Двумерные массивы

- Двумерный массив – своего рода таблица, с определенным количеством строк и столбцов.
- В двумерном массиве каждый элемент фиксируется номером строки и столбца, на пересечении которых он расположен.

- В математике двумерный массив называется матрицей.

Двумерные массивы в Python

- Для работы с матрицами в Python используются списки. Каждый элемент списка-матрицы содержит вложенный список

```
>>> matrix=[[1,2,3],
            [4,5,6],
            [7,8,9],
            [10,11,12]]
>>> matrix[0][0]
1
>>> matrix[2][1]
8
>>> matrix[3]
[10, 11, 12]
>>> matrix[3,3]
Traceback (most recent call last):
  File "<pyshell#40>", line 1, in <module>
    matrix[3,3]
TypeError: list indices must be integers or slices, not tuple
>>>
```

Вывод матрицы в Python построчно

Здесь:

row – переменная которая отвечает за строки

elem – переменная, которая отвечает за столбцы

matrix – имя матрицы

```
for row in matrix:
    for elem in row:
        print(elem,end=" ")
    print()
```

```
1 2 3
4 5 6
7 8 9
10 11 12
```

Вывод матрицы в Python по элементам

```
>>> for i in range(4):
        for j in range(3):
            print ("matrix(",i,",",j,")=",matrix[i][j])

matrix( 0 , 0 )= 1
matrix( 0 , 1 )= 2
matrix( 0 , 2 )= 3
matrix( 1 , 0 )= 4
matrix( 1 , 1 )= 5
matrix( 1 , 2 )= 6
matrix( 2 , 0 )= 7
matrix( 2 , 1 )= 8
matrix( 2 , 2 )= 9
matrix( 3 , 0 )= 10
matrix( 3 , 1 )= 11
matrix( 3 , 2 )= 12
>>>
```

Ввод матрицы в Python по элементам

Код программы:

```
N = 3
M = 2
#Заполнение массива нулями
A = [ [0]*M for i in range(N) ]
#Ввод элементов массива
for i in range(N):
    for j in range(M):
        print("A(",i,",",j,")=",end = '')
        A[i][j]=int(input())
```

Результат программы:

```
A( 0 , 0 )=1
A( 0 , 1 )=2
A( 1 , 0 )=3
A( 1 , 1 )=4
A( 2 , 0 )=5
A( 2 , 1 )=6
```

Ввод матрицы в Python по строкам

Код программы:

```
n = int(input("кол-во строк в массиве="))
m = int(input("кол-во столбцов в массиве="))
a = []

print("Элементы в строке разделяются пробелами")
print("После введения строки - нажмите пробел")

for i in range(n):
    print("Введите ", m, "элементов", i, "й строки матрицы")
    a.append([int(j) for j in input().split()])
```

Здесь:

split() – разделяет введенную строку на элементы через пробелы

j – введенная строка целого типа

Ввод матрицы в Python по строкам

```
кол-во строк в массиве=4
кол-во столбцов в массиве=3
После введения строки - нажмите пробел
Введите  3 элементов 0 й строки матрицы
1 2 3
Введите  3 элементов 1 й строки матрицы
4 5 6
Введите  3 элементов 2 й строки матрицы
-1 -2 -3
Введите  3 элементов 3 й строки матрицы
0 11 12
```

Ввод матрицы в Python по строкам

```
кол-во строк в массиве=3
кол-во столбцов в массиве=2
Элементы в строке разделяются пробелами
После введения строки - нажмите пробел
Введите  2 элементов 0 й строки матрицы
1 2 3
Введите  2 элементов 1 й строки матрицы
1
Введите  2 элементов 2 й строки матрицы
4 5
>>> for row in a:
        for j in row:
                print(j, end=" ")
        print()

1 2 3
1
4 5
>>> print(a[1][1])
Traceback (most recent call last):
  File "<pyshell#13>", line 1, in <module>
    print(a[1][1])
IndexError: list index out of range
```

