

Лабораторная работа №3

Вариант 4 (как в списке группы)

Задание:

- 4) . В массиве хранятся сведения о количестве осадков, выпавших за каждый день января. Определить общее количество осадков за январь.
4. В квадратной матрице A(10,10) наибольший элемент среди стоящих на главной и побочной диагоналях поменять местами с элементом, стоящим на пересечении этих диагоналей.

Язык программирования: Python

Выполнение:

1.

```
В массиве хранятся сведения о кол-ве осадков,  
выпавших за каждый день января. Определить общее кол-во осадков за январь.  
'''  
from random import randint, sample  
  
l = [randint(1,50) for _ in range(31)]  
  
print('Список осадков:', l)  
print('Общее кол-во осадков:', sum(l))  
'''  
  
"D:\6 sem\СЯП\3\env\Scripts\python.exe" "D:/6 sem/СЯП/3/1e.py"  
Список осадков: [33, 25, 23, 30, 32, 50, 25, 43, 38, 15, 34, 49, 8, 34, 34, 23, 48, 31, 6, 10, 46, 36, 35, 47, 7, 17, 40, 32, 26, 14, 35]  
Общее кол-во осадков: 926  
  
Process finished with exit code 0
```

2.

```
Квадратичная матрица a 10x10. Наибольший элемент среди  
стоящих на главной и побочной диагонали поменять местами с элементом,  
стоящим на пересечении этих диагоналей  
'''  
from random import randint  
import numpy as np  
a = [[randint(1, 100) for _ in range(9)] for _ in range(9)]  
print('Матрица до:\n', np.matrix(a))  
  
diag = []  
reverse_diag = []  
  
for i in range(9):  
    diag.append(a[i][i])  
    reverse_diag.append(a[i][abs(i-8)])  
  
print('Элементы главной дигонали:', diag)  
print('Элементы побочной дигонали:', reverse_diag)  
  
max_value = max(diag + reverse_diag)
```

```
for i in range(len(diag)):
    if diag[i] == max_value:
        diag[4], diag[i] = diag[i], diag[4]
        print('Максимальны элемент на главной диагонали:', max_value)
        for i in range(9):
            a[i][i] = diag[i]

        break
    if reverse_diag[i] == max_value:
        reverse_diag[5], reverse_diag[i] = reverse_diag[i], reverse_diag[5]
        print('Максимальны элемент на побочной диагонали:', max_value)
        for i in range(9):
            a[i][abs(i-8)] = reverse_diag[i]

        break
print('Матрица после:\n', np.matrix(a))
```

Матрица до:

```
[[ 73  45  27  63  62  37  70   1  44]
 [ 21  23  55  76  28  60  86  64  63]
 [ 69  91  24  86  91  91  14  23  34]
 [   4  49  85  90  91  46  54  97  69]
 [ 90  68  30  17  88  70  34  46  40]
 [ 80  69  82  18  35  51  79 100  39]
 [ 78  16  27   9  91  27  32  34  51]
 [ 69   6  38  75  91  47  58  19  90]
 [ 37  57  21  15  28  26  59  70  56]]
```

Элементы главной дигонали: [73, 23, 24, 90, 88, 51, 32, 19, 56]

Элементы побочной дигонали: [44, 64, 14, 46, 88, 18, 27, 6, 37]

Максимальны элемент на главной диагонали: 90

Матрица после:

```
[[ 73  45  27  63  62  37  70   1  44]
 [ 21  23  55  76  28  60  86  64  63]
 [ 69  91  24  86  91  91  14  23  34]
 [   4  49  85  88  91  46  54  97  69]
 [ 90  68  30  17  90  70  34  46  40]
 [ 80  69  82  18  35  51  79 100  39]
 [ 78  16  27   9  91  27  32  34  51]
 [ 69   6  38  75  91  47  58  19  90]
 [ 37  57  21  15  28  26  59  70  56]]
```