

PROGRAMIRANJE 1

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

PROGRAMIRANJE 1

Zbirka zadataka

**Beograd
2016.**

Autori:

dr Milena Vujošević Jančić, docent na Matematičkom fakultetu u Beogradu

dr Jovana Kovačević, docent na Matematičkom fakultetu u Beogradu

Danijela Simić, asistent na Matematičkom fakultetu u Beogradu

Anđelka Zečević, asistent na Matematičkom fakultetu u Beogradu

PROGRAMIRANJE 1

Zbirka zadataka

Sadržaj

0.1	Višedimenzioni nizovi	v
0.2	Rešenja	xix

0.1 Višedimenzioni nizovi

Zadatak 0.1.1 Napisati program koji učitava i zatim ispisuje vrednosti učitane matrice. Pretpostaviti da je maksimalna dimenzija matrice 50×50 , a da se sa ulaza najpre učitavaju dva cela broja m i n , a potom i elementi matrice celih brojeva dimenzije $m \times n$. 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 Napisati program koji za učitaneu celobrojnu matricu dimenzije $m \times n$ izračunava njenu Euklidsku normu. Pretpostaviti da je maksimalna dimenzija matrice 50×50 . U slučaju greške ispisati odovarajuću poruku. UPUTSTVO: *Euklidska norma matrice je kvadratni koren sume kvadrata svih elemenata matrice.*

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 ispisuju vrednosti matrice mat dimenzije $m \times n$.

Napisati program koji najpre učitava, a zatim i ispisuje vrednosti učitane matrice. Pretpostaviti da je maksimalna dimenzija matrice 50×50 . 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 za učitane matricu celih brojeva¹ ispisuje odgovarajuću transponovanu matricu. Pretpostaviti da je maksimalna dimenzija matrice 50×50 . U slučaju greške ispisati odgovarajuću 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
```

¹Pod pojmom *učitati matricu* ili za *datu 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 $m \times n$. Napisati program koji za učitane matricu celih brojeva, i dva cela broja k i t ispisuje matricu dobijenu razmenjivanjem k -te i t -te vrste ulazne matrice. Pretpostaviti da je maksimalna dimenzija matrice 50×50 . 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 indkxi vrsta.
```

[Rešenje 0.1.5]

Zadatak 0.1.6 Napisati program koji za učitane matricu celih brojeva ispisuje indkse onih elemenata matrice koji su jednaki zbiru svih svojih susednih


```

- - - - - s b s -
- s s s - - s s s -
- s a s - - - - -
- s s s - - - - -
- - - - - - - s s
- - - - - - - s c

```

Slika 1: *Susedni elementi u matrici.*

elemenata. Pretpostaviti da je maksimalna dimenzija matrice 50×50 . U slučaju greške ispisati odovarajuću poruku. UPUTSTVO: *Broj susednih elemenata matrice zavisi od položaja elementa u matrici. Na slici 1 su slovom s obeleženi susedni elementi matrice za elemente a, b i c.*

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 element niza b_i izračunava kao srednju vrednost elemenata i -te vrste matrice. Napisati program koji za učitane matricu celih brojeva ispisuje dobijeni niz. Pretpostaviti da je maksimalna dimenzija matrice 50×50 . 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. Napisati funkcije:

- (a) `int refleksivna(int a[][MAX], int n)` kojom se za relaciju zadatom matricom a dimenzije $n \times n$ ispituje da li je refleksivna;
- (b) `int simetricna(int a[][MAX], int n)` kojom se za relaciju zadatom matricom a dimenzije $n \times n$ ispituje da li je simetrična;
- (c) `int tranzitivna(int a[][MAX], int n)` kojom se za relaciju zadatom matricom a ispituje dimenzije $n \times n$ da li je tranzitivna;
- (d) `int ekvivalencija(int a[][MAX], int n)` kojom se za relaciju koja je zadata matricom a dimenzije $n \times n$ ispituje da li je relacija ekvivalencije.

