

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	xviii

0.1 Višedimenzioni nizovi

Zadatak 0.1.1 Napisati funkcije za rad sa celobrojnim matricama:

- (a) `void ucitaj(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 50×50 . U slučaju greške ispisati odovarajuću poruku.

Milena: 1) Ako je ucitaj onda je ispisi, ili ako je ispis onda je ucitavanje — ovako se ne slaze

Milena: 2) Meni nema logike da vec prvi zadatak sa matricom obuhvata funkcije. Delujem mi metodoloski bolje najpre ucitati i ispisati matricu iz main funkcije, i nesto sa njom uraditi, pa onda uraditi isto to sa funkcijama. Ovako je od jednom pretesko.

Milena: 3) Deluje mi malo dosadno da svaki put u tekstu zadatka navodimo kako se ucitava matrica, napre dimenzije pa njeni elementi. Krenula sam to da ispravljam ali nisam sigurna da sam u pravu. Deluje mi da je dovoljno na jednom mestu objasniti kako se ucitava matrica, a onda posle samo koristiti termin "ucitati matricu". Mozda negde, npr nakon prvog takvog zadatka u fus notu

moze da se stavi ta napomena: tipa "Pod pojmom učitavanja matrice uvek se podrazumeva da se najpre unosi dimenzija pa zatim elementi matrice"

Milena: 4) Pretpostavku o dimenziji matrice i ispisu greske sam svuda stavila na kraj, da budemo dosledni sa formulacijama.

Milena: 5) Generalno sam protiv formulacija u kojima se navodi KAKO se nesto radi tj u kojima se daju vremenske odrednice tipa, program prvo ucita ta i ta dva broja, pa zatim uradi to i to... To je kao da im diktiramo resenje zadatka. Mislim da formulacija ne treba da sadrzi vremenske odrednice niti sugestije na tu temu, vec opis sta program treba da uradi. Preformulisala sam neke stvari na taj nacin, a ti vidi kako ti se to svidja i razmisli za ostale. Generalno je to povezano sa učitavanjem matrice kada kazemo prvo učitaj dimenzije pa zatim vrednosti pa zatim dva broja pa onda izracunaj nesto pa onda ispiši to... Formulacija bi pre trebala da bude "Napisati program koji za ucitanu matricu izracunava nesto i ispisuje odgovarajuću poruku"...

Milena: 6) Mozda suma_sporedna i suma_ispod i suma_iznad ili tako nekako, tek da ta imena budu smisljena, sumd i sumD mi nisu dovoljno lepa imena.

Milena: 7) Zadatak 12, umesto 0 i 1 ispisivati odgovarajuće poruke - to ce pojednostaviti i skratiti i formulaciju zadatka.

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.
```

Zadatak 0.1.2 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 i ispisuje odgovarajuću transponovanu matricu. 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
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
```

Primer 3

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

Zadatak 0.1.3 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 $m \times n$. Program učitava dimenzije matrice, a potom i elemente matrice celih brojeva, a potom i dva cela broja k i t . Ispisati matricu dobijenu razmenjivanjem k -te i t -te vrste. 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 indekse 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 indekse 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 indekse vrsta: -1 50
Neispravni indeksi vrsta.
```

Zadatak 0.1.4 Napisati program koji učitava dimenzije matrice, dva prirodna broja n i m , a zatim redom i elemente matrice celih brojeva. Ispisati indekse 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 50×50 . 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
```

Zadatak 0.1.5 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 učitava dimenzije matrice, dva prirodna broja n i m , a zatim redom i elemente matrice celih brojeva. Pozvati funkciju i ispisati 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
```

Zadatak 0.1.6 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 reflektivna(int a[][MAX], int n)` kojom se za relaciju zadatom matricom `a` dimenzije `n` ispituje da li je reflektivna.
- (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 50×50 . 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 reflektivna.
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 reflektivna.
Relacija jeste simetricna.
Relacija nije tranzitivna.
Relacija nije ekvivalencija.
```

Zadatak 0.1.7 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 sum_sporedna(float a[][MAX], int n)` koja računa zbir elemenata na sporednoj dijagonali matrice.
- (c) Napisati funkciju `float sumD(float a[][MAX], int n)` koja određuje sumu elemenata iznad glavne dijagonale.
- (d) Napisati funkciju `float sumd(float a[][MAX], int n)` koja određuje sumu elemenata ispod sporedne dijagonale matrice.

