Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение

высшего образования

«Ульяновский государственный технический университет»

Кафедра: «Вычислительная техника»

Лабораторная работа №2

Дисциплина: «Разработка профессиональных приложений»

«Двумерные структуры данных»

Вариант 22

Выполнил

Студент группы ИВТАСбд-22

Хабибуллин Х. Р.

Проверил

преподаватель кафедры

«Вычислительная техника»

Исхаков И. И.

Ульяновск, 2023

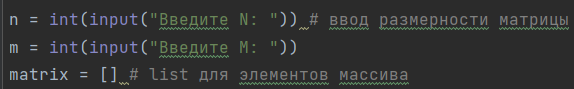
**Задание по варианту**

Выполнить обработку элементов прямоугольной матрицы A, имеющей N строк и M столбцов. Исходная матрица состоит из нулей и единиц. Добавить к матрице еще один столбец, каждый элемент которого делает количество единиц в каждой строке чётным.

**Описание реализации**