Napisati program koji za učitano dimenziju n i kvadratnu matricu dimenzije $n \times n$ ispituje osobine odgovarajuće relacije. Pretpostaviti da je maksimalna dimenzija matrice 50×50 i da matrica može imati za vrednosti elemenata samo brojeve 0 i 1. U slučaju greške ispisati odgovarajuću 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 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 50×50 . 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 100×100 . 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 učitanu kvadratnu matricu ispisuje redni broj kolone koja ima najveći zbir elemenata. Pretpostaviti da je maksimalna dimenzija matrice 50×50 . 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 učitanu kvadratnu 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 50×50 . 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 50×50 . 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 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 za učitano kvadratnu matricu ispituje da li su njeni elementi po kolonama, vrstama i dijagonalama (glavnoj i sporednoj) sortirani strogo rastuće. Pretpostaviti da je maksimalna dimenzija matrice 50×50 . 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 za učitane kvadratnu matricu ispisuje da li su zbrovi elemenata njenih kolona uređeni u strogo rastućem poretku. Pretpostaviti da je maksimalna dimenzija matrice koja se unosi 10×10 . 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. Napisati program koji za unetu kvadratnu matricu proverava da li je ortonormirana. 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 Kvadratna matrica je *magični kvadrat* ako je suma elemenata u svakoj koloni i svakoj vrsti jednaka. Napisati program koji proverava da li je data kvadratna matrica magični kvadrat i ispisuje odgovarajuću poruku na

standardni izlaz. Pretpostaviti da je maksimalna dimenzija matrice je 50×50 . U slučaju greške ispisati odgovarajuću poruku.

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 kvadratne matrice celih brojeva u grupama koje su paralelne sa sporednom dijagonalom matrice počevši od gornjeg levog ugla matrice. Pretpostaviti da je maksimalna dimenzija matrice 100×100 . 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 50×50 .

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 50×50 . 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 spiralmom redosledu počevši od gornjeg levog ugla krećući se u smeru suprotnom od smera kazaljke na satu. Maksimalna dimenzija matrice je 50×50 . 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 50×50 .

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 sadrzana u prvoj matrici.
```

[Rešenje 0.1.22]

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++)
25     for(j=0; j<n; j++)
```

```

scanf("%d", &mat[i][j]);
27
/* Ispis elemenata matrice. */
29 for(i=0; i<m; i++)
{
31     for(j=0; j<n; j++)
        printf("%d ", mat[i][j]);
33     printf("\n");
}
35
37 return 0;
}

```

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");
        exit(EXIT_FAILURE);
21     }

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

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

29

31     for(i=0; i<m; i++)
        for(j=0; j<n; j++)
33         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>
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 int main()
39 {
40     int mat[MAX][MAX];
41     int m, n;
42
43     učitavanje(mat, &m, &n);
44     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 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 (*m <= 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 transponovana(int a[][MAX], int m, int n, int b[][MAX])
39 {
40     int i, j;
41
42     for(i=0; i<m; i++)
43         for(j=0; j<n; j++)
44             b[j][i] = a[i][j];
45 }
```

```

47 int main()
48 {
49     int mat[MAX][MAX], t[MAX][MAX];
50     int m, n;
51
52     učitavanje(mat, &m, &n);
53     transponovana(mat, m, n, t);
54     ispis(t, n, m);
55
56     return 0;
57 }

```

Rešenje 0.1.5

```

#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 (*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]);
}

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

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

```

```

38 void razmeni(int mat[][MAX], int m, int n, int k, int t)
{
40     int j, pom;

42     for(j = 0; j < n; j++)
    {
44         pom = mat[k][j];
        mat[k][j] = mat[t][j];
46         mat[t][j] = pom;
    }
48 }

50 int main()
{
52     int mat[MAX][MAX];
    int m, n;
54     int k, t;

56     učitavanje(mat, &m, &n);

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

60

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

66

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

70     ispis(mat, m, n);

72     return 0;
}

```

Rešenje 0.1.6

```

#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 (*m <= 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
27 int main()
28 {
29     int mat[MAX][MAX];
30     int m, n, i, j, suma;
31     int k, t;
32
33     učitavanje(mat, &m, &n);
34
35     printf("Indeksi elemenata koji su jednaki zbiru suseda su:\n");
36     for(i=0; i<m; i++)
37         for(j=0; j<n; j++)
38         {
39             suma = 0;
40
41             for(k=-1; k<=1; k++)
42                 for(t=-1; t<=1; t++)
43                     if (i+k >= 0 && i+k < n && j+t >= 0 && j+t < n)
44                         suma += mat[i+k][j+t];
45
46             if (suma - mat[i][j] == mat[i][j])
47                 printf("%d %d\n", i, j);
48         }
49
50     return 0;
51 }

```

Rešenje 0.1.7

```

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: ");

```

```

12     scanf("%d%d", m, n);
13
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 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>
  #include <stdlib.h>

```

