

Задание 1

С клавиатуры вводится двумерный массив. Посчитать сумму элементов двумерного массива, удовлетворяющих условию, в соответствии с вариантом.

Пример

Элементов массива, у которых сумма квадратов номера строки и номера столбца равна заданному числу x .

```
n = int(input('Введите кол-во строк матрицы: '))
matrix = []
for i in range(n):
    s = input(f'Строка {i}: ')

    row = []
    for elem in s.split():
        row.append(int(elem))
    matrix.append(row)

x = int(input('Введите x: '))
s = 0

for i, row in enumerate(matrix):
    for j, elem in enumerate(row):
        if i ** 2 + j ** 2 == x:
            s += elem

print(f'sum: {s}')
```

Индивидуальные задания

1. Элементов, у которых номер строки равен номеру столбца.
2. Элементов, у которых разность номера строки и номера столбца равна 1.
3. Элементов, у которых сумма номера строки и номера столбца равна $n - 1$.
4. Элементов, у которых сумма номера строки и номера столбца равна числу x . Число x вводится.
5. Элементов, которые меньше номера своей строки.
6. Элементов, которые больше суммы номера своего столбца и номера своей строки.
7. Элементов, которые расположены на главной и побочной диагоналях. В этом варианте считать, что матрица квадратная.

Квадратная матрица – это такая матрица, у которой одинаковое число строк и столбцов.

Главная диагональ матрицы – это элементы от верхнего левого до нижнего правого.

Побочная диагональ матрицы – это элементы от верхнего правого до нижнего левого.

Задание 2

С клавиатуры вводится двумерный массив. Определить, есть ли в двумерном массиве хотя бы один элемент с заданными свойствами, если “да”, то вывести номера его строки и столбца.

Пример

Элемент, кратный x . Число x тоже вводится.

```
n = int(input('Введите кол-во строк матрицы: '))
matrix = []
for i in range(n):
    s = input(f'Строка {i}: ')

    row = []
    for elem in s.split():
        row.append(int(elem))
    matrix.append(row)

x = int(input('Введите x: '))
result = None

for i, row in enumerate(matrix):
    for j, elem in enumerate(row):
        if elem % x == 0:
            result = i, j
            break
```

```
if result is None:
    print('not found')
else:
    print(result)
```

Индивидуальные задания

1. Неравный нулю элемент.
2. Элемент, являющийся нечетным числом.
3. Кратный 4 элемент.
4. Элемент, делящийся на 3 и на 5.
5. Элемент, который при делении на 7 дает в остатке 1, 2 или 3.
6. Элемент, который при делении на 3 и на 5 дает одинаковые остатки.
7. Элемент, который при делении на 3 и на 5 дает нечетные остатки.

Задание 3

С клавиатуры вводится двумерный массив.

Пример

Найти минимальный и максимальный элементы и вывести индексы их местоположения.

```
n = int(input('Введите кол-во строк матрицы: '))
matrix = []
for i in range(n):
    s = input(f'Строка {i}: ')

    row = []
    for elem in s.split():
        row.append(int(elem))
    matrix.append(row)

max_elem, max_i, max_j = None, None, None
min_elem, min_i, min_j = None, None, None

for i, row in enumerate(matrix):
    for j, elem in enumerate(row):
        if max_elem is None or elem > max_elem:
            max_elem, max_i, max_j = elem, i, j
        if min_elem is None or elem < min_elem:
            min_elem, min_i, min_j = elem, i, j

print(f'min: {min_i}, {min_j}')
print(f'max: {max_i}, {max_j}')
```

```
Введите кол-во строк матрицы: 3
Строка 0: 100 200 300
Строка 1: 3 1 2
Строка 2: 30 20 10
min: 1, 1
max: 0, 2
```

Индивидуальные задания

1. Вывести номер последнего столбца, содержащего max элемент.
(т.е. если max элемент встречается в разных столбцах, выбирать самый правый)
2. Вывести номера столбцов, в которых есть элементы равные min.
3. Вывести номер строки, сумма элементов которой минимальна. Если таких несколько, то достаточно вывести номер последней из них.
4. Вывести номера столбцов, суммы элементов в которых не превышают заданного значения х.
5. Вывести номер столбца, сумма элементов которого max. Если таких несколько, то вывести все.
6. Вывести номера строк, элементы которых образуют возрастающую последовательность.
7. Вывести номера строк, суммы элементов в которых равны.

