Лабораторная работа №5   
Гайворонская Екатерина Александровна   
010304 КМСб-о23

ЛАБОРАТОРНАЯ РАБОТА 5.

Использование двумерных массивов для работы с данными

Задачи:

1. Изучить порядок создания и использования двухмерных массивов.

2. Изучить основные задачи обработки двухмерных матриц.

3. Научиться применять двухмерные массивы для решения задач, в которых требуется обработка матричных данных.

Вариант p = 0 q = 1

**Порядок выполнения работы**

**Задача1.** Вычислить сумму и число положительных элементов матрицы A, размера n×n , которые расположены над главной диагональю. Число n задается условием 2 ≤ n ≤ 10 .

Порядок решения:

1. Создаем функцию, которая будет считать сумму элементов и их кол-во.
2. Заполняем двумерный массив и ставим ограничения 2<=n<=10

Программа для решения задачи:

def diagonal(matrix):

    total\_sum = 0

    count\_ = 0

    for i in range(len(matrix)):

        for j in range(i + 1, len(matrix)):

            if matrix[i][j] > 0:

                total\_sum += matrix[i][j]

                count\_ += 1

    return total\_sum, count\_

n = int(input("Введите размер матрицы n (2<=n<=10): "))

if 2 <= n <= 10:

    A = []

    for i in range(n):

        row = list(map(int, input(f"Введите элементы {i+1}-й строки через пробел: ").split()))

        A.append(row)

    for row in A:

        print(row)

    sum\_result, positive\_count = diagonal(A)

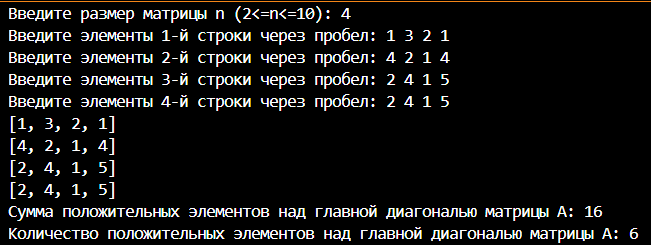
    print(f"Сумма положительных элементов над главной диагональю матрицы A: {sum\_result}")

    print(f"Количество положительных элементов над главной диагональю матрицы A: {positive\_count}")

else:

    print("Неверный размер матрицы. Пожалуйста, введите n от 2 до 10.")

Результаты выполнения программы:



**Задача2.** Дана целая квадратная матрица n-го порядка. Определить, является ли она магическим квадратом, т.е. такой матрицей, в которой для элементов каждой строки и соответствующего столбца выполняется условие: сумма элементов в i-й строке равна сумме элементов в i-м столбце, i = 0, n - 1 .

Порядок решения:

1. Создаем функцию, которая будет проверять сумму элементов в строках и столбцах.
2. Заполняем матрицу и проверяем условия магического квадрата.

Программа для решения задачи:

def is\_magic\_square(matrix):

    n = len(matrix)

    # Проверяем суммы элементов в строках

    for row in matrix:

        if sum(row) != sum(matrix[0]):

            return False

    # Проверяем суммы элементов в столбцах

    for j in range(n):

        if sum(matrix[i][j] for i in range(n)) != sum(matrix[i][0] for i in range(n)):

            return False

    return True

n = int(input("Введите размер квадратной матрицы n: "))

square\_matrix = []

print(f"Введите элементы квадратной матрицы {n}x{n}:")

for \_ in range(n):

    row = list(map(int, input().split()))

    square\_matrix.append(row)

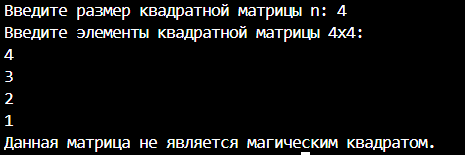
if is\_magic\_square(square\_matrix):

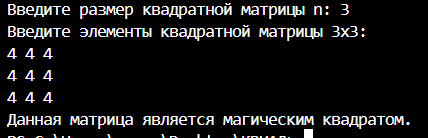
    print("Данная матрица является магическим квадратом.")

else:

    print("Данная матрица не является магическим квадратом.")

Результаты выполнения программы:





**Задача3.** Дана матрица, состоящая из действительных чисел, размером n×m . С экрана вводятся числа k и р, указывается к строкам или столбцам относятся эти номера. Программа должная поменять местами k-ю строку (или столбец) с р строкой (или столбцом) матрицы по требованию пользователя. Результат вывести в виде матрицы.

Порядок решения:

1. Создаем две функции для переворачивания строк и столбцов.
2. Создаем функцию для ввода матрицы.
3. Выбираем элементы для замены

Программа для решения задачи:

def swap\_rows(matrix, k, r):

    matrix[k-1], matrix[r-1] = matrix[r-1], matrix[k-1]

def swap\_columns(matrix, k, r):

    for i in range(len(matrix)):

        matrix[i][k-1], matrix[i][r-1] = matrix[i][r-1], matrix[i][k-1]

def print\_matrix(matrix):

    for row in matrix:

        print(' '.join(map(str, row)))

n = int(input("Введите количество строк матрицы: "))

m = int(input("Введите количество столбцов матрицы: "))

matrix = []

for \_ in range(n):

    row = list(map(float, input().split()))

    matrix.append(row)

print("Введите номера строк k и r для замены:")

k = int(input())

r = int(input())

print("Выберите замену: 1 - строки, 2 - столбцы")

choice = int(input())

if choice == 1:

    swap\_rows(matrix, k, r)

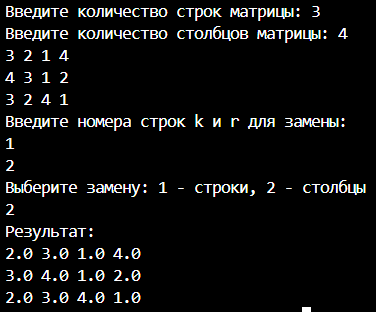
elif choice == 2:

    swap\_columns(matrix, k, r)

print("Результат:")

print\_matrix(matrix)

Результаты выполнения программы:



**Задача4.** Определить для двухмерного массива А размера n×m наибольший и наименьший элемент. Указать в какой строке и в каком столбце они расположены.

Порядок решения:

1. Создаем функцию для поиска минимального и максимального элемента.
2. Заполняем массив.

Программа для решения задачи:

def find\_min\_max(matrix):

    min\_element = matrix[0][0]

    max\_element = matrix[0][0]

    min\_row = 0

    min\_col = 0

    max\_row = 0

    max\_col = 0

    for i in range(len(matrix)):

        for j in range(len(matrix[i])):

            if matrix[i][j] < min\_element:

                min\_element = matrix[i][j]

                min\_row = i

                min\_col = j

            if matrix[i][j] > max\_element:

                max\_element = matrix[i][j]

                max\_row = i

                max\_col = j

    return min\_element, min\_row, min\_col, max\_element, max\_row, max\_col

n = int(input("Введите количество строк массива: "))

m = int(input("Введите количество столбцов массива: "))

matrix = []

for i in range(n):

    row = list(map(int, input(f"Введите элементы {i+1}-й строки через пробел: ").split()))

    matrix.append(row)

min\_element, min\_row, min\_col, max\_element, max\_row, max\_col = find\_min\_max(matrix)

print(f"Массив:")

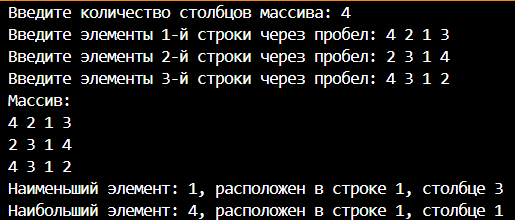
for row in matrix:

    print(' '.join(map(str, row)))

print(f"Наименьший элемент: {min\_element}, расположен в строке {min\_row + 1}, столбце {min\_col + 1}")

print(f"Наибольший элемент: {max\_element}, расположен в строке {max\_row + 1}, столбце {max\_col + 1}")

Результаты выполнения программы:



**Задача5.** Задан двухмерный массив А размера n\*n . Распечатайте его элементы, двигаясь по часовой стрелке сначала по первой строке от a11 до a1n, затем по n-му столбцу от a2 до ann a , затем по n-й строке от an n-1 до an1, потом по первому столбцу от an-11 до a21. После этого проделываем тот же порядок действий, продолжая двигаться по второй строке, затем по n-1-му столбцу, потом по n-1-й строке и по второму столбцу. Продолжая так, пока не будут распечатаны все элементы.

Порядок решения:

1. Создаем функцию для двумерного массива по часовой стрелке.

Программа для решения задачи:

def print\_clockwise(arr, n):

    for i in range(n // 2):

        for j in range(i, n - i):

            print(arr[i][j], end=" ")

        for j in range(i + 1, n - i):

            print(arr[j][n - i - 1], end=" ")

        for j in range(n - i - 2, i - 1, -1):

            print(arr[n - i - 1][j], end=" ")

        for j in range(n - i - 2, i, -1):

            print(arr[j][i], end=" ")

# Пример

matrix = [[1, 2, 3], [4, 5, 6], [7, 8, 9]]

n = 3

print\_clockwise(matrix, n)

Результаты выполнения программы:

