0.1.16** Matrica je *ortonormirana* ako je skalarni proizvod svakog para različitih vrsta jednak 0, a skalarni proizvod vrste sa samom sobom 1. Proveriti da li je uneta matrica ortonormirana i ispisati odgovarajuću poruku. U slučaju greške ispisati odgovarajuću poruku. NAPOMENA: *Skalarni proizvod vektora*  $a = (a_1, a_2, \dots, a_n)$  *i*  $b = (b_1, b_2, \dots, b_n)$  *je*  $a_1 \cdot b_1 + a_2 \cdot b_2 + \dots + a_n \cdot b_n$ .

#### Primer 1

```
INTERAKCIJA SA PROGRAMOM:
Uneti dimenziju matrice: 4
Uneti elemente matrice:
1 0 0 0
0 0 1 0
0 0 0 1
0 1 0 0
Matrica jeste ortonormirana.
```

#### Primer 2

```
INTERAKCIJA SA PROGRAMOM:
Uneti dimenziju matrice: 3
Uneti elemente matrice:
1 2 3
4 5 6
7 8 9
Matrica nije ortonormirana.
```

#### Primer 3

```
INTERAKCIJA SA PROGRAMOM:
Uneti dimenziju matrice: 3
Uneti elemente matrice:
2 -2 1
1 2 2
2 1 -2
Matrica nije ortonormirana.
```

#### Primer 4

```
INTERAKCIJA SA PROGRAMOM:
Uneti dimenziju matrice: 5
Uneti elemente matrice:
-1 0 0 0 0
0 0 0 1 0
0 0 -1 0 0
0 1 0 0 0
0 0 0 0 -1
Matrica jeste ortonormirana.
```

[Rešenje 0.1.16]

**Zadatak 0.1.17** Datu kvadratna matrica je *magični kvadrat* ako je suma elemenata u svakoj koloni i svakoj vrsti jednaka. Napisati program koji proverava da li data kvadratna matrica je ona magični kvadrat i ispisuje odgovarajuću poruku na standardni izlaz. Maksimalna dimenzija matrice je  $50 \times 50$ .

#### Primer 1

```
INTERAKCIJA SA PROGRAMOM:
Uneti dimenziju matrice: 4
Uneti elemente matrice:
1 5 3 1
2 1 2 5
3 2 2 3
4 2 3 1
Matrica jeste magicni kvadrat.
```

#### Primer 2

```
INTERAKCIJA SA PROGRAMOM:
Uneti dimenziju matrice: 3
Uneti elemente matrice:
1 2 3
4 5 6
-1 3 3
Matrica nije magicni kvadrat.
```

[Rešenje 0.1.17]

**\* Zadatak 0.1.18** Ispisati elemente učitane matrice celih brojeva u grupama koje su paralelne sa sporednom dijagonalom matrice počevši od gornjeg levog ugla matrice. Maksimalna dimenzija matrice je  $100 \times 100$ . U slučaju greške ispisati odgovarajuću poruku.

### Primer 1

```
INTERAKCIJA SA PROGRAMOM:
Uneti dimenziju matrice: 3
Uneti elemente matrice:
1 2 3
4 5 6
7 8 9
Ispis je:
1
2 4
3 5 7
6 8
9
```

### Primer 2

```
INTERAKCIJA SA PROGRAMOM:
Uneti dimenziju matrice: 5
Uneti elemente matrice:
7 -8 1 2 3
90 11 0 5 4
12 -9 14 23 8
80 6 88 17 62
-22 10 44 57 -200
Ispis je:
7
-8 90
1 11 12
2 0 -9 80
3 5 14 6 -22
4 23 88 10
8 17 44
62 57
-200
```

[Rešenje 0.1.18]

**\* Zadatak 0.1.19** Napisati funkciju `void mnozenje(int a[][max], int m, int n, int b[][max], int k, int t, int c[][max])` koja računa matricu  $c$  kao proizvod matrica  $a$  i  $b$ . Dimenzija matrice  $a$  je  $n \times m$ , a dimenzija matrice  $b$  je  $k \times t$ . Napisati program koji ispisuje proizvod učitanih matrica. Ukoliko množenje matrica nije moguće ili je došlo do greške prilikom unosa podataka ispisati odgovarajuću poruku. Maksimalna dimenzija obe matrice je  $50 \times 50$ .

### Primer 1

```
INTERAKCIJA SA PROGRAMOM:
Uneti dimenzije matrice: 3 4
Uneti elemente matrice:
1 2 8 9
-4 5 2 3
7 6 4 10
Uneti dimenzije matrice: 4 2
Uneti elemente matrice:
11 5
6 7
8 9
0 -3
Rezultat mnozenja je:
87 64
2 24
145 83
```

### Primer 2

```
INTERAKCIJA SA PROGRAMOM:
Uneti dimenzije matrice: 5 2
Uneti elemente matrice:
1 7
9 0
-10 2
92 3
14 -8
Uneti dimenzije matrice: 2 4
Uneti elemente matrice:
7 8 9 10
-11 2 34 78
Rezultat mnozenja je:
-70 22 247 556
63 72 81 90
-92 -76 -22 56
611 742 930 1154
186 96 -146 -484
```



### Primer 2

```
INTERAKCIJA SA PROGRAMOM:
Uneti dimenzije matrice: 3 4
Uneti elemente matrice:
1 2 8 9
-4 5 2 3
7 6 4 10
Uneti dimenzije matrice: 5 2
Uneti elemente matrice:
11 5
6 7
8 9
0 -3
4 4
Mnozenje matrica nije moguće.
```

[Rešenje 0.1.19]

\* **Zadatak 0.1.20** Element matrice naziva se *sedlo* ako je istovremeno najmanji u svojoj vrsti, a najveći u svojoj koloni. Napisati program koji ispisuje indekse i vrednosti onih elemenata matrice realnih brojeva koji su sedlo. Maksimalna dimenzija matrice je  $50 \times 50$ . U slučaju greške ispisati odgovarajuću poruku.