```

3  #define MAX 50
5
6  void ucitavanje(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 reflektivna(int a[][MAX], int n)
27 {
28     int i;
29
30     for(i=0; i<n; i++)
31         if (a[i][i] != 1)
32             return 0;
33
34     return 1;
35 }
36
37 int simetricna(int a[][MAX], int n)
38 {
39     int i, j;
40
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
46     return 1;
47 }
48
49 int tranzitivna(int a[][MAX], int n)
50 {
51     int i, j, k;
52
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;
58
59     return 1;
60 }
61
62 int ekvivalencija(int a[][MAX], int n)
63 {
64     if (refleksivna(a, n) && simetricna(a, n) && tranzitivna(a, n))
65         return 1;
66
67     return 0;
68 }
69
70 int main()
71 {
72     int a[MAX][MAX];
73     int n;
74
75     učitavanje(a, &n);
76
77     if (refleksivna(a, n))
78         printf("Relacija jeste refleksivna.\n");
79     else
80         printf("Relacija nije refleksivna.\n");
81
82     if (simetricna(a, n))
83         printf("Relacija jeste simetricna.\n");
84     else
85         printf("Relacija nije simetricna.\n");
86
87     if (tranzitivna(a, n))
88         printf("Relacija jeste tranzitivna.\n");
89     else
90         printf("Relacija nije tranzitivna.\n");
91
92     if (ekvivalencija(a, n))
93         printf("Relacija jeste ekvivalencija.\n");
94     else
95         printf("Relacija nije ekvivalencija.\n");
96
97     return 0;
98 }

```

Rešenje 0.1.9

```

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

```

```
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");
20
21    for(i=0; i<*n; i++)
22        for(j=0; j<*n; j++)
23            scanf("%f", &mat[i][j]);
24 }
25
26 float trag(float a[][MAX], int n)
27 {
28     float suma = 0;
29     int i;
30
31     for(i=0; i<n; i++)
32         suma += a[i][i];
33
34     return suma;
35 }
36
37 float suma_sporodna(float a[][MAX], int n)
38 {
39     float suma = 0;
40     int i;
41
42     for(i=0; i<n; i++)
43         suma += a[i][n-i-1];
44
45     return suma;
46 }
47
48 float suma_iznad(float a[][MAX], int n)
49 {
50     float suma = 0;
51     int i, j;
52
53     for(i=0; i<n; i++)
54         for(j=i+1; j<n; j++)
55             suma += a[i][j];
56 }
```

```

    return suma;
58 }

60 float suma_ispod(float a[][MAX], int n)
61 {
62     float suma = 0;
63     int i, j;
64
65     for(i=0; i<n; i++)
66         for(j=n-i-1; j>i; j--)
67             suma += a[i][j];
68
69     return suma;
70 }

72 int main()
73 {
74     float a[MAX][MAX];
75     int n;
76
77     učitavanje(a, &n);
78
79     printf("Trag je %.3f.\n", trag(a, n));
80     printf("Suma na sporednoj dijagonali je %.3f.\n", suma_sporedna(a,
        n));
81     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
        ));
83
84     return 0;
85 }

```

Rešenje 0.1.10

```

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

4 #define MAX 50

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

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

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

```

```

    }

18     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 donje_trougaona(int a[][MAX], int n)
27 {
28     int i, j;

29
30     for(i=0; i<n; i++)
31         for(j=i+1; j<n; j++)
32             if (a[i][j] != 0)
33                 return 0;
34
35     return 1;
36 }

37
38 int main()
39 {
40     int a[MAX][MAX];
41     int n;

42
43     učitavanje(a, &n);

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

49
50     return 0;
51 }

```

Rešenje 0.1.11

```

#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
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 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
37     max_zbir = trenutni_zbir;
38
39     for(j=1; j<n; j++)
40     {
41
42         trenutni_zbir = 0;
43         for(i=0; i<n; i++)
44             trenutni_zbir += a[i][j];
45
46         if (trenutni_zbir > max_zbir)
47         {
48             max_zbir = trenutni_zbir;
49             indeks_kolone = j;
50         }
51     }
52
53     printf("Indeks kolone je: %d\n", indeks_kolone);
54
55     return 0;
56 }

```

Rešenje 0.1.12

```

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

```



```

7 void učitavanje(float 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("%f", &mat[i][j]);
25 }
26
27 int main()
28 {
29     float a[MAX][MAX];
30     int n, i, j;
31     float gornji_trougao = 0, donji_trougao = 0;
32
33     učitavanje(a, &n);
34
35     for(i=0; i<n/2; i++)
36         for(j=i+1; j<n-i-1; j++)
37             gornji_trougao += a[i][j];
38
39     for(i=n/2; i<n; i++)
40         for(j=n-i; j<i; j++)
41             donji_trougao += a[i][j];
42
43     printf("%f %f\n", gornji_trougao, donji_trougao);
44
45     printf("Razlika je: %.2f\n", gornji_trougao - donji_trougao);
46
47     return 0;
48 }

```

Rešenje 0.1.13