Napisati program koji testira napisane funkcije. Program učitava dimenziju n kvadratne matrice, a potom učitava i elemente matrice realnih brojeva. Ispisati 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.
```

Zadatak 0.1.8 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.
```

Zadatak 0.1.9 Napisati program koji za učitane 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
```

Zadatak 0.1.10 Program učitava ceo broj n , a zatim i elemente kvadratne matrice realnih brojeva dimenzije $n \times n$. Napisati program koji 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
```

Zadatak 0.1.11 Napisati program koji za učitane 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.

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
```

Zadatak 0.1.12 Napisati program koji učitava vrednosti matrice i ispituje da li su elementi matrice po kolonama, vrstama i dijagonalama (glavnoj i sporednoj) sortirani strogo rastuće. Ukoliko su svi elementi po kolonama sortirani rastuće ispisati 1, a ukoliko nisu 0. Ukoliko su svi elementi po vrstama sortirani rastuće ispisati 1, u suprotnom 0. Ukoliko su elementi na obe dijagonale sortirani rastuće ispisati 1, a u suprotnom 0. 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
Sortiranost po vrstama, kolonama
i dijagonalama je: 1 1 1
```

Primer 2

```
INTERAKCIJA SA PROGRAMOM:
Unesi dimenziju matrice: 2
Unesi elemente matrice:
6 9
4 10
Sortiranost po vrstama, kolonama
i dijagonalama je: 1 0 0
```

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
Sortiranost po vrstama, kolonama
i dijagonalama je: 0 1 1
```

Primer 4

```
INTERAKCIJA SA PROGRAMOM:
Unesi dimenziju matrice: 1
Unesi elemente matrice:
5
Sortiranost po vrstama, kolonama
i dijagonalama je: 1 1 1
```

Zadatak 0.1.13 Napisati program koji učitava vrednosti matrice i ispituje da li su zbirovi elemenata kolona matrice uredjeni u strogo rastućem poretku i ispisuje odgovarajuću poruku. Pretpostaviti da je maksimalna dimenzija matrice koja se unosi 10×10 . U slučaju greške ispisati odgovarajuću 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.
```

Zadatak 0.1.14 Matrica je *ortonormirana* ako je skalarni proizvod svakog para različitih vrsta jednak 0, a skalarni proizvod vrste sa samom sobom 1. Pro-

gram učitava ceo broj n ($0 < n \leq 200$), a potom i elemente kvadratne matrice celih brojeva dimenzije $n \times n$. Proveriti da li je uneta matrica ortonormirana i na standardni izlaz 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.
```

Zadatak 0.1.15 Datu kvadratna matrica je *magični kvadrat* ako je suma elemenata u svakoj koloni i svakoj vrsti jednaka. Napisati program koji učitava prirodni broj n i zatim elemente kvadratne matrice dimenzije $n \times n$, proverava da li je ona magični kvadrat i ispisuje odgovarajuću poruku na standardni izlaz. Maksimalna dimenzija matrice je 50×50 .

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.
```

* **Zadatak 0.1.16** Program učitava dimenziju kvadratne matrice n , a zatim elemente matrice celih brojeva. Ispisati elemente matrice u grupama koje su paralelne sa sporednom dijagonalom matrice počevši od gornjeg levog ugla matrice. Maksimalna dimenzija matrice je 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
```

* **Zadatak 0.1.17** 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 $k \times t$. Program učitava dva cela broja m i n , a potom i matricu celih brojeva dimenzije $m \times n$. Potom se učitavaju dva cela broja k i t i matrica celih brojeva dimenzije $k \times t$. Ispisati proizvod matrica a i b . 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.
```

* **Zadatak 0.1.18** Element matrice naziva se *sedlo* ako je istovremeno najmanji u svojoj vrsti, a najveći u svojoj koloni. Program učitava cele brojeve m i n , a zatim i elemente matrice realnih brojeva dimenzije $n \times m$. Ispisati indekse i vrednosti onih elemenata matrice 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
```

* **Zadatak 0.1.19** Program učitava cele brojeve m i n , a zatim i elemente matrice celih brojeva dimenzije $n \times m$. Ispisati elemente matrice u spirалnom 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
```

*** Zadatak 0.1.20** Matrica a se sadrži u matrici b ukoliko postoji podmatrica matrice b identična matrici a . Program učitava dva cela broja m i n , a potom i matricu celih brojeva dimenzije $m \times n$. Potom se učitavaju dva cela broja k i t i matrica celih brojeva dimenzije $k \times t$. Napisati program koji proverava da li se druga matrica sadrži u prvoj učitanj 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 sadrzana 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 sadrzana 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.
```

0.2 Rešenja

Rešenje 0.1.1

```
1 #include <stdio.h>
2 #include <stdlib.h>
3
4 #define MAX 50
5
6 void ucitaj(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 int main()
39 {
40     int mat[MAX][MAX];
41     int m, n;
42
43     ucitaj(mat, &m, &n);
44     ispis(mat, m, n);
45
46     return 0;
47 }
```

47 }

Rešenje 0.1.2

```
1  #include <stdio.h>
2  #include <stdlib.h>
3
4  #define MAX 50
5
6  void ucitaj(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 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 }
46
47 int main()
```

```

{
49     int mat[MAX][MAX], t[MAX][MAX];
    int m, n;

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

55     return 0;
57 }

```

Rešenje 0.1.3

```

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

4 #define MAX 50

6 void ucitaj(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     {
        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("%d", &mat[i][j]);
24 }

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

30     for(i=0; i<m; i++)
    {
32         for(j=0; j<n; j++)
            printf("%d ", mat[i][j]);
34         printf("\n");
    }
36 }

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     ucitaj(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 0.1.4

```

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

4  #define MAX 50

6  void ucitaj(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)
    {

```

```

16     printf("Neispravna dimenzija matrice\n");
    exit(EXIT_FAILURE);
18 }

20 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
28 int main()
29 {
    int mat[MAX][MAX];
    int m, n, i, j, suma;
    int k, t;

32     ucitaj(mat, &m, &n);

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

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

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

50     return 0;
}

```

Rešenje 0.1.5

```

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

4  #define MAX 50

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

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

```

```

12     if (*n <= 0 || *n > MAX || *m <= 0 || *m > MAX)
13     {
14         printf("Neispravna dimenzija matrice\n");
15         exit(EXIT_FAILURE);
16     }
17
18     printf("Uneti matricu celih brojeva\n");
19
20     for(i=0; i<*m; i++)
21         for(j=0; j<*n; j++)
22             scanf("%d", &mat[i][j]);
23 }
24
25 void kreiraj_niz(int mat[][MAX], int m, int n, double b[])
26 {
27     int i, j, suma;
28
29     for(i=0; i<m; i++)
30     {
31         suma = 0;
32         for(j=0; j<n; j++)
33             suma += mat[i][j];
34
35         b[i] = (double)suma/n;
36     }
37 }
38
39 int main()
40 {
41     int mat[MAX][MAX];
42     double b[MAX];
43     int m, n, i;
44
45     ucitaj(mat, &m, &n);
46
47     kreiraj_niz(mat, m, n, b);
48
49     printf("Dobijeni niz je:\n");
50     for(i=0; i<m; i++)
51         printf("%g ", b[i]);
52     printf("\n");
53
54     return 0;
55 }

```

Rešenje 0.1.6

```

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

```

```
5  #define MAX 50
6
7  void ucitaj(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 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++)
```



```

57         if (a[i][j] == 1 && a[j][k] == 1 && a[i][k] == 0)
            return 0;
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     ucitaj(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 simatrica.\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 0.1.7