Заполнение матрицы случайными числами

```
from random import*
N = int(input("Введите N "))
M = int(input("Введите M "))

matr = [[randint(-1000, 1000) for j in range(M)] for i in range(N)]

print("Исходный массив")
for i in range(N):
    for j in range(M):
        print ("{:4d}".format(matr[i][j]), end = " " )
    print()
```

Здесь:

Для вывода используется

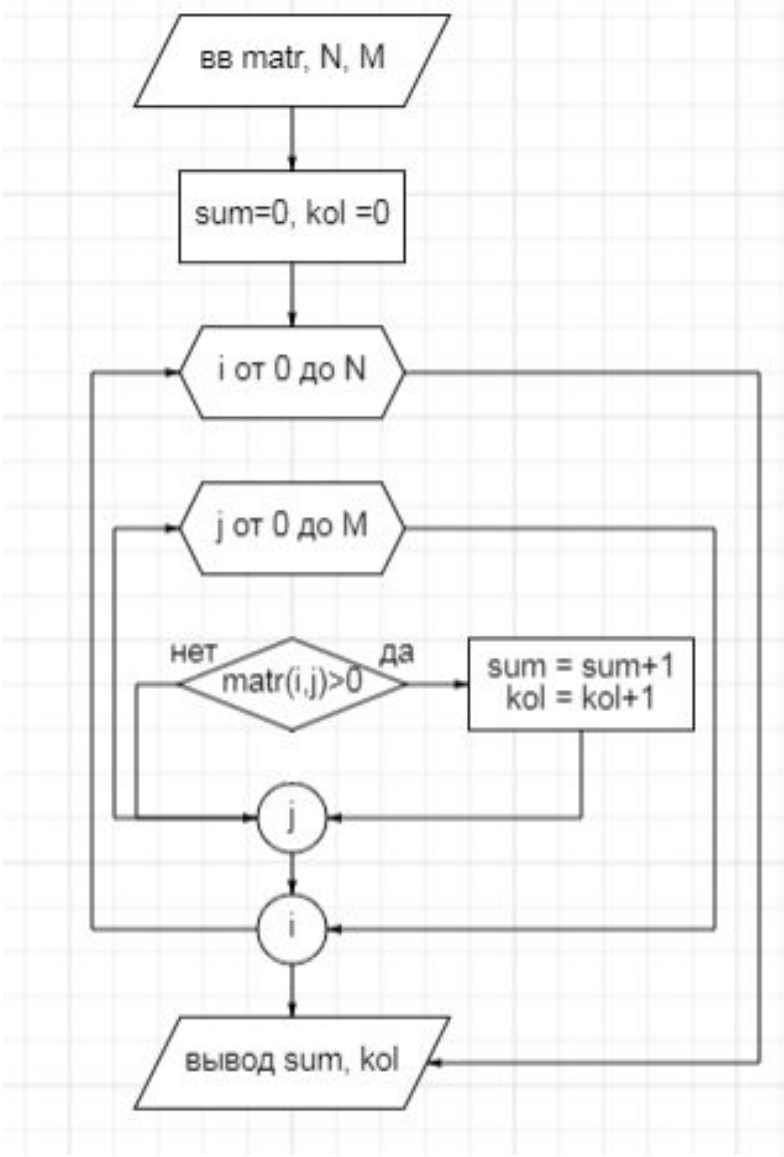
метод format:

{:4d} – длина поля для целых чисел

```
Введите N 4
Введите M 4
Исходный массив
-654 -286 -902  -44
 895 -485  647  356
-348 -404   61 -844
 691  618  438  948
>>>
 RESTART: C:/Users/Alex/

Введите N 4
Введите M 4
Исходный массив
   9   -9   10   -9
   1   10   -8    2
 -10   -4   -1   10
  -4   -4   -1    6
>>>
```

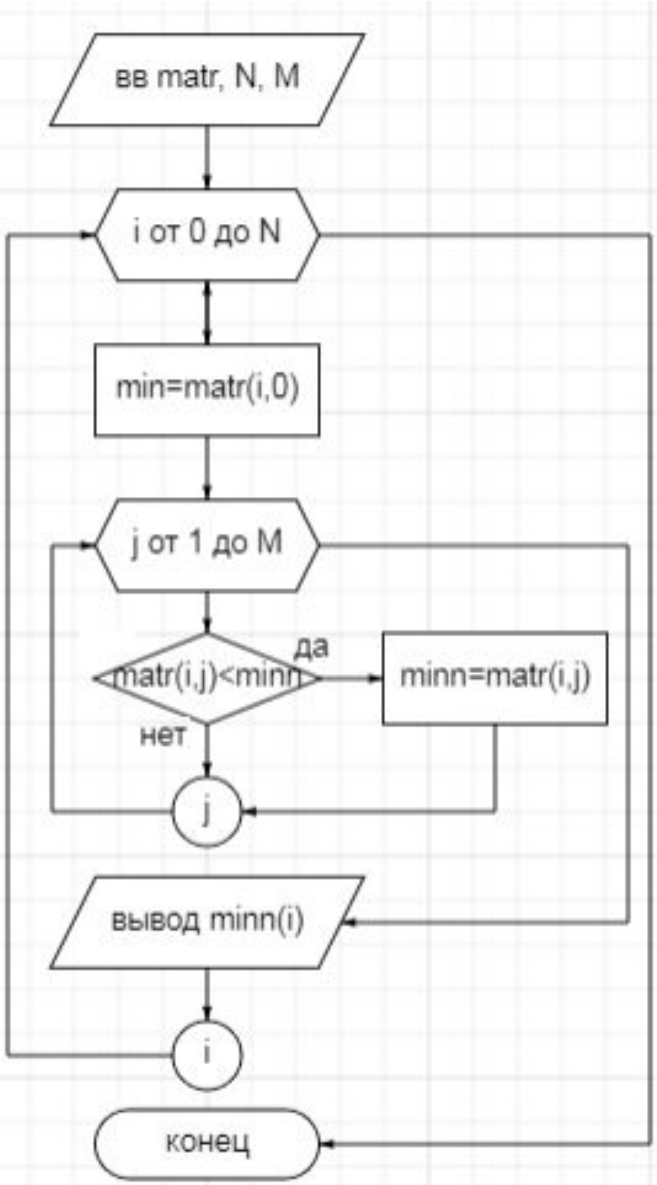
Обработка всех элементов матрицы

Найти сумму и кол-во положительных элементов




```
matr=[[-1, 2, 3],
      [11, -2, 0],
      [0, 15, -7],
      [-2, -3, 10]]
N=4; M=3
sum=0; kol =0
for i in range(N):
    for j in range(M):
        if matr[i][j]>0:
            kol+=1
            sum +=matr[i][j]

print ("sum=", sum)
print("kol=",kol)
```

Обработка элементов матрицы по строкам пример №1



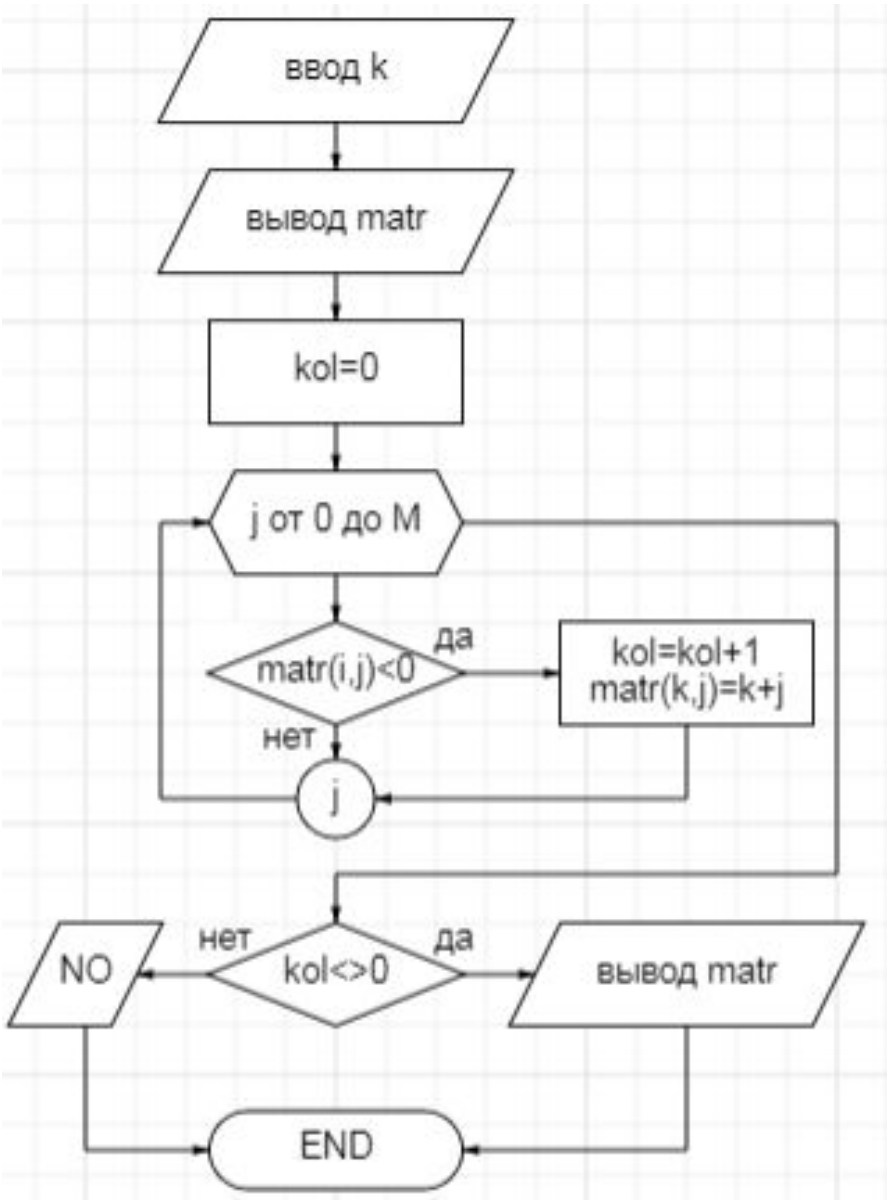
Найти минимальный элемент в каждой строке

```
matr=[[-1, 2, 3],
      [11, -2, 0],
      [0, 15, -7],
      [2, 3, 10]]
N=4; M=3
for i in range(N):
    minn=matr[i][0]
    for j in range(1,M):
        if matr[i][j]<minn:
            minn=matr[i][j]
    print("min в ",i,"-й строке= ",minn)
```

Обработка элементов матрицы по строкам пример №2

В k-й строке заменить отрицательные элементы на сумму его индексов. Если замен не было - вывести сообщение об этом

```
введите номер строки2
Исходный массив
-1 2 3
11 -2 0
0 15 -7
2 3 10
Измененный массив
-1 2 3
11 -2 0
0 15 4
2 3 10
```



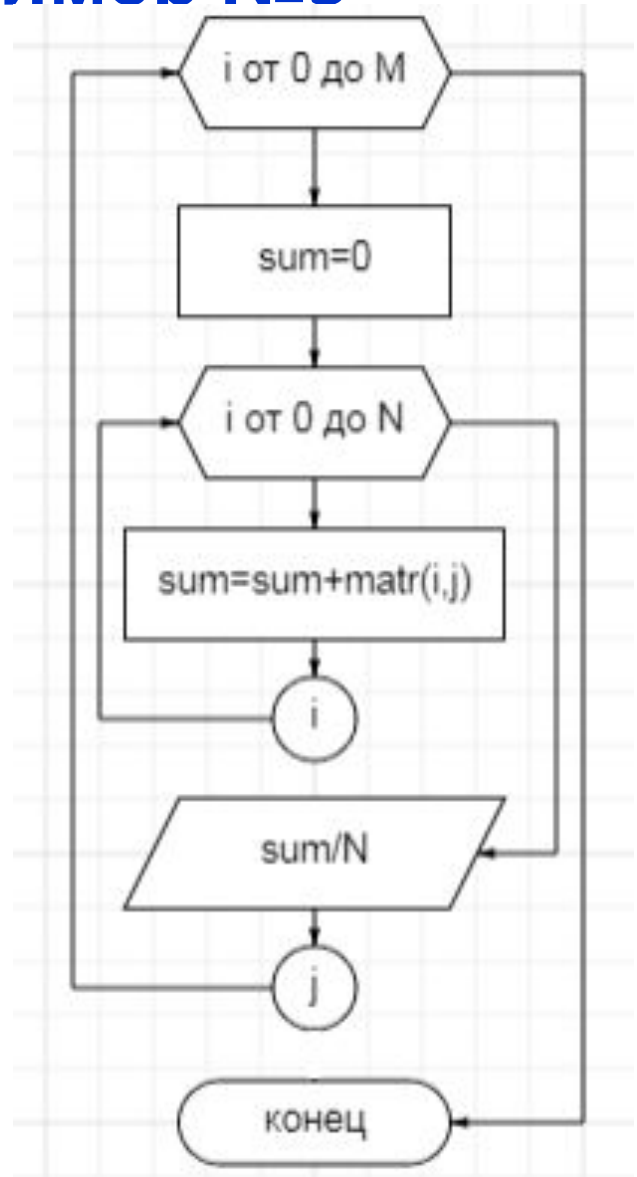
Обработка элементов матрицы по строкам пример №2

В k-й строке заменить отрицательные элементы на сумму его индексов. Если замен не было - вывести сообщение об этом

```
k=int(input("введите номер строки"))
print("Исходный массив")
for i in matr:
    for j in i:
        print(j, end=" ")
    print()
kol=0
for j in range(M):
        if matr[k][j]<0:
            kol+=1
            matr[k][j]=k+j
if kol!=0:
      print("Измененный массив")
      for i in matr:
          for j in i:
              print(j, end=" ")
          print()
else:
      print("Изменений не было")
```

Обработка элементов матрицы по столбцам пример №3

Найти среднюю сумму элементов в каждом столбце



Обработка элементов матрицы по столбцам пример №3

Найти среднюю сумму элементов в каждом столбце

Вариант №1

```
from random import*
N = int(input("Введите N "))
M = int(input("Введите M "))

matr = [[randint(-100, 100) for j in range(M)] for i in range(N)]

print("Исходный массив")
for i in range(N):
    for j in range(M):
        print ("{:8d}".format(matr[i][j]), end = " " )
    print()

for j in range(M):
    sr=0;sum=0
    for i in range(N):
        sum+=matr[i][j]
    print ("среднее в ", j, "-м столбце = ", "{:.2f}".format(sum/N))
```

Обработка элементов матрицы по столбцам пример №3

Найти среднюю сумму элементов в каждом столбце

Вариант №2

```
from random import*
N = int(input("Введите N "))
M = int(input("Введите M "))

matr = [[randint(-100, 100) for j in range(M)] for i in range(N)]

print("Исходный массив")
for i in range(N):
   for j in range(M):
      print ("{:8d}".format(matr[i][j]), end = " " )
   print()
sred=[]
for j in range(M):
    sr=0;sum=0
    for i in range(N):
        sum+=matr[i][j]
    sred.append(int((sum/N)*100/100))

print ("среднее в каждом столбце", sred)
```

Обработка элементов матрицы по столбцам пример №4

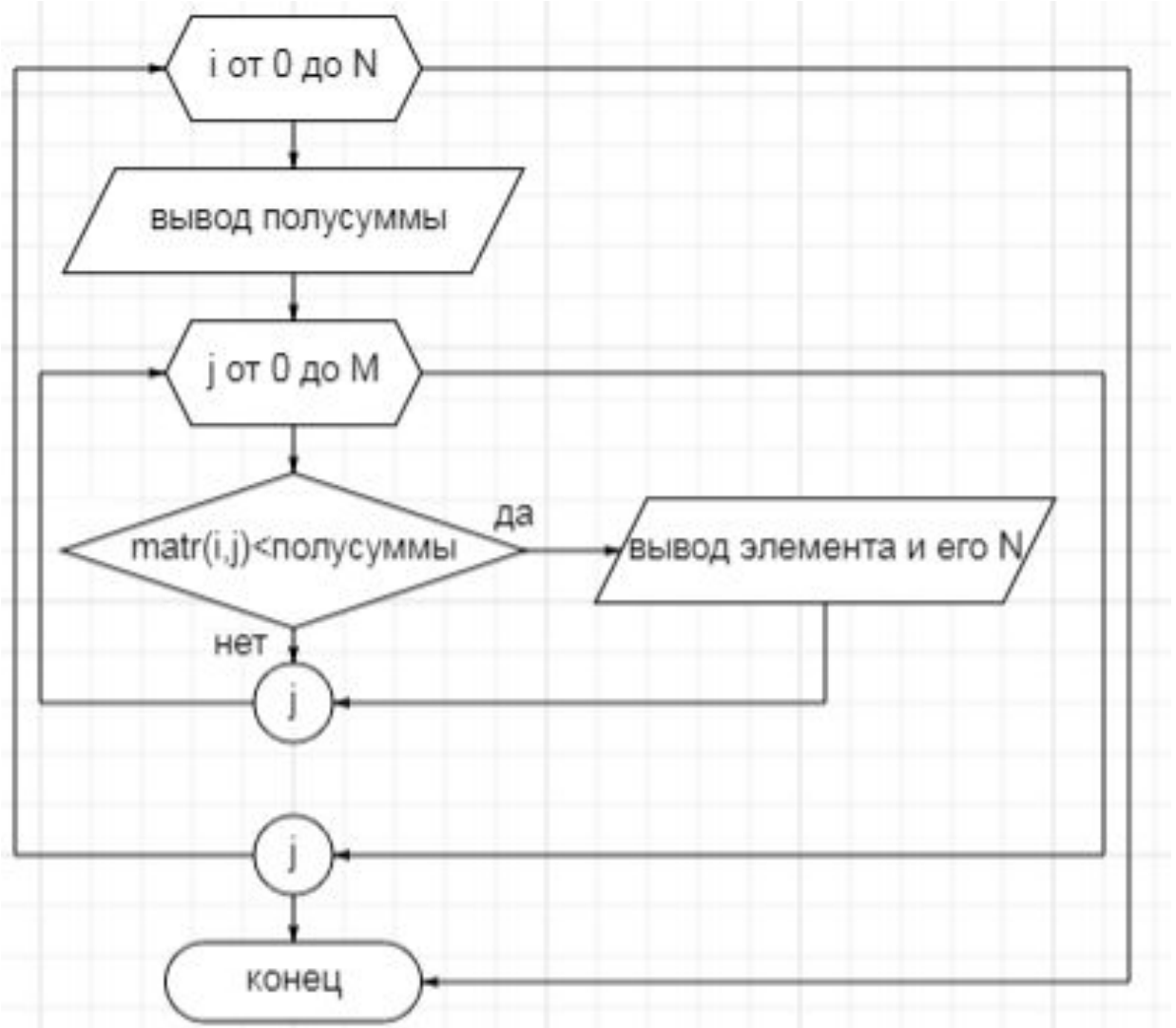
В каждом втором столбце найти количество элементов, кратных 3 и заменить их на номер строки, в которой они расположены + общее количество замен

```
Введите N 4
Введите M 6
Исходный массив
      39      -75       63       11       72      -39
      14       14      -53      -67       47       24
     -30      -45      -13      -95      -70      -22
      33      -99       45        5       35      -10
В  0 -м столбце  3 замен
В  2 -м столбце  2 замен
В  4 -м столбце  1 замен
всего замен =  6
измененный массив
       6      -75        6       11        6      -39
      14       14      -53      -67       47       24
       8      -45      -13      -95      -70      -22
       9      -99        9        5       35      -10
```

```
from random import*
N = int(input("Введите N "))
M = int(input("Введите M "))

matr = [[randint(-100, 100) for j in range(M)] for i in range(N)]

print("Исходный массив")
for i in range(N):
   for j in range(M):
      print ("{:8d}".format(matr[i][j]), end = " " )
   print()

kolob = 0
for j in range(0,M,2):
    kol3=0
    for i in range(N):
        if matr[i][j]%3==0:
            kol3+=1
    if kol3!=0: print("В ",j,"-м столбце ", kol3, "замен")
    kolob+=kol3
if kolob==0:
    print("Замен нет")
else:
    print("всего замен = ",kolob)
    print("измененный массив")
    for i in range(N):
        for j in range(M):
            if matr[i][j]%3==0 and j%2==0: matr[i][j]=kolob+i
            print ("{:8d}".format(matr[i][j]), end = " " )
        print()
```

Обработка элементов матрицы по строкам пример №5

Если элемент матрицы больше полусуммы 1го и последнего элемента строки, где он находится, вывести этот элемент и его индекс.



Обработка элементов матрицы по строкам пример №5

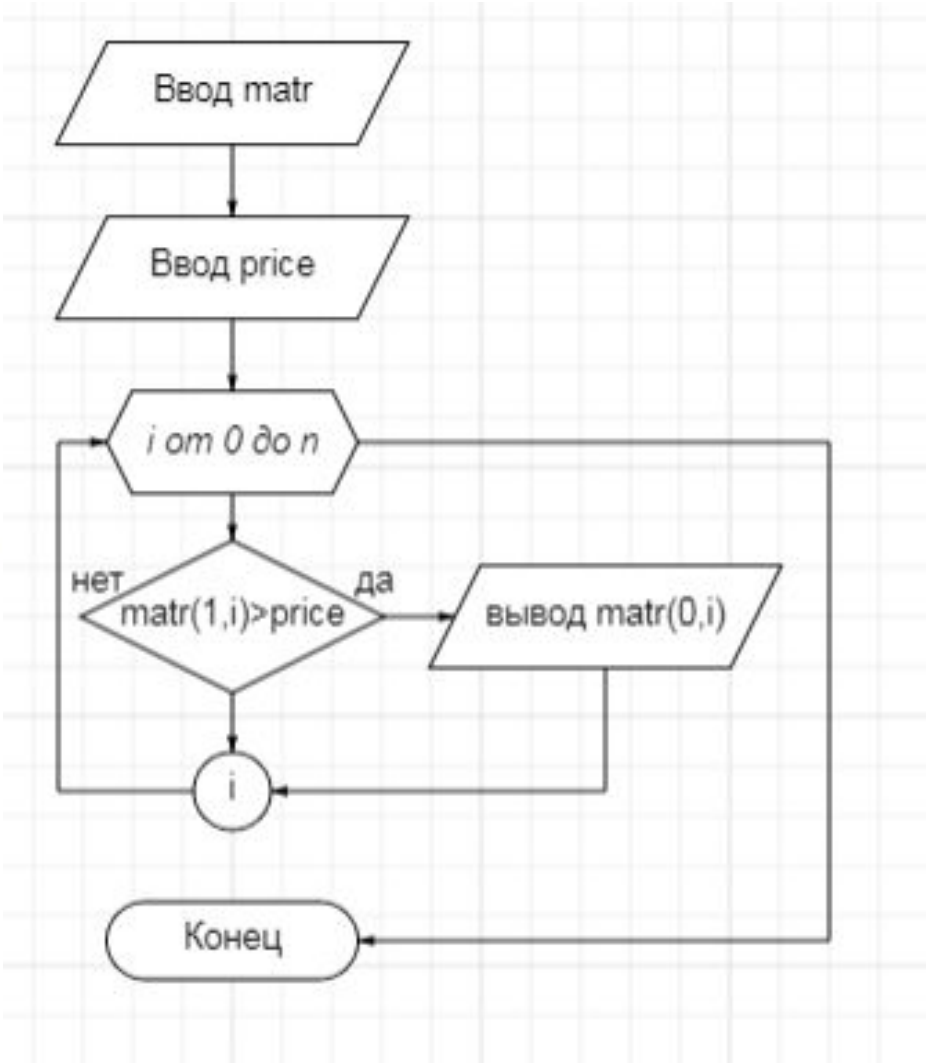
Если элемент матрицы больше полусуммы 1го и последнего элемента строки, где он находится, вывести этот элемент и его индекс.

```
from random import*
N = int(input("Введите N "))
M = int(input("Введите M "))

matr = [[randint(-100, 100) for j in range(M)] for i in range(N)]

print("Исходный массив")
for i in range(N):
    for j in range(M):
        print ("{:8d}".format(matr[i][j]), end = " " )
    print()
for i in range(N):
    print("polusumma ", i, "-ой строки = ",(matr[i][1]+ matr[i][M-1])/2)
    for j in range(M):
        if matr[i][j]<(matr[i][1]+ matr[i][M-1])/2:
            print ("matr(",i,",",j,")=",matr[i][j])
```

Соответствие строк/столбцов. Пример №6

Задание: Фрукты продаются по соответствующим ценам. С клавиатуры ввести цену. На экран вывести те фрукты, цена на которые больше введенной.

```
Исходный массив:
апельсины - 100.0
лимоны - 120.0
киви - 150.0
манго - 300.0
яблоки - 90.0
введите цену 120
фрукты, цена на которые > введенной:
киви
манго
```



Соответствие строк/столбцов. Пример №6

Пример №6. Задание: Фрукты продаются по соответствующим ценам. С клавиатуры ввести цену. На экран вывести те фрукты, цена на которые больше введенной.

```
matr=[["апельсины","лимоны", "киви", "манго", "яблоки"],
      [100.00, 120.00, 150.00,300.00,90.00]]

print("Исходный массив:")
for i in range (5):
   print (matr[0][i], "-", matr[1][i])

price = float(input("введите цену "))
print ("Фрукты, цена на которые > введенной:")
for i in range (5):
    if matr[1][i]>price:
        print (matr[0][i])
```

Квадратные матрицы

Если сумма количество столбцов в матрице равна сумме ее строк, то такой массив называется квадратной матрицей.

Квадратная матрица характеризуется диагональю.

Для обработки элементов, расположенных на главной диагонали, достаточно одного цикла.

Поиск суммы

элементов главной диагонали

```
for i in range(N):
    sum+=matr[i][i]
```

-6	19	28	20	84	87	30
17	6	6	54	31	31	61
22	53	10	8	48	-2	35
80	16	68	27	18	81	53
52	32	-1	46	59	81	73
-8	44	81	33	57	40	41
36	25	30	16	-5	14	87

Квадратные матрицы

Обработка элементов, лежащих выше главной диагонали:

```
for i in range(N):
    for j in range(N):
        if i<j:
            ….
```

Обработка элементов, лежащих ниже главной диагонали:

```
for i in range(N):
    for j in range(N):
        if i>j:
            ….
```

-6	19	28	20	84	87	30
17	6	6	54	31	31	61
22	53	10	8	48	-2	35
80	16	68	27	18	81	53
52	32	-1	46	59	81	73
-8	44	81	33	57	40	41
36	25	30	16	-5	14	87