Задание 4

Общее условие

С клавиатуры вводятся натуральные числа n , m – количество строк и столбцов матрицы. Если нужно, вводится ещё что-то. Создать матрицу заданного размера, состоящую из случайных чисел от 1 до 10. Вывести эту матрицу на экран. Передать эту матрицу в качестве аргумента в функцию, указанную в варианте, результат функции вывести на экран.

Пример

Функция **matrix_transposed(matrix)**.

Принимает аргумент **matrix** – исходную матрицу.

Возвращает транспонированную матрицу.

Транспонирование матрицы – это когда элементы строк исходной матрицы составляют столбцы, а элементы столбцов – строки. Матрица типа переворачивается относительно её диагонали.

```
import random

def zeros(n, m):
    result = []
    for i in range(n):
        result.append([0] * m)
    return result

def random_matrix(n, m):
    result = zeros(n, m)
    for i in range(n):
        for j in range(m):
            result[i][j] = random.randint(1, 10)
    return result

def max_matrix_elem_len(mat):
    max_len = 0
    for row in mat:
        for elem in row:
            max_len = max(max_len, len(str(elem)))
    return max_len

def print_matrix(mat):
    w = max_matrix_elem_len(mat)
    for row in mat:
        for elem in row:
            print(f'{elem :{w}}', end=' ')
        print()

def matrix_transposed(matrix):
    n, m = len(matrix), len(matrix[0])

    result = zeros(m, n)
    for i in range(n):
        for j in range(m):
            result[j][i] = matrix[i][j]

    return result

n, m = map(int, input('n, m: ').split())
matrix = random_matrix(n, m)

print('исходная матрица:')
print_matrix(matrix)

matrix_t = matrix_transposed(matrix)
print('транспонированная матрица:')
print_matrix(matrix_t)
```

```
n, m: 3 7
исходная матрица:
5 6 9 6 6 7 6
7 10 4 2 3 2 6
1 9 6 2 6 9 8
транспонированная матрица:
5 7 1
```

```
6 10 9
9 4 6
6 2 2
6 3 6
7 2 9
6 6 8
```

Индивидуальные задания

Не забудьте про общее условие!

1. Функция **matrix_row(matrix, i)**.
Принимает аргументы: matrix – исходная матрица, i – индекс.
Возвращает список, состоящий из элементов i-той строки матрицы.
2. Функция **matrix_column(matrix, j)**.
Принимает аргументы: matrix – исходная матрица, j – индекс.
Возвращает список, состоящий из элементов j-того столбца матрицы.
3. Функция **matrix_trace(matrix)**.
Принимает аргумент matrix – исходная квадратная матрица.
Возвращает список, состоящий из элементов, расположенных на главной диагонали матрицы.
Квадратная матрица – это такая матрица, у которой одинаковое число строк и столбцов.
Главная диагональ матрицы – это элементы от верхнего левого до нижнего правого.
4. Функция **matrix_sum_rows(matrix)**.
Принимает аргумент matrix – исходная матрица.
Возвращает список, состоящий из сумм элементов каждой строки матрицы.
5. Функция **matrix_sum_columns(matrix)**.
Принимает аргумент matrix – исходная матрица.
Возвращает список, состоящий из сумм элементов каждого столбца матрицы.
6. Функция **matrix_argmax_rows(matrix)**.
Принимает аргумент matrix – исходная матрица.
Возвращает список, состоящий из индексов максимальных элементов в каждой строке матрицы.
7. Функция **matrix_argmax_columns(matrix)**.
Принимает аргумент matrix – исходная матрица.
Возвращает список, состоящий из индексов максимальных элементов в каждом столбце матрицы.