### Primer 1

```
INTERAKCIJA SA PROGRAMOM:
Uneti dimenzije matrice: 2 3
1 2 3
0 5 6
Sedlo: 0 0 1
```

### Primer 2

```
INTERAKCIJA SA PROGRAMOM:
Uneti dimenziju matrice: 3 3
10 3 20
15 5 100
30 -1 200
Sedlo: 1 1 5
```

[Rešenje 0.1.20]

\* **Zadatak 0.1.21** Napisati program koji ispisuje elemente matrice celih brojeva u spiralnom redosledu počevši od gornjeg levog ugla krećući se u smeru suprotnom od smera kazaljke na satu. Maksimalna dimenzija matrice je  $50 \times 50$ . U slučaju greške ispisati odgovarajuću poruku.

### Primer 1

```
INTERAKCIJA SA PROGRAMOM:
Uneti dimenziju matrice: 3 3
Uneti elemente matrice:
1 2 3
4 5 6
7 8 9
Ispis je:
1 2 3 6 9 8 7
4 5
```

### Primer 2

```
INTERAKCIJA SA PROGRAMOM:
Uneti dimenziju matrice: 5 7
Uneti elemente matrice:
7 -8 1 2 3 -54 87
90 11 0 5 4 9 18
12 -9 14 23 8 -22 74
80 6 88 17 62 38 41
-22 10 44 57 -200 39 55
Ispis je:
7 -8 1 2 3 -54 87 18 74 41 55
39 -200 57 44 10 -22 80 12 90
11 0 5 4 9 -22 38 62 17 88 6
-9 14 23 8
```

[Rešenje 0.1.21]

**\* Zadatak 0.1.22** Matrica  $a$  se sadrži u matrici  $b$  ukoliko postoji podmatrica matrice  $b$  identična matrici  $a$ . Napisati program koji za dve učitane matrice celih brojeva proverava da li se druga matrica sadrži u prvoj učitanoj matrici. Maksimalna dimenzija obe matrice je  $50 \times 50$ .

### Primer 1

```
INTERAKCIJA SA PROGRAMOM:
Uneti dimenzije matrice: 3 4
Uneti elemente matrice:
1 2 8 9
-4 5 2 3
7 6 4 10
Uneti dimenzije matrice: 2 2
Uneti elemente matrice:
2 3
4 10
Druga matrica je sadržana u prvoj matrici.
```

### Primer 2

```
INTERAKCIJA SA PROGRAMOM:
Uneti dimenzije matrice: 3 4
Uneti elemente matrice:
1 2 8 9
-4 5 2 3
7 6 4 10
Uneti dimenzije matrice: 2 2
Uneti elemente matrice:
2 8
6 4
Druga matrica nije sadržana u prvoj matrici.
```

### Primer 3

```
INTERAKCIJA SA PROGRAMOM:
Uneti dimenzije matrice: 5 5
Uneti elemente matrice:
7 -8 1 2 3
90 11 0 5 4
12 -9 14 23 8
80 6 88 17 62
-22 10 44 57 -200
Uneti dimenzije matrice: 3 4
Uneti elemente matrice:
90 11 0 5
12 -9 14 23
80 6 88 17
Druga matrica je sadržana u prvoj matrici.
```

## 0.2 Rešenja

### Rešenje 0.1.1

```
1  #include <stdio.h>
   #include <stdlib.h>
3
   #define MAX 50
5
   int main()
7 {
   int mat[MAX][MAX];
9   int m, n;
11
   int i, j;
13
   printf("Uneti dimenzije matrice: ");
   scanf("%d%d", &m, &n);
15
   if (n <= 0 || n > MAX || m <= 0 || m > MAX)
17 {
       printf("Neispravna dimenzija matrice\n");
       exit(EXIT_FAILURE);
19   }
21
   printf("Uneti matricu celih brojeva\n");
23
   for(i=0; i<m; i++)
       for(j=0; j<n; j++)
25           scanf("%d", &mat[i][j]);
27
   /* Ispis elemenata matrice. */
   for(i=0; i<m; i++)
   {
31       for(j=0; j<n; j++)
           printf("%d ", mat[i][j]);
33       printf("\n");
   }
35
   return 0;
37 }
```

### Rešenje 0.1.2

```

1 #include <stdio.h>
  #include <stdlib.h>
3 #include <math.h>

5 #define MAX 50

7 int main()
  {
9     int mat[MAX][MAX];
      int m, n;
11    int suma = 0;

13    int i, j;

15    printf("Uneti dimenzije matrice: ");
      scanf("%d%d", &m, &n);

17    if (n <= 0 || n > MAX || m <= 0 || m > MAX)
19    {
        printf("Neispravna dimenzija matrice\n");
21        exit(EXIT_FAILURE);
    }

23    printf("Uneti matricu celih brojeva\n");

25    for(i=0; i<m; i++)
27        for(j=0; j<n; j++)
            scanf("%d", &mat[i][j]);

29

31    for(i=0; i<m; i++)
33        for(j=0; j<n; j++)
            suma += mat[i][j] * mat[i][j];

35    printf("Euklidska norma je %.3lf.\n", sqrt(suma));

37    return 0;

39 }

```

### Rešenje 0.1.3

```

1 #include <stdio.h>
  #include <stdlib.h>
3
5 #define MAX 50

7 void učitavanje(int mat[][MAX], int* m, int* n)
  {
      int i, j;

```

```

9      printf("Uneti dimenzije matrice: ");
11     scanf("%d%d", m, n);

13     if (*n <= 0 || *n > MAX || *m <= 0 || *m > MAX)
14     {
15         printf("Neispravna dimenzija matrice\n");
16         exit(EXIT_FAILURE);
17     }

19     printf("Uneti matricu celih brojeva\n");

21     for(i=0; i<*m; i++)
22         for(j=0; j<*n; j++)
23             scanf("%d", &mat[i][j]);
24 }

25 void ispis(int mat[][MAX], int m, int n)
26 {
27     int i, j;

29     for(i=0; i<m; i++)
30     {
31         for(j=0; j<n; j++)
32             printf("%d ", mat[i][j]);
33         printf("\n");
34     }
35 }

37 int main()
38 {
39     int mat[MAX][MAX];
40     int m, n;

42     ucitavanje(mat, &m, &n);
43     ispis(mat, m, n);

45     return 0;
47 }

```

### Rešenje 0.1.4

```

1  #include <stdio.h>
2  #include <stdlib.h>
3
4  #define MAX 50
5
6  void ucitavanje(int mat[][MAX], int* m, int* n)
7  {
8      int i, j;
9

```

```

11     printf("Uneti dimenzije matrice: ");
    scanf("%d%d", m, n);

13     if (*n <= 0 || *n > MAX || *m <= 0 || *m > MAX)
    {
15         printf("Neispravna dimenzija matrice\n");
        exit(EXIT_FAILURE);
17     }

19     printf("Uneti matricu celih brojeva\n");

21     for(i=0; i<*m; i++)
        for(j=0; j<*n; j++)
23         scanf("%d", &mat[i][j]);
    }

25 void ispis(int mat[][MAX], int m, int n)
27 {
    int i, j;

29     for(i=0; i<m; i++)
    {
31         for(j=0; j<n; j++)
33             printf("%d ", mat[i][j]);
        printf("\n");
35     }
    }

37 void transponovana(int a[][MAX], int m, int n, int b[][MAX])
39 {
    int i, j;

41     for(i=0; i<m; i++)
43         for(j=0; j<n; j++)
            b[j][i] = a[i][j];
45 }

47 int main()
49 {
    int mat[MAX][MAX], t[MAX][MAX];
    int m, n;

51     ucitavanje(mat, &m, &n);
53     transponovana(mat, m, n, t);
    ispis(t, n, m);

55     return 0;
57 }

```

## Rešenje 0.1.5

```

1  #include <stdio.h>
2  #include <stdlib.h>
3
4  #define MAX 50
5
6  void učitavanje(int mat[][MAX], int* m, int* n)
7  {
8      int i, j;
9
10     printf("Uneti dimenzije matrice: ");
11     scanf("%d%d", m, n);
12
13     if (*n <= 0 || *n > MAX || *m <= 0 || *m > MAX)
14     {
15         printf("Neispravna dimenzija matrice\n");
16         exit(EXIT_FAILURE);
17     }
18
19     printf("Uneti matricu celih brojeva\n");
20
21     for(i=0; i<*m; i++)
22         for(j=0; j<*n; j++)
23             scanf("%d", &mat[i][j]);
24 }
25
26 void ispis(int mat[][MAX], int m, int n)
27 {
28     int i, j;
29
30     for(i=0; i<m; i++)
31     {
32         for(j=0; j<n; j++)
33             printf("%d ", mat[i][j]);
34         printf("\n");
35     }
36 }
37
38 void razmeni(int mat[][MAX], int m, int n, int k, int t)
39 {
40     int j, pom;
41
42     for(j = 0; j < n; j++)
43     {
44         pom = mat[k][j];
45         mat[k][j] = mat[t][j];
46         mat[t][j] = pom;
47     }
48 }
49
50 int main()
51 {

```

```

52     int mat[MAX][MAX];
53     int m, n;
54     int k, t;

56     učitavanje(mat, &m, &n);

58     printf("Uneti indekse vrsta: ");
59     scanf("%d%d", &k, &t);

60
61     if (k < 0 || k >= m || t < 0 || t >= m)
62     {
63         printf("Neispravni indeksi vrsta.\n");
64         return -1;
65     }

66     razmeni(mat, m, n, k, t);

68     ispis(mat, m, n);

70     return 0;
71 }

```

## Rešenje 0.1.6

```

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

#define MAX 50

void učitavanje(int mat[][MAX], int* m, int* n)
{
    int i, j;

    printf("Uneti dimenzije matrice: ");
    scanf("%d%d", m, n);

    if (*m <= 0 || *n > MAX || *m <= 0 || *m > MAX)
    {
        printf("Neispravna dimenzija matrice\n");
        exit(EXIT_FAILURE);
    }

    printf("Uneti matricu celih brojeva\n");

    for(i=0; i<*m; i++)
        for(j=0; j<*n; j++)
            scanf("%d", &mat[i][j]);
}

int main()

```



```

28 {
    int mat[MAX][MAX];
30     int m, n, i, j, suma;
    int k, t;

32     učitavanje(mat, &m, &n);

34     printf("Indeksi elemenata koji su jednaki zbiru suseda su:\n");
    for(i=0; i<m; i++)
36         for(j=0; j<n; j++)
38             {
                suma = 0;

40                 for(k=-1; k<=1; k++)
42                     for(t=-1; t<=1; t++)
                        if (i+k >= 0 && i+k < n && j+t >= 0 && j+t < n)
44                             suma += mat[i+k][j+t];

46                 if (suma - mat[i][j] == mat[i][j])
                    printf("%d %d\n", i, j);
48             }

50     return 0;
}