```

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

```

```
6 void ucitaj(float 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++)
            scanf("%f", &mat[i][j]);
24 }

26 float trag(float a[][MAX], int n)
{
28     float suma = 0;
    int i;

30     for(i=0; i<n; i++)
        suma += a[i][i];

34     return suma;
}

36 float sum_sporedna(float a[][MAX], int n)
38 {
    float suma = 0;
    int i;

40     for(i=0; i<n; i++)
        suma += a[i][n-i-1];

44     return suma;
46 }

48 float sumD(float a[][MAX], int n)
{
50     float suma = 0;
    int i, j;

52     for(i=0; i<n; i++)
        for(j=i+1; j<n; j++)
            suma += a[i][j];

56     return suma;
```

```

58 }

60 float sumd(float a[][MAX], int n)
61 {
62     float suma = 0;
63     int i, j;

64     for(i=0; i<n; i++)
65         for(j=n-i-1; j>i; j--)
66             suma += a[i][j];

67     return suma;
68 }

70 }

72 int main()
73 {
74     float a[MAX][MAX];
75     int n;

76     ucitaj(a, &n);

77     printf("Trag je %.3f.\n", trag(a, n));
80     printf("Suma na sporednoj dijagonali je %.3f.\n", sum_sporedna(a, n
        ));
81     printf("Suma iznad glavne dijagonale je %.3f.\n", sumD(a, n));
82     printf("Suma ispod sporedne dijagonale je %.3f.\n", sumd(a, n));

83     return 0;
84 }

```

Rešenje 0.1.8

```

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

4 #define MAX 50

6 void ucitaj(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     }

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

```

```

20     for(i=0; i<*n; i++)
21         for(j=0; j<*n; j++)
22             scanf("%d", &mat[i][j]);
23 }
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     ucitaj(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 0.1.9

```

1 #include <stdio.h>
2 #include <stdlib.h>
3
4 #define MAX 50
5
6 void ucitaj(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 }

```

```

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 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     ucitaj(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 0.1.10

```

1 #include <stdio.h>
2 #include <stdlib.h>
3
4 #define MAX 50
5
6 void ucitaj(float mat[][MAX], int* n)
7 {
8     int i, j;
9
10

```

```

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

13     if (*n <= 0 || *n > 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<*n; i++)
22         for(j=0; j<*n; j++)
23             scanf("%f", &mat[i][j]);
24 }

25 int main()
26 {
27     float a[MAX][MAX];
28     int n, i, j;
29     float gornji_trougao = 0, donji_trougao = 0;

31     ucitaj(a, &n);

33     for(i=0; i<n/2; i++)
34         for(j=i+1; j<n-i-1; j++)
35             gornji_trougao += a[i][j];

37     for(i=n/2; i<n; i++)
38         for(j=n-i; j<i; j++)
39             donji_trougao += a[i][j];

41     printf("%f %f\n", gornji_trougao, donji_trougao);

43     printf("Razlika je: %.2f\n", gornji_trougao - donji_trougao);

45     return 0;
47 }

```

Rešenje 0.1.11

```

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

4 #define MAX 50

6 void ucitaj(int mat[][MAX], int* m, int* n)
7 {
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);
16   }
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 int main()
   {
28     int a[MAX][MAX];
       int n, i, j, m, x, y, p, k;
30     int suma;

32     ucitaj(a, &m, &n);

34     printf("Uneti dva cela broja: ");
       scanf("%d%d", &p, &k);
36
       printf("Sume podmatrica su: ");
38     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 0.1.12

```

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

4 #define MAX 50

```

```

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

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

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

34     return 1;
}

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

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

44     return 1;
46 }

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

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

56     return 1;
}

```



```

58 int sortirani_po_vrstama(int mat[][MAX], int n)
60 {
62     int i;
64     for(i=0; i<n; i++)
66         if (!sortirana_vrstama(mat, n, i))
68             return 0;
69
70     return 1;
71 }
72
73 int sortirana_glavna(int mat[][MAX], int n)
74 {
76     int i;
78     for(i=0; i<n-1; i++)
80         if (mat[i][i] >= mat[i+1][i+1])
82             return 0;
83
84     return 1;
85 }
86
87 int sortirana_sporedna(int mat[][MAX], int n)
88 {
90     int i;
92     for(i=0; i<n-1; i++)
94         if (mat[i][n-i-1] >= mat[i+1][n-i-2])
96             return 0;
97
98     return 1;
99 }
100
101 int sortirani_po_dijagonalama(int mat[][MAX], int n)
102 {
104     if (!sortirana_glavna(mat, n))
106         return 0;
107
108     if (!sortirana_sporedna(mat, n))
109         return 0;
110
111     return 1;
112 }
113
114 int main()
115 {
116     int mat[MAX][MAX];
117     int n;
118
119     ucitaj(mat, &n);

```

```

110 printf("Sortiranost po vrstama, kolonama i dijagonalama je: ");
    printf("%d ", sortirani_po_vrstama(mat, n));
112 printf("%d ", sortirani_po_kolonama(mat, n));
    printf("%d\n", sortirani_po_dijagonalama(mat, n));
114
    return 0;
116 }

```

Rešenje 0.1.13

```

1  #include <stdio.h>
    #include <stdlib.h>
3
    #define MAX 10
5
void ucitaj(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);
46
47         if (suma1 >= suma2)
48             return 0;
49
50         suma1 = suma2;
51     }
52
53     return 1;
54 }
55
56 int main()
57 {
58     int mat[MAX][MAX];
59     int n;
60
61     ucitaj(mat, &n);
62
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
68     return 0;
69 }

```

Rešenje 0.1.14

```

1  #include <stdio.h>
2  #include <stdlib.h>
3
4  #define MAX 200
5
6  void ucitaj(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     ucitaj(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 0.1.15

```

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

```

```

5 void ucitaj(int mat[][MAX], int* n)
7 {
9     int i, j;

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

13    if (*n <= 0 || *n > 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<*n; i++)
22        for(j=0; j<*n; j++)
23        {
24            scanf("%d", &mat[i][j]);
25        }

26    int suma_kolone(int mat[][MAX], int n, int j)
27    {
28        int i, suma = 0;

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

31        return suma;
32    }

33    int suma_vrste(int mat[][MAX], int n, int i)
34    {
35        int j, suma = 0;

36        for(j=0; j<n; j++)
37            suma += mat[i][j];

38        return suma;
39    }

40    int magicni_kvadrat(int mat[][MAX], int n)
41    {
42        int suma = suma_kolone(mat, n, 0);
43        int i, j;

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

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

```

```

57     return 0;
59     return 1;
61 }
63 int main()
64 {
65     int mat[MAX][MAX];
66     int n;
67
68     ucitaj(mat, &n);
69
70     if (magicni_kvadrat(mat, n))
71         printf("Matrica jeste magicni kvadrat.\n");
72     else
73         printf("Matrica nije magicni kvadrat.\n");
74
75     return 0;
76 }

```

Rešenje 0.1.16

```

1  #include <stdio.h>
2  #include <stdlib.h>
3
4  #define MAX 100
5
6  void ucitaj(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     ucitaj(mat, &n);

34     for(k=0; k<n; k++)
35     {
36         j = k;
37         i = 0;

38         while(j >= 0)
39         {
40             printf("%d ", mat[i][j]);
41             i++;
42             j--;
43         }

44         printf("\n");
45     }

46     for(k=1; k<n; k++)
47     {
48         i = k;
49         j = n-1;

50         while(i < n)
51         {
52             printf("%d ", mat[i][j]);
53             i++;
54             j--;
55         }