```

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

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

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

20

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

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

46                 printf("%d ", suma);
48             }

50     printf("\n");

52     return 0;
   }

```

Rešenje 0.1.14

```

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

```

```

3  #define MAX 50
5
6  void ucitavanje(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 sortirana_kolona(int mat[][MAX], int n, int j)
27 {
28     int i;
29
30     for(i=0; i<n-1; i++)
31         if (mat[i][j] >= mat[i+1][j])
32             return 0;
33
34     return 1;
35 }
36
37 int sortirani_po_kolonama(int mat[][MAX], int n)
38 {
39     int j;
40
41     for(j=0; j<n; j++)
42         if (!sortirana_kolona(mat, n, j))
43             return 0;
44
45     return 1;
46 }
47
48 int sortirana_vrsta(int mat[][MAX], int n, int i)
49 {
50     int j;
51
52     for(j=0; j<n-1; j++)
53         if (mat[i][j] >= mat[i][j+1])
54             return 0;

```

```

55     return 1;
57 }

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)
70 {
71     int i;

73     for(i=0; i<n-1; i++)
74         if (mat[i][i] >= mat[i+1][i+1])
75             return 0;

77     return 1;
78 }

79 int sortirana_sporedna(int mat[][MAX], int n)
80 {
81     int i;

83     for(i=0; i<n-1; i++)
84         if (mat[i][n-i-1] >= mat[i+1][n-i-2])
85             return 0;

87     return 1;
88 }

89 int sortirani_po_dijagonalama(int mat[][MAX], int n)
90 {
91     if (!sortirana_glavna(mat, n))
92         return 0;

94     if (!sortirana_sporedna(mat, n))
95         return 0;

97     return 1;
98 }

100 }

101

102 int main()
103 {
104     int mat[MAX][MAX];
105     int n;

```

```

107     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");
122     else
123         printf("Elementi nisu sortirani po dijagonalama.\n");
124
125     return 0;
126 }

```

Rešenje 0.1.15

```

1  #include <stdio.h>
2  #include <stdlib.h>
3
4  #define MAX 10
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 suma_kolone(int mat[][MAX], int n, int j)
27 {
28     int suma = 0, i;

```

```

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

33     return suma;
34 }

35 int uredjene_sume(int mat[][MAX], int n)
36 {
37     int suma1, suma2;
38     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 int main()
56 {
57     int mat[MAX][MAX];
58     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 {
    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 skalarni_proizvod(int mat[][MAX], int n, int i, int j)
27 {
        int suma = 0, k;
29
        for(k=0; k<n; k++)
31            suma += mat[i][k] * mat[j][k];
33
        return suma;
    }
35
    int ortonormirana(int mat[][MAX], int n)
37 {
        int i, j;
39
        for(i=0; i<n; i++)
            for(j=0; j<n; j++)
41                if (i==j && skalarni_proizvod(mat, n, i, i) != 1)
43                    return 0;
                else if (i != j && skalarni_proizvod(mat, n, i, j) != 0)
45                    return 0;
47
        return 1;
    }
49
    int main()
51 {
        int mat[MAX][MAX];
53        int n;
55
        ucitavanje(mat, &n);
57
        if (ortonormirana(mat, n))
            printf("Matrica jeste ortonormirana.\n");

```

```

59     else
        printf("Matrica nije ortonormirana.\n");
61
        return 0;
63 }

```

Rešenje 0.1.17

```

1  #include <stdio.h>
   #include <stdlib.h>
3
   #define MAX 50
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 i, suma = 0;
29
       for(i=0; i<n; i++)
           suma += mat[i][j];
31
       return suma;
33   }
35
   int suma_vrste(int mat[][MAX], int n, int i)
37   {
       int j, suma = 0;
39
       for(j=0; j<n; j++)
           suma += mat[i][j];
41
       return suma;
43

```



```

}
45
int magicni_kvadrat(int mat[][MAX], int n)
47
{
    int suma = suma_kolone(mat, n, 0);
49
    int i, j;

    for(j=1; j<n; j++)
        if (suma_kolone(mat, n, j) != suma)
53
            return 0;

    for(i=0; i<n; i++)
        if (suma_vrste(mat, n, i) != suma)
57
            return 0;

    return 1;
59
}
61
int main()
63
{
    int mat[MAX][MAX];
65
    int n;

    ucitavanje(mat, &n);
67

    if (magicni_kvadrat(mat, n))
        printf("Matrica jeste magicni kvadrat.\n");
69
    else
        printf("Matrica nije magicni kvadrat.\n");
71
    return 0;
73
}
75

```

Rešenje 0.1.18

