# **ФГБОУ ВО «Московский политехнический университет»**

## **Лабораторная работа №5**

Двумерный массивы и функции

**Задание 1 Вариант №19**

По дисциплине:

Программирование

Выполнил

студент 1 курса

группы 201-321

Сметанина А.А.

Проверил:

\_\_\_\_\_\_\_\_\_\_\_\_ Никишина И.Н.

# **МОСКВА 2020**

**Цель работы**

Получить практический навык в написании программ обработки двумерных массивов с

использованием функций.

**Постановка задачи**

Дана целочисленная квадратная матрица. Определить:

1. Сумму элементов в тех строках, которые не содержат отрицательных элементов.
2. Минимум среди сумм элементов диагоналей, параллельных главной диагонали матрицы.

**Теоретическая часть**

В Python реализовать массив можно через вложенные списки. В следующем

листинге программы показан прием формирования двумерного списка и способы

инициализации его элементов. В этой программе следует обратить внимание на то, что

элемент списка, ранее инициированный числом, получает значение строкового типа и

преобразование делается по умолчанию.

Для работы с матрицей использован модуль NumPy, позволяющий выполнять все

необходимые математические операции с n-мерными массивами.

**Описание программы**

Программа написана на алгоритмическом языке Python 3.6, реализована в среде macOS

Catalina и состоит из частей, отвечающих за ввод данных, вычисление и представление данных на экране монитора.

**Описание алгоритма**

1. Получаем на вход размерность матрицы – одно натуральное число, поскольку матрица квадратная. Печатаем сообщение и завершаем выполнение программы в случае ошибки.
2. Создаём массив, заполняя его псевдослучайными целыми числами из некоторого диапазона (numpy.random.randint требует этого).
3. Печатаем полученную матрицу.
4. Находим сумму строк, где все элементы неотрицательны и печатаем её.
5. Находим минимальную сумму диагоналей, параллельных главной, и смещение (то есть положение этой диагонали) и печатаем её.
6. Печатаем матрицу, выделяя диагональ, в которой найдена минимальная сумма.

**Описание входных и выходных данных**

Программа получает на вход натуральное число, выводит целые числа и строки.

**Листинг программы**

import colorama

import numpy as np

def min\_diagonal\_sum(matrix):

\_, size = matrix.shape

min\_d, offset = sum(matrix.diagonal()), 0

for i in range(-size + 2, size - 1):

if sum(matrix.diagonal(i)) < min\_d:

min\_d, offset = sum(matrix.diagonal(i)), i

return offset, min\_d

def summ(matrix):

res = 0

for row in matrix:

if all(i >= 0 for i in row):

res += sum(row)

return res

def justify(s):

return str(s).rjust(3).ljust(3)

def highlight\_diagonal(matrix, offset):

\_, size = matrix.shape

is\_d = lambda n, m: n + offset == m

for row in range(size):

for i in range(size):

if is\_d(row, i):

out = str(matrix[row][i])

print(colorama.Fore.GREEN, justify(out), sep='', end=' ')

print(colorama.Fore.RESET, end='', sep='')

else:

print(justify(matrix[row][i]), end=' ', sep='')

print()

def print\_matrix(matrix):

for row in matrix:

print(' '.join(justify(i) for i in row))

def main():

try:

size = int(input("Введите размерность матрицы: "))

assert size > 0

except ValueError:

print("Неверный ввод, ожидалось натуральное число")

return

except AssertionError:

print("Неверный ввод, ожидалось натуральное число")

return

matrix = np.random.randint(low=-100, high=100, size=(size, size))

print(f"Матрица {size}x{size}: ")

print\_matrix(matrix)

print(f"Сумма строк с элементами >= 0: {summ(matrix)}")

offset, min\_d = min\_diagonal\_sum(matrix)

print(f"Мин сумма элементов диагоналей, || главной: {min\_d}")

print("Которая находится: ")

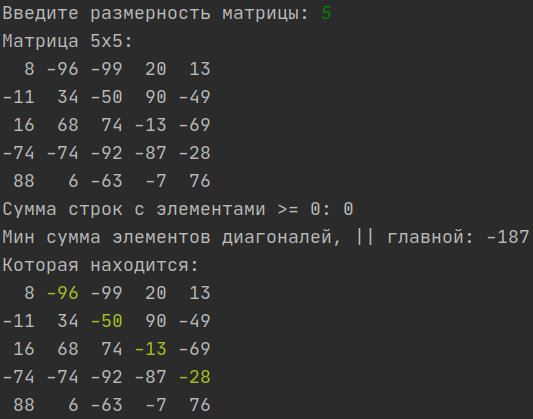
highlight\_diagonal(matrix, offset)

if \_\_name\_\_ == "\_\_main\_\_":

main()

Блок-схема приведена в Приложении 1.

**Результат работы программы**



**Список используемой литературы**

1. <https://pythonworld.ru>
2. [https://ru.wikibooks.org/wiki/Python/Справочник\_по\_языку\_Python\_3.1](https://ru.wikibooks.org/wiki/Python/)

**Приложение 1**