56         printf("\n");
57     }

58     return 0;
59 }

```

Rešenje 0.1.17

```

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

4 #define MAX 50

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

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

```

```

12     if (*n <= 0 || *n > MAX || *m <= 0 || *m > MAX)
13     {
14         printf("Neispravna dimenzija matrice\n");
15         exit(EXIT_FAILURE);
16     }
17
18     printf("Uneti matricu celih brojeva\n");
19
20     for(i=0; i<*m; i++)
21         for(j=0; j<*n; j++)
22             scanf("%d", &mat[i][j]);
23 }
24
25 void ispis(int mat[][MAX], int m, int n)
26 {
27     int i, j;
28
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 }
36
37 void mnozenje(int a[][MAX], int m, int n, int b[][MAX], int k, int t,
38               int c[][MAX])
39 {
40     int i, j, w;
41
42     for(i=0; i<m; i++)
43         for(j=0; j<t; j++)
44         {
45             c[i][j] = 0;
46             for(w=0; w<n; w++)
47                 c[i][j] += a[i][w] * b[w][j];
48         }
49 }
50
51 int main()
52 {
53     int a[MAX][MAX], b[MAX][MAX], c[MAX][MAX];
54     int m, n;
55     int k, t;
56
57     ucitaj(a, &m, &n);
58     ucitaj(b, &k, &t);
59
60     if (n != k)
61     {
62         printf("Mnozenje matrica nije moguće.\n");

```



```

        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 0.1.18

```

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

4  #define MAX 50

6  void ucitaj(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++)
        for(j=0; j<*n; j++)
22             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;

32     ucitaj(mat, &m, &n);

34     for(i=0; i<m; i++)
    {
36

```

```

38     min_vrstne = mat[i][0];
    indeks_kolone = 0;

40

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

48     max_kolone = mat[0][indeks_kolone];

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

54     if (min_vrstne == max_kolone)
        printf("Sedlo: %d %d %g\n", i, indeks_kolone, min_vrstne);
56
    }

58     return 0;

60 }

```

Rešenje 0.1.19

```

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

3

   #define MAX 50

5
   void ucitaj(int mat[][MAX], int* m, int* n)
7   {
       int i, j;

9

       printf("Uneti dimenzije matrice: ");
11      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

```

```

27 int main()
{
29     int mat[MAX][MAX];
    int m, n, brojac, i, j;

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

33     ucitaj(mat, &m, &n);

35     gore_i = 1;
37     dole_i = m-1;

39     levo_j = 0;
    desno_j = n-1;

41     i = 0;
43     j = 0;

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

49         switch(pravac)
        {
51             case 1:
                if (j == desno_j)
53                 {
                    pravac = 2;
55                     desno_j--;
                    i++;
57                 }
                else
59                     j++;
                break;
61             case 2:
                if (i == dole_i)
63                 {
                    pravac = 3;
65                     dole_i--;
                    j--;
67                 }
                else
69                     i++;
                break;
71             case 3:
                if (j == levo_j)
73                 {
                    pravac = 4;
75                     levo_j++;
                    i--;
77                 }
        }
    }
}

```

```

79         else
80             j--;
81         break;
82     case 4:
83         if (i == gore_i)
84         {
85             pravac = 1;
86             gore_i++;
87             j++;
88         }
89         else
90             i--;
91     }
92 }
93 return 0;
94 }

```

Rešenje 0.1.20

```

1 #include <stdio.h>
2 #include <stdlib.h>
3
4 #define MAX 50
5
6 void ucitaj(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 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 }

```

```

32     for(i=0; i<= m-k; i++)
        for(j=0; j<= n-t; j++)
        {
34             jeste_pod = 1;
            for(x=0; x<k && jeste_pod; x++)
36                 for(y=0; y<t && jeste_pod; y++)
                    if (a[i+x][j+y] != b[x][y])
38                         jeste_pod = 0;

40             if (jeste_pod)
                return 1;
42         }

44     return 0;
}

46 int main()
48 {
    int a[MAX][MAX], b[MAX][MAX];
50     int m, n;
    int k, t;

52     ucitaj(a, &m, &n);
54     ucitaj(b, &k, &t);

56     if (podmatrica(a, m, n, b, k, t))
        printf("Druga matrica je sadrzana u prvoj matrici.\n");
58     else
        printf("Druga matrica nije sadrzana u prvoj matrici.\n");
60 }

```