```

### Rešenje 0.1.7

```

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

4 #define MAX 50

6 void učitavanje(int mat[][MAX], int* m, int* n)
{
8     int i, j;

10     printf("Uneti dimenzije matrice: ");
    scanf("%d%d", m, n);

12     if (*n <= 0 || *n > MAX || *m <= 0 || *m > MAX)
14     {
        printf("Neispravna dimenzija matrice\n");
16         exit(EXIT_FAILURE);
    }

18     printf("Uneti matricu celih brojeva\n");

20     for(i=0; i<*m; i++)
22         for(j=0; j<*n; j++)
            scanf("%d", &mat[i][j]);
24 }

```

```

26 void kreiraj_niz(int mat[][MAX], int m, int n, double b[])
27 {
28     int i, j, suma;
29
30     for(i=0; i<m; i++)
31     {
32         suma = 0;
33         for(j=0; j<n; j++)
34             suma += mat[i][j];
35
36         b[i] = (double)suma/n;
37     }
38 }
39
40 int main()
41 {
42     int mat[MAX][MAX];
43     double b[MAX];
44     int m, n, i;
45
46     učitavanje(mat, &m, &n);
47
48     kreiraj_niz(mat, m, n, b);
49
50     printf("Dobijeni niz je:\n");
51     for(i=0; i<m; i++)
52         printf("%g ", b[i]);
53     printf("\n");
54
55     return 0;
56 }

```

## Rešenje 0.1.8

```

1  #include <stdio.h>
2  #include <stdlib.h>
3
4  #define MAX 50
5
6  void učitavanje(int mat[][MAX], int* n)
7  {
8      int i, j;
9
10     printf("Uneti dimenzije matrice: ");
11     scanf("%d", n);
12
13     if (*n <= 0 || *n > MAX)
14     {
15         printf("Neispravna dimenzija matrice\n");
16         exit(EXIT_FAILURE);
17     }
18 }

```

```

17     }

19     printf("Uneti matricu celih brojeva\n");

21     for(i=0; i<*n; i++)
22         for(j=0; j<*n; j++)
23             scanf("%d", &mat[i][j]);
24 }

25 int reflektivna(int a[][MAX], int n)
26 {
27     int i;

29     for(i=0; i<n; i++)
30         if (a[i][i] != 1)
31             return 0;

33     return 1;
34 }

35 int simetricna(int a[][MAX], int n)
36 {
37     int i, j;

41     for(i=0; i<n; i++)
42         for(j=0; j<n; j++)
43             if (a[i][j] != a[j][i])
44                 return 0;

45     return 1;
46 }

47 int tranzitivna(int a[][MAX], int n)
48 {
49     int i, j, k;

53     for(i=0; i<n; i++)
54         for(j=0; j<n; j++)
55             for(k=0; k<n; k++)
56                 if (a[i][j] == 1 && a[j][k] == 1 && a[i][k] == 0)
57                     return 0;

59     return 1;
60 }

61 int ekvivalencija(int a[][MAX], int n)
62 {
63     if (reflektivna(a, n) && simetricna(a, n) && tranzitivna(a, n))
64         return 1;

66     return 0;
67 }

```

```

69 int main()
70 {
71     int a[MAX][MAX];
72     int n;
73
74     učitavanje(a, &n);
75
76     if (refleksivna(a, n))
77         printf("Relacija jeste refleksivna.\n");
78     else
79         printf("Relacija nije refleksivna.\n");
80
81     if (simetricna(a, n))
82         printf("Relacija jeste simetricna.\n");
83     else
84         printf("Relacija nije simetricna.\n");
85
86     if (tranzitivna(a, n))
87         printf("Relacija jeste tranzitivna.\n");
88     else
89         printf("Relacija nije tranzitivna.\n");
90
91     if (ekvivalencija(a, n))
92         printf("Relacija jeste ekvivalencija.\n");
93     else
94         printf("Relacija nije ekvivalencija.\n");
95
96     return 0;
97 }

```

### Rešenje 0.1.9

```

1 #include <stdio.h>
2 #include <stdlib.h>
3
4 #define MAX 50
5
6 void učitavanje(float mat[][MAX], int* n)
7 {
8     int i, j;
9
10    printf("Uneti dimenzije matrice: ");
11    scanf("%d", n);
12
13    if (*n <= 0 || *n > MAX)
14    {
15        printf("Neispravna dimenzija matrice\n");
16        exit(EXIT_FAILURE);
17    }
18 }

```

```
printf("Uneti matricu celih brojeva\n");
20
for(i=0; i<*n; i++)
22     for(j=0; j<*n; j++)
        scanf("%f", &mat[i][j]);
24 }

float trag(float a[][MAX], int n)
26 {
    float suma = 0;
    int i;
    for(i=0; i<n; i++)
32         suma += a[i][i];

    return suma;
34 }

float suma_sporedna(float a[][MAX], int n)
36 {
    float suma = 0;
    int i;
    for(i=0; i<n; i++)
42         suma += a[i][n-i-1];

    return suma;
44 }

float suma_iznad(float a[][MAX], int n)
46 {
    float suma = 0;
    int i, j;
    for(i=0; i<n; i++)
54         for(j=i+1; j<n; j++)
            suma += a[i][j];

    return suma;
56 }

float suma_ispod(float a[][MAX], int n)
58 {
    float suma = 0;
    int i, j;
    for(i=0; i<n; i++)
64         for(j=n-i-1; j>i; j--)
            suma += a[i][j];

    return suma;
66 }
70 }
```

```

72 int main()
{
74     float a[MAX][MAX];
    int n;

76     učitavanje(a, &n);

78     printf("Trag je %.3f.\n", trag(a, n));
    printf("Suma na sporednoj dijagonali je %.3f.\n", suma_sporedna(a,
80         n));
    printf("Suma iznad glavne dijagonale je %.3f.\n", suma_iznad(a, n))
    ;
82     printf("Suma ispod sporedne dijagonale je %.3f.\n", suma_ispod(a, n
        ));
84     return 0;
}

```

## Rešenje 0.1.10

```

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

4 #define MAX 50

6 void učitavanje(int mat[][MAX], int* n)
{
8     int i, j;

10     printf("Uneti dimenzije matrice: ");
    scanf("%d", n);

12     if (*n <= 0 || *n > MAX)
    {
14         printf("Neispravna dimenzija matrice\n");
        exit(EXIT_FAILURE);
    }

18     printf("Uneti matricu celih brojeva\n");

20     for(i=0; i<*n; i++)
        for(j=0; j<*n; j++)
22         scanf("%d", &mat[i][j]);
24 }

26 int donje_trougona(int a[][MAX], int n)
{
28     int i, j;

30     for(i=0; i<n; i++)

```

```

    for(j=i+1; j<n; j++)
32     if (a[i][j] != 0)
        return 0;
34
    return 1;
36 }

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

42     ucitavanje(a, &n);

44     if (donje_trougaona(a, n))
46         printf("Matrica jeste donje trougaona.\n");
    else
48         printf("Matrica nije donje trougaona.\n");

50     return 0;
}

```

## Rešenje 0.1.11

```

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

4  #define MAX 50

6  void ucitavanje(int mat[][MAX], int* n)
{
8     int i, j;

10     printf("Uneti dimenzije matrice: ");
    scanf("%d", n);

12     if (*n <= 0 || *n > MAX)
14     {
        printf("Neispravna dimenzija matrice\n");
16         exit(EXIT_FAILURE);
    }

18     printf("Uneti matricu celih brojeva\n");

20     for(i=0; i<*n; i++)
        for(j=0; j<*n; j++)
22             scanf("%d", &mat[i][j]);
24 }

26 int main()
{

```

```

28     int a[MAX][MAX];
29     int n, i, j;
30     int max_zbir, trenutni_zbir = 0, indeks_kolone = 0;

31
32     učitavanje(a, &n);

33
34     for(i=0; i<n; i++)
35         trenutni_zbir += a[i][0];

36     max_zbir = trenutni_zbir;

37
38     for(j=1; j<n; j++)
39     {

40
41         trenutni_zbir = 0;
42         for(i=0; i<n; i++)
43             trenutni_zbir += a[i][j];

44
45         if (trenutni_zbir > max_zbir)
46         {
47             max_zbir = trenutni_zbir;
48             indeks_kolone = j;
49         }
50     }

51
52     printf("Indeks kolone je: %d\n", indeks_kolone);

53
54     return 0;
55 }

```

## Rešenje 0.1.12

```

1  #include <stdio.h>
2  #include <stdlib.h>
3
4  #define MAX 50
5
6  void učitavanje(float mat[][MAX], int* n)
7  {
8      int i, j;
9
10     printf("Uneti dimenzije matrice: ");
11     scanf("%d", n);
12
13     if (*n <= 0 || *n > MAX)
14     {
15         printf("Neispravna dimenzija matrice\n");
16         exit(EXIT_FAILURE);
17     }
18
19     printf("Uneti matricu celih brojeva\n");

```



```

21     for(i=0; i<*n; i++)
22         for(j=0; j<*n; j++)
23             scanf("%f", &mat[i][j]);
24 }
25
26 int main()
27 {
28     float a[MAX][MAX];
29     int n, i, j;
30     float gornji_trougao = 0, donji_trougao = 0;
31
32     ucitavanje(a, &n);
33
34     for(i=0; i<n/2; i++)
35         for(j=i+1; j<n-i-1; j++)
36             gornji_trougao += a[i][j];
37
38     for(i=n/2; i<n; i++)
39         for(j=n-i; j<i; j++)
40             donji_trougao += a[i][j];
41
42     printf("%f %f\n", gornji_trougao, donji_trougao);
43
44     printf("Razlika je: %.2f\n", gornji_trougao - donji_trougao);
45
46     return 0;
47 }

```

### Rešenje 0.1.13

```

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

#define MAX 50

void ucitavanje(int mat[][MAX], int* m, int* n)
{
    int i, j;

    printf("Uneti dimenzije matrice: ");
    scanf("%d%d", m, n);

    if (*n <= 0 || *n > MAX || *m <= 0 || *m > MAX)
    {
        printf("Neispravna dimenzija matrice\n");
        exit(EXIT_FAILURE);
    }

    printf("Uneti matricu celih brojeva\n");
}

```

```

22     for(i=0; i<*m; i++)
        for(j=0; j<*n; j++)
            scanf("%d", &mat[i][j]);
24 }

26 int main()
{
28     int a[MAX][MAX];
    int n, i, j, m, x, y, p, k;
30     int suma;

32     učitavanje(a, &m, &n);

34     printf("Uneti dva cela broja: ");
    scanf("%d%d", &p, &k);

36     printf("Sume podmatrica su: ");
    for(i=0; i<= m-p; i++)
        for(j=0; j<= n-k; j++)
40         {
            suma = 0;
42             for(x=0; x<p; x++)
                for(y=0; y<k; y++)
44                 suma += a[i+x][j+y];

46             printf("%d ", suma);
        }

48     printf("\n");

50     return 0;
52 }

```

### Rešenje 0.1.14

```

1  #include <stdio.h>
    #include <stdlib.h>
3
    #define MAX 50
5
    void učitavanje(int mat[][MAX], int* n)
7  {
    int i, j;
9
    printf("Uneti dimenzije matrice: ");
11    scanf("%d", n);

13    if (*n <= 0 || *n > MAX)
    {
15        printf("Neispravna dimenzija matrice\n");
        exit(EXIT_FAILURE);
    }

```

```

17     }

19     printf("Uneti matricu celih brojeva\n");

21     for(i=0; i<*n; i++)
22         for(j=0; j<*n; j++)
23             scanf("%d", &mat[i][j]);
24 }

25 int sortirana_kolona(int mat[][MAX], int n, int j)
26 {
27     int i;

29     for(i=0; i<n-1; i++)
30         if (mat[i][j] >= mat[i+1][j])
31             return 0;

33     return 1;
34 }

35

37 int sortirani_po_kolonama(int mat[][MAX], int n)
38 {
39     int j;

41     for(j=0; j<n; j++)
42         if (!sortirana_kolona(mat, n, j))
43             return 0;

45     return 1;
46 }

47

49 int sortirana_vrsta(int mat[][MAX], int n, int i)
50 {
51     int j;

53     for(j=0; j<n-1; j++)
54         if (mat[i][j] >= mat[i][j+1])
55             return 0;

57     return 1;
58 }

59 int sortirani_po_vrstama(int mat[][MAX], int n)
60 {
61     int i;

63     for(i=0; i<n; i++)
64         if (!sortirana_vrsta(mat, n, i))
65             return 0;

67     return 1;
68 }

```

```

69  int sortirana_glavna(int mat[][MAX], int n)
71  {
73      int i;
75      for(i=0; i<n-1; i++)
76          if (mat[i][i] >= mat[i+1][i+1])
77              return 0;
78
79      return 1;
80  }
81
82  int sortirana_sporedna(int mat[][MAX], int n)
83  {
84      int i;
85      for(i=0; i<n-1; i++)
86          if (mat[i][n-i-1] >= mat[i+1][n-i-2])
87              return 0;
88
89      return 1;
90  }
91
92  int sortirani_po_dijagonalama(int mat[][MAX], int n)
93  {
94      if (!sortirana_glavna(mat, n))
95          return 0;
96
97      if (!sortirana_sporedna(mat, n))
98          return 0;
99
100     return 1;
101 }
102
103 int main()
104 {
105     int mat[MAX][MAX];
106     int n;
107
108     učitavanje(mat, &n);
109
110     if (sortirani_po_kolonama(mat, n))
111         printf("Elementi su sortirani po kolonama.\n");
112     else
113         printf("Elementi nisu sortirani po kolonama.\n");
114
115     if (sortirani_po_vrstama(mat, n))
116         printf("Elementi su sortirani po vrstama.\n");
117     else
118         printf("Elementi nisu sortirani po vrstama.\n");
119
120     if (sortirani_po_dijagonalama(mat, n))

```

```

121     printf("Elementi su sortirani po dijagonalama.\n");
    else
123     printf("Elementi nisu sortirani po dijagonalama.\n");

125     return 0;
}

```

## Rešenje 0.1.15

```

1  #include <stdio.h>
   #include <stdlib.h>
3
   #define MAX 10
5
   void ucitavanje(int mat[][MAX], int* n)
7  {
   int i, j;
9
   printf("Uneti dimenzije matrice: ");
11  scanf("%d", n);
13
   if (*n <= 0 || *n > MAX)
   {
15     printf("Neispravna dimenzija matrice\n");
     exit(EXIT_FAILURE);
17  }
19
   printf("Uneti matricu celih brojeva\n");
21
   for(i=0; i<*n; i++)
       for(j=0; j<*n; j++)
23         scanf("%d", &mat[i][j]);
   }
25
   int suma_kolone(int mat[][MAX], int n, int j)
27  {
   int suma = 0, i;
29
   for(i=0; i<n; i++)
31     suma += mat[i][j];
33
   return suma;
   }
35
   int uredjene_sume(int mat[][MAX], int n)
37  {
   int suma1, suma2;
39   int j;
41
   suma1 = suma_kolone(mat, n, 0);

```

```

43     for(j=1; j<n; j++)
44     {
45         suma2 = suma_kolone(mat, n, j);

47         if (suma1 >= suma2)
48             return 0;

49         suma1 = suma2;
51     }

53     return 1;
54 }

55
56 int main()
57 {
58     int mat[MAX][MAX];
59     int n;

61     učitavanje(mat, &n);

63     if (uredjene_sume(mat, n))
64         printf("Sume jesu uredjenje strogo rastuce.\n");
65     else
66         printf("Sume nisu uredjenje strogo rastuce.\n");
67
69     return 0;
70 }

```

## Rešenje 0.1.16

```

1  #include <stdio.h>
2  #include <stdlib.h>
3
4  #define MAX 200
5
6  void učitavanje(int mat[][MAX], int* n)
7  {
8      int i, j;
9
10     printf("Uneti dimenzije matrice: ");
11     scanf("%d", n);
12
13     if (*n <= 0 || *n > MAX)
14     {
15         printf("Neispravna dimenzija matrice\n");
16         exit(EXIT_FAILURE);
17     }
18
19     printf("Uneti matricu celih brojeva\n");

```

```

21     for(i=0; i<*n; i++)
22         for(j=0; j<*n; j++)
23             scanf("%d", &mat[i][j]);
24 }
25
26 int skalarni_proizvod(int mat[][MAX], int n, int i, int j)
27 {
28     int suma = 0, k;
29
30     for(k=0; k<n; k++)
31         suma += mat[i][k] * mat[j][k];
32
33     return suma;
34 }
35
36 int ortonormirana(int mat[][MAX], int n)
37 {
38     int i, j;
39
40     for(i=0; i<n; i++)
41         for(j=0; j<n; j++)
42             if (i==j && skalarni_proizvod(mat, n, i, i) != 1)
43                 return 0;
44             else if (i != j && skalarni_proizvod(mat, n, i, j) != 0)
45                 return 0;
46
47     return 1;
48 }
49
50 int main()
51 {
52     int mat[MAX][MAX];
53     int n;
54
55     učitavanje(mat, &n);
56
57     if (ortonormirana(mat, n))
58         printf("Matrica jeste ortonormirana.\n");
59     else
60         printf("Matrica nije ortonormirana.\n");
61
62     return 0;
63 }

```

## Rešenje 0.1.17

```

1  #include <stdio.h>
2  #include <stdlib.h>
3
4  #define MAX 50
5

```

```

7 void učitavanje(int mat[][MAX], int* n)
8 {
9     int i, j;
10
11     printf("Uneti dimenzije matrice: ");
12     scanf("%d", n);
13
14     if (*n <= 0 || *n > MAX)
15     {
16         printf("Neispravna dimenzija matrice\n");
17         exit(EXIT_FAILURE);
18     }
19
20     printf("Uneti matricu celih brojeva\n");
21
22     for(i=0; i<*n; i++)
23         for(j=0; j<*n; j++)
24             scanf("%d", &mat[i][j]);
25 }
26
27 int suma_kolone(int mat[][MAX], int n, int j)
28 {
29     int i, suma = 0;
30
31     for(i=0; i<n; i++)
32         suma += mat[i][j];
33
34     return suma;
35 }
36
37 int suma_vrste(int mat[][MAX], int n, int i)
38 {
39     int j, suma = 0;
40
41     for(j=0; j<n; j++)
42         suma += mat[i][j];
43
44     return suma;
45 }
46
47 int magicni_kvadrat(int mat[][MAX], int n)
48 {
49     int suma = suma_kolone(mat, n, 0);
50     int i, j;
51
52     for(j=1; j<n; j++)
53         if (suma_kolone(mat, n, j) != suma)
54             return 0;
55
56     for(i=0; i<n; i++)
57         if (suma_vrste(mat, n, i) != suma)
58             return 0;

```



```

59     return 1;
60 }
61
62 int main()
63 {
64     int mat[MAX][MAX];
65     int n;
66
67     učitavanje(mat, &n);
68
69     if (magicni_kvadrat(mat, n))
70         printf("Matrica jeste magicni kvadrat.\n");
71     else
72         printf("Matrica nije magicni kvadrat.\n");
73
74     return 0;
75 }

```

## Rešenje 0.1.18

```

1  #include <stdio.h>
2  #include <stdlib.h>
3
4  #define MAX 100
5
6  void učitavanje(int mat[][MAX], int* n)
7  {
8      int i, j;
9
10     printf("Uneti dimenzije matrice: ");
11     scanf("%d", n);
12
13     if (*n <= 0 || *n > MAX)
14     {
15         printf("Neispravna dimenzija matrice\n");
16         exit(EXIT_FAILURE);
17     }
18
19     printf("Uneti matricu celih brojeva\n");
20
21     for(i=0; i<*n; i++)
22         for(j=0; j<*n; j++)
23             scanf("%d", &mat[i][j]);
24 }
25
26 int main()
27 {
28     int mat[MAX][MAX];
29     int n;
30     int i, j, k;

```

```

32   ucitavanje(mat, &n);
34   for(k=0; k<n; k++)
35   {
36       j = k;
37       i = 0;
38
39       while(j >= 0)
40       {
41           printf("%d ", mat[i][j]);
42           i++;
43           j--;
44       }
45
46       printf("\n");
47   }
48
49   for(k=1; k<n; k++)
50   {
51       i = k;
52       j = n-1;
53
54       while(i < n)
55       {
56           printf("%d ", mat[i][j]);
57           i++;
58           j--;
59       }
60
61       printf("\n");
62   }
63
64   return 0;
65 }

```

### Rešenje 0.1.19

```

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

#define MAX 50

void ucitavanje(int mat[][MAX], int* m, int* n)
{
    int i, j;

    printf("Uneti dimenzije matrice: ");
    scanf("%d%d", m, n);
}

```

```

14     if (*n <= 0 || *n > MAX || *m <= 0 || *m > MAX)
15     {
16         printf("Neispravna dimenzija matrice\n");
17         exit(EXIT_FAILURE);
18     }
19
20     printf("Uneti matricu celih brojeva\n");
21
22     for(i=0; i<*m; i++)
23         for(j=0; j<*n; j++)
24             scanf("%d", &mat[i][j]);
25
26 void ispis(int mat[][MAX], int m, int n)
27 {
28     int i, j;
29
30     for(i=0; i<m; i++)
31     {
32         for(j=0; j<n; j++)
33             printf("%d ", mat[i][j]);
34         printf("\n");
35     }
36 }
37
38 void mnozenje(int a[][MAX], int m, int n, int b[][MAX], int k, int t,
39               int c[][MAX])
40 {
41     int i, j, w;
42
43     for(i=0; i<m; i++)
44         for(j=0; j<t; j++)
45         {
46             c[i][j] = 0;
47             for(w=0; w<n; w++)
48                 c[i][j] += a[i][w] * b[w][j];
49         }
50 }
51
52 int main()
53 {
54     int a[MAX][MAX], b[MAX][MAX], c[MAX][MAX];
55     int m, n;
56     int k, t;
57
58     učitavanje(a, &m, &n);
59     učitavanje(b, &k, &t);
60
61     if (n != k)
62     {
63         printf("Mnozenje matrica nije moguće.\n");
64         return -1;

```

```

64     }

66     mnozenje(a, m, n, b, k, t, c);

68     printf("Rezultat mnozenja je:\n");
    ispis(c, m, t);

70     return 0;

72 }

```

## Rešenje 0.1.20

```

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

4 #define MAX 50

6 void učitavanje(double mat[][MAX], int* m, int* n)
{
8     int i, j;

10     printf("Uneti dimenzije matrice: ");
    scanf("%d%d", m, n);

12     if (*n <= 0 || *n > MAX || *m <= 0 || *m > MAX)
14     {
        printf("Neispravna dimenzija matrice\n");
        exit(EXIT_FAILURE);
    }

18     printf("Uneti matricu celih brojeva\n");

20     for(i=0; i<*m; i++)
22         for(j=0; j<*n; j++)
            scanf("%lf", &mat[i][j]);
24 }

26 int main()
{
28     double mat[MAX][MAX];
    int m, n, k, i, j;

30     int indeks_kolone;
    double max_kolone, min_vrste;

34     učitavanje(mat, &m, &n);

36     for(i=0; i<m; i++)
    {
38         min_vrste = mat[i][0];

```

```

    indeks_kolone = 0;

40     for(j=1; j<n; j++)
42         if (mat[i][j] < min_vrstje)
44             {
46                 min_vrstje = mat[i][j];
48                 indeks_kolone = j;
49             }

50     max_kolone = mat[0][indeks_kolone];

51     for(k=1; k<m; k++)
52         if (mat[k][indeks_kolone] > max_kolone)
53             max_kolone = mat[k][indeks_kolone];

54     if (min_vrstje == max_kolone)
55         printf("Sedlo: %d %d %g\n", i, indeks_kolone, min_vrstje);
56
57     }
58
59     return 0;
60 }

```

## Rešenje 0.1.21

```

1  #include <stdio.h>
2  #include <stdlib.h>
3
4  #define MAX 50
5
6  void učitavanje(int mat[][MAX], int* m, int* n)
7  {
8      int i, j;
9
10     printf("Uneti dimenzije matrice: ");
11     scanf("%d%d", m, n);
12
13     if (*n <= 0 || *n > MAX || *m <= 0 || *m > MAX)
14     {
15         printf("Neispravna dimenzija matrice\n");
16         exit(EXIT_FAILURE);
17     }
18
19     printf("Uneti matricu celih brojeva\n");
20
21     for(i=0; i<*m; i++)
22         for(j=0; j<*n; j++)
23             scanf("%d", &mat[i][j]);
24 }
25
26 int main()

```

```

27 {
28     int mat[MAX][MAX];
29     int m, n, brojac, i, j;

30
31     int pravac = 1;
32     int gore_i, dole_i, levo_j, desno_j;

33     učitavanje(mat, &m, &n);

34
35     gore_i = 1;
36     dole_i = m-1;

37
38     levo_j = 0;
39     desno_j = n-1;

40
41     i = 0;
42     j = 0;

43
44     for(brojac=0; brojac < m*n; brojac++)
45     {
46         printf("%d ", mat[i][j]);

47
48         switch(pravac)
49         {
50             case 1:
51                 if (j == desno_j)
52                 {
53                     pravac = 2;
54                     desno_j--;
55                     i++;
56                 }
57                 else
58                     j++;
59                 break;
60             case 2:
61                 if (i == dole_i)
62                 {
63                     pravac = 3;
64                     dole_i--;
65                     j--;
66                 }
67                 else
68                     i++;
69                 break;
70             case 3:
71                 if (j == levo_j)
72                 {
73                     pravac = 4;
74                     levo_j++;
75                     i--;
76                 }
77                 else

```

```

79         j--;
           break;
81     case 4:
           if (i == gore_i)
83         {
               pravac = 1;
85               gore_i++;
               j++;
87         }
           else
89             i--;
       }
91     }

93     return 0;
}

```

## Rešenje 0.1.22

```

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

#define MAX 50

void ucitavanje(int mat[][MAX], int* m, int* n)
{
    int i, j;

    printf("Uneti dimenzije matrice: ");
    scanf("%d%d", m, n);

    if (*n <= 0 || *n > MAX || *m <= 0 || *m > MAX)
    {
        printf("Neispravna dimenzija matrice\n");
        exit(EXIT_FAILURE);
    }

    printf("Uneti matricu celih brojeva\n");

    for(i=0; i<*m; i++)
        for(j=0; j<*n; j++)
            scanf("%d", &mat[i][j]);
}

int podmatrica(int a[][MAX], int m, int n, int b[][MAX], int k, int t)
{
    int i, j, x, y;
    int jeste_pod;

    for(i=0; i<= m-k; i++)

```

```

32     for(j=0; j<= n-t; j++)
33     {
34         jeste_pod = 1;
35         for(x=0; x<k && jeste_pod; x++)
36             for(y=0; y<t && jeste_pod; y++)
37                 if (a[i+x][j+y] != b[x][y])
38                     jeste_pod = 0;
39
40         if (jeste_pod)
41             return 1;
42     }
43
44     return 0;
45 }
46
47 int main()
48 {
49     int a[MAX][MAX], b[MAX][MAX];
50     int m, n;
51     int k, t;
52
53     ucitavanje(a, &m, &n);
54     ucitavanje(b, &k, &t);
55
56     if (podmatrica(a, m, n, b, k, t))
57         printf("Druga matrica je sadrzana u prvoj matrici.\n");
58     else
59         printf("Druga matrica nije sadrzana u prvoj matrici.\n");
60 }

```