

1 Uvod u Python

- 1 Napisati funkciju `sum_d` za računanje sume elemenata glavne dijagonale kvadratne matrice proizvoljnih dimenzija.

Matematički opis: $s = \sum_{i=1}^n a_{i,i}$.

Interfejs funkcije: `sum_d(A)`, gde `A` predstavlja ulaznu matricu. Povratna vrednost funkcije predstavlja sumu elemenata glavne dijagonale. Ukoliko ulazna matrica nije kvadratna, funkcija treba da izazove `Exception`.

Napisati skript datoteku `zadatak1.py` koja testira funkciju za sledeću matricu:

<code>A = [</code>					Rešenje:
2	1	6	1		14
1	3	8	2		
5	9	4	3		
1	1	8	5]		

- 2 Napisati funkciju `sum_greater` za računanje sume onih elemenata matrice koji su veći od prosečne vrednosti elemenata matrice. Ulazna matrica je proizvoljnih dimenzija.

Interfejs funkcije: `sum_greater(A)`, gde `A` predstavlja ulaznu matricu. Povratna vrednost funkcije predstavlja sumu elemenata većih od proseka.

Napisati skript datoteku `zadatak2.py` koja testira funkciju za sledeću matricu:

<code>A = [</code>							Rešenje:
2	1	2	6	8	1	-2	137
15	4	7	18	4	0	12	
11	6	9	-1	4	8	0	
2	8	6	8	1	8	7]	

- 3 Data je matrica `A` dimenzije $n \times m$. Napisati funkciju koja pronalazi vektor `b` dimenzije m čiji i -ti element ($i \in \{1..m\}$) predstavlja indeks reda minimalnog elementa i -te kolone matrice `A`. Zabranjeno je koristiti ugrađenu Python funkciju `min`.

Napisati zadatu funkciju i skript datoteku `zadatak3.py`, koja testira funkciju matricom `A`:

<code>A = [</code>				Rešenje:
-2	5	-3		<code>b = [2 2 0]</code>
-1	-1	0		
-3	-5	1]		

- 4 Napisati funkciju koja prima kvadratnu matricu `A` veličine $n \times n$ i vraća vektor `b` dužine n čiji i -ti element predstavlja zbir elemenata i -tog reda matrice `A` koji se nalaze van sporedne dijagonale matrice `A`.

Napisati zadatu funkciju i skript datoteku `zadatak4.py`, koja testira funkciju matricom `A`:

<code>A = [</code>				Rešenje:
-0.5	10	2		<code>b = [9.5 -1 -9]</code>
-1	1	0		
1	-6	-3]		

- 5 Data je kvadratna matrica `A` dimenzije $n \times n$. Napisati funkciju koja za zadatu matricu vraća vektor `b` dimenzije 2, čiji prvi element predstavlja indeks reda, a drugi element predstavlja indeks kolone minimalnog elementa koji se nalazi na glavnoj ili sporednoj dijagonali matrice `A`.

Napisati zadatu funkciju i skript datoteku `zadatak5.py`, koja testira funkciju matricom `A`:

```
A = [
    -2  5  -7
    -1 -1  -8
    -3 -5   1]
```

Rešenje:
b = [0 2]

- 6 Napisati funkciju koja obrće redosled elemenata neparnih redova matrice.

Napisati zadatu funkciju i skript datoteku *zadatak6.py*, koja testira funkciju matricom A:

```
A = [
    0.3  1  -5
    -1   4   0
    1    5   2]
```

Rešenje:
A = [
 -5 1 0.3
 -1 4 0
 2 5 1]

- 7 Data je matrica A dimenzije $n \times m$. Napisati funkciju koja pronalazi vektor b dimenzije n čiji i -ti element ($i \in \{1..n\}$) predstavlja indeks kolone maksimalnog elementa i -tog reda matrice A. Zabranjeno je koristiti ugrađenu Python funkciju *max*.

Napisati zadatu funkciju i skript datoteku *zadatak7.py*, koja testira funkciju matricom A:

```
A = [
    -2  5  3
    -1 -1  0
    -1 -5 -3]
```

Rešenje:
b = [1 2 0]

- 8 Napisati funkciju koja u svakom redu proizvoljne kvadratne matrice A zamenjuje elemente na glavnoj i sporednoj dijagonali.

Napisati zadatu funkciju i skript datoteku *zadatak8.py*, koja testira funkciju matricom A:

```
A = [
    -0.5  10   2
    -1    1   0
    1    -6  -3]
```

Rešenje:
A = [
 2 10 -0.5
 -1 1 0
 -3 -6 1]

- 9 Napisati funkciju koja obrće redosled elemenata neparnih kolona matrice.

Napisati zadatu funkciju i skript datoteku *zadatak9.py*, koja testira funkciju matricom A:

```
A = [
    0.3  1  -5
    -1   4   0
    1    5   2]
```

Rešenje:
A = [
 1 1 2
 -1. 4 0
 0.3 5 -5]

- 10 Neka su tačke 2D euklidskog prostora definisane kao matrica u kojoj je prva kolona x komponenta a druga kolona y komponenta. Napisati funkciju *diameter* koja za opisanu matricu određuje udaljenost između para najudaljenijih tačaka.

Interfejs funkcije: `diameter(points)`, gde `points` predstavlja ulaznu matricu. Povratnu vrednost funkcije predstavlja izračunatu najveću udaljenost tačaka.

- 10.a Napisati skript datoteku *zadatak10.py* koja testira funkciju za sledeću matricu:

<code>points = [</code>	Rešenje:
<code>1.0 0.0</code>	8.9359
<code>4.0 8.0</code>	
<code>2.1 1.2</code>	
<code>3.2 1.9</code>	
<code>5.6 4.3</code>	
<code>7.9 2.3</code>	
<code>-1.0 3.1];</code>	

- 10.b Nacrtati položaj tačaka.

- 11 Napisati funkciju koja izračunava broj akcija koji treba da se dodeli svakom zaposlenom u zavisnosti od broja godina radnog staža.

Interfejs funkcije: `stocksShare(years, total_stocks)`, gde su `years` godine radnog staža svih zaposlenih, a `total_stocks` ukupan broj akcija za raspodelu svim zaposlenima.

Napisati skript datoteku *zadatak11.py* koja testira funkciju za sledeći ulaz:

<code>years = [2 3 4 6 1 2 4 8]</code>	Rešenje:
<code>stocks = stocks_share(years, 1000)</code>	<code>stocks =</code>
<code>sum(stocks)</code>	67 100 133 200 33 67 133 267
	<code>sum = 1000</code>

- 12 Napisati funkciju koja sortira vrste matrice u odnosu na opadajuću vrednost prvog elementa.

Interfejs funkcije: `sort_by_first_column(A)`, gde je `A` ulazna matrica.

Napisati skript datoteku *zadatak12.m* koja testira funkciju za sledeći ulaz:

<code>A = [</code>	Rešenje:
<code>66 1</code>	<code>A = [</code>
<code>100 2</code>	266 8
<code>133 3</code>	200 4
<code>200 4</code>	133 7
<code>33 5</code>	133 3
<code>66 6</code>	100 2
<code>133 7</code>	66 1
<code>266 8];</code>	66 6
	33 5]