

Vežbe 3 - Zadaci

Zadatak 1. Izgenerisati podatke (kao slučajne brojeve) i smestiti ih u rečnik čiji su ključevi temperatura, relativna vlažnost vazduha, količina ugljendioksida, jačina osvetljenosti. Upisati podatke u fajl pod nazivom „data_log.txt“, tako da svaka kolona čuva izmerene vrednosti.

```
1 import random
2
3 data = {
4     'temperature': [],
5     'humidity': [],
6     'co2_level': [],
7     'illumination': []
8 }
9
10 for key in data.keys():
11     for i in range(10):
12         data[key].append(random.randint(0, 100))
13
14 my_file = open('data_log.txt', 'w+')
15
16 my_file.write('Temperature Humidity CO2 Level Illumination \n')
17
18 for i in range(len(data['temperature'])):
19     my_file.write(str(data['temperature'][i]).ljust(10)+
20                  str(data['humidity'][i]).ljust(10)+
21                  str(data['co2_level'][i]).ljust(10)+
22                  str(data['illumination'][i]).ljust(10)+'\n')
23
24 my_file.close()
```

Zadatak 2. Na osnovu generisanog fajla iz prethodnog zadatka, izvršiti inverznu konverziju. Očitati podatke iz fajla, na osnovu kojih se formira rečnik sa očitanim podacima.

```
1 my_file = open('data_log.txt', 'r+')
2 line = my_file.readline()
3
4 new_data = {
5     'temperature': [],
6     'humidity': [],
7     'co2_level': [],
8     'illumination': []
9 }
10
11 while line:
12     line = my_file.readline()
13     line = line.strip()
14     read_list = line.split(' ')
15
```

```

16     # prvi nacin
17     while '' in read_list:
18         read_list.remove('')
19
20     # drugi nacin
21     #read_list = [el for el in read_list if el != '']
22
23     if read_list == []:
24         break
25
26     new_data['temperature'].append(int(read_list[0]))
27     new_data['humidity'].append(int(read_list[1]))
28     new_data['co2_level'].append(int(read_list[2]))
29     new_data['illumination'].append(int(read_list[3]))
30
31     print(new_data)

```

Zadatak 3. Pronaći minimalni i maksimalni element unete matrice.

```

1  # prvi nacin
2  matrix = [
3      [2,3,4],
4      [-3,7,-1],
5      [0.5, -2.4, 3]]
6
7  max_el_row = []
8  min_el_row = []
9  for i in matrix:
10     max_el_row.append(max(i))
11     min_el_row.append(min(i))
12
13  print('Maksimalni element matrice je {}'.format(max(max_el_row)))
14  print('Minimalni element matrice je {}'.format(min(min_el_row)))
15
16  # drugi nacin
17  import numpy as np
18
19  x = int(input('Unesite broj redova matrice: '))
20  y = int(input('Unesite broj kolona matrice: '))
21
22  matrica = np.random.rand(x,y)
23  print(matrica)
24
25  max_el = np.max(matrica)
26  min_el = np.min(matrica)
27
28  print('Maksimalni element matrice je {}'.format(max_el))
29  print('Minimalni element matrice je {}'.format(min_el))

```

Zadatak 4. Transponovati zadatau matricu.

```
1 matrix = [  
2     [1,2],  
3     [3,4],  
4     [5,6]]  
5  
6 # I nacin  
7 result = [  
8     [0,0,0],  
9     [0,0,0]]  
10 print(result)  
11  
12 for i in range(len(matrix)):  
13     for j in range(len(matrix[0])):  
14         print(j, i)  
15         result[j][i] = matrix[i][j]  
16  
17 print(matrix)  
18 print(result)  
19  
20 # II nacin  
21 result_2 = [[matrix[i][j] for i in range(len(matrix))]  
22             for j in range(len(matrix[0]))]  
23 print(result_2)  
24  
25 # III nacin  
26 import numpy as np  
27  
28 matrix = np.array(matrix)  
29 print(matrix.transpose())
```

Zadatak 5. Odrediti zbir elemenata ispod sporedne dijagonale.

```
1 matrix = [  
2     [1,2,3],  
3     [4,5,6],  
4     [7,8,9]]  
5  
6 sum = 0  
7  
8 size = len(matrix)  
9  
10 for i in range(size):  
11     for j in range(size):  
12         if j < (size-1-i):  
13             sum += matrix[i][j]  
14  
15 print(sum)
```

Zadatak 6. Proveriti da li je uneta matrica magični kvadrat.

```
1  # prvi nacin
2  def magic_square(matrix):
3      size = len(matrix[0])
4      sum_list = []
5
6      # Vertical
7      for col in range(size):
8          sum_list.append(sum(row[col] for row in matrix))
9
10     # Horizontal
11     sum_list.extend([sum(lines) for lines in matrix])
12
13     # Diagonals
14     result_diagonal = 0
15     for i in range(0,size):
16         result_diagonal += matrix[i][i]
17     sum_list.append(result_diagonal)
18
19     result_diagonal =0
20     for i in range(size-1, -1, -1):
21         result_diagonal += matrix[i][i]
22     sum_list.append(result_diagonal)
23
24     if len(set(sum_list)) > 1:
25         return False
26     return True
27
28 matrix = [
29     [2,7,6],
30     [9,5,1],
31     [4,3,8]
32 ]
33
34 magic_square(matrix)
35
36 # drugi nacin
37
38 import numpy as np
39
40 def magic_square_2(matrix):
41     matrix = np.array(matrix)
42
43     sum_list = []
44     sum_list.extend(matrix.sum(axis=0)) # sum columns
45     sum_list.extend(matrix.sum(axis=1)) # sum rows
46
```

```
47     sum_list.append(np.trace(matrix)) # glavna dijagonala
48     sum_list.append(np.trace(np.fliplr(matrix))) # sporedna dijagonala
49
50     if len(set(sum_list)) > 1:
51         return False
52     return True
53
54 magic_square_2(matrix)
```