```

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

4
#define MAX 100

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

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

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

```

```
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 main()
28 {
29     int mat[MAX][MAX];
30     int n;
31     int i, j, k;
32
33     učitavanje(mat, &n);
34
35     for(k=0; k<n; k++)
36     {
37         j = k;
38         i = 0;
39
40         while(j >= 0)
41         {
42             printf("%d ", mat[i][j]);
43             i++;
44             j--;
45         }
46
47         printf("\n");
48     }
49
50     for(k=1; k<n; k++)
51     {
52         i = k;
53         j = n-1;
54
55         while(i < n)
56         {
57             printf("%d ", mat[i][j]);
58             i++;
59             j--;
60         }
61
62         printf("\n");
63     }
64
65     return 0;
66 }
```

Rešenje 0.1.19

```
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 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 }
```

```

50 int main()
51 {
52     int a[MAX][MAX], b[MAX][MAX], c[MAX][MAX];
53     int m, n;
54     int k, t;
55
56     ucitavanje(a, &m, &n);
57     ucitavanje(b, &k, &t);
58
59     if (n != k)
60     {
61         printf("Mnozenje matrica nije moguće.\n");
62         return -1;
63     }
64
65     mnozenje(a, m, n, b, k, t, c);
66
67     printf("Rezultat mnozenja je:\n");
68     ispis(c, m, t);
69
70     return 0;
71
72 }

```

Rešenje 0.1.20

```

1 #include <stdio.h>
2 #include <stdlib.h>
3
4 #define MAX 50
5
6 void ucitavanje(double 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("%lf", &mat[i][j]);
24 }

```

```

26 int main()
27 {
28     double mat[MAX][MAX];
29     int m, n, k, i, j;
30
31     int indeks_kolone;
32     double max_kolone, min_vrstec;
33
34     učitavanje(mat, &m, &n);
35
36     for(i=0; i<m; i++)
37     {
38         min_vrstec = mat[i][0];
39         indeks_kolone = 0;
40
41         for(j=1; j<n; j++)
42             if (mat[i][j] < min_vrstec)
43             {
44                 min_vrstec = mat[i][j];
45                 indeks_kolone = j;
46             }
47
48         max_kolone = mat[0][indeks_kolone];
49
50         for(k=1; k<m; k++)
51             if (mat[k][indeks_kolone] > max_kolone)
52                 max_kolone = mat[k][indeks_kolone];
53
54         if (min_vrstec == max_kolone)
55             printf("Sedlo: %d %d %g\n", i, indeks_kolone, min_vrstec);
56     }
57
58     return 0;
59 }

```

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);

```

```
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 prava = 1;
32      int gore_i, dole_i, levo_j, desno_j;
33
34      ucitavanje(mat, &m, &n);
35
36      gore_i = 1;
37      dole_i = m-1;
38
39      levo_j = 0;
40      desno_j = n-1;
41
42      i = 0;
43      j = 0;
44
45      for(brojac=0; brojac < m*n; brojac++)
46      {
47          printf("%d ", mat[i][j]);
48
49          switch(prava)
50          {
51              case 1:
52                  if (j == desno_j)
53                  {
54                      prava = 2;
55                      desno_j--;
56                      i++;
57                  }
58                  else
59                      j++;
60                  break;
61              case 2:
62                  if (i == dole_i)
63                  {
64                      prava = 3;
```

```

65         dole_i--;
66         j--;
67     }
68     else
69         i++;
70     break;
71 case 3:
72     if (j == levo_j)
73     {
74         pravic = 4;
75         levo_j++;
76         i--;
77     }
78     else
79         j--;
80     break;
81 case 4:
82     if (i == gore_i)
83     {
84         pravic = 1;
85         gore_i++;
86         j++;
87     }
88     else
89         i--;
90 }
91 }
92
93 return 0;
94 }

```

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);
    }
}

```

```

20     printf("Uneti matricu celih brojeva\n");
22     for(i=0; i<*m; i++)
23         for(j=0; j<*n; j++)
24             scanf("%d", &mat[i][j]);
26 int podmatrica(int a[][MAX], int m, int n, int b[][MAX], int k, int t
    )
27 {
28     int i, j, x, y;
29     int jeste_pod;
30
31     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 }

```