Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение высшего образования «Ульяновский государственный технический университет»

Кафедра «Вычислительная техника»

**Отчет по лабораторной работе №2**

Дисциплина: «Разработка профессиональных приложений»

**Двумерные структуры данных**

Вариант №30

Выполнил:

студент группы ИВТАСбд-21

Шелепова С.Д.

Проверил:

преподаватель кафедры

«Вычислительная техника»

Исхаков И.И.

Ульяновск, 2023

1. Задание по варианту

Выполнить обработку элементов прямоугольной матрицы A, имеющей N строк и M столбцов. Добавить к элементам каждого столбца такой новый элемент, чтобы сумма положительных элементов стала бы равна модулю суммы отрицательных элементов. Результат оформить в виде матрицы из N + 1 строк и M столбцов.

1. Описание реализации

Для решения данной задачи понадобилось подключить библиотеку numpy. Было реализовано пять функций. Ввод элементов матрицы реализован с помощью random и randint.

1. Описание возникших затруднений.

В данной лабораторной работе возникли трудности с декомпозицией программы на функции.

1. Описание альтернативных способов решения

Альтернативных решений не нашлось.

1. **Исходный код.**

from random import randint  
import re  
import os  
import numpy as np  
  
  
def prepare\_matrix():  
 *'''генерирует случайную матрицу'''* n = randint(1, 10)  
 m = randint(1, 10)  
 print(f'Размер матрицы: {m}x{n}')  
 matrix = np.array([[randint(-10, 10) for j in range(m)] for i in range(n)])  
 return matrix  
  
  
def get\_sums(matrix):  
 *'''получает суммы положительных и отрицательных чисел'''* print("Исходная матрица")  
 print(np.array2string(matrix))  
 print("#" \* 20)  
 negatives = np.abs(np.where(matrix < 0, matrix, 0).sum(0))  
 print(f'отрицательные: {negatives}')  
 positives = np.where(matrix > 0, matrix, 0).sum(0)  
 print(f'положительные: {positives}')  
 return negatives, positives  
  
  
def calculate\_diffs(negatives, positives):  
 *'''вычисляет, сколько необходимо добавить к столбцам'''* diffs = []  
 for i in range(len(negatives)):  
 diff = negatives[i] - positives[i]  
 diffs.append(diff)  
 print(f'разницы: {diffs}')  
 return diffs  
  
  
def write\_result(result\_str):  
 *'''записывает выходную строку в файл'''* output\_str = re.sub("[\[\]]", " ", result\_str)  
 with open("output.txt", "w+") as f:  
 f.write(output\_str)  
  
  
def main():  
 matrix = prepare\_matrix()  
 # сохраняем исходную матрицу для вывода в файл  
 source\_matrix\_str = np.array2string(matrix)  
 negatives, positives = get\_sums(matrix)  
 diffs = calculate\_diffs(negatives, positives)  
 # добавляем строку с вычисленными элементами к матрице  
 matrix = np.vstack([matrix, diffs])  
 # следующие 2 строчки только для наглядного вывода, не обязательны  
 negatives, positives = get\_sums(matrix)  
 diffs = calculate\_diffs(negatives, positives)  
 processed\_matrix\_str = np.array2string(matrix)  
 result\_str = source\_matrix\_str + os.linesep \* 5 + processed\_matrix\_str  
 result\_str = re.sub('[\[\]]', "", result\_str)  
 write\_result(result\_str)  
  
if \_\_name\_\_ == '\_\_main\_\_':  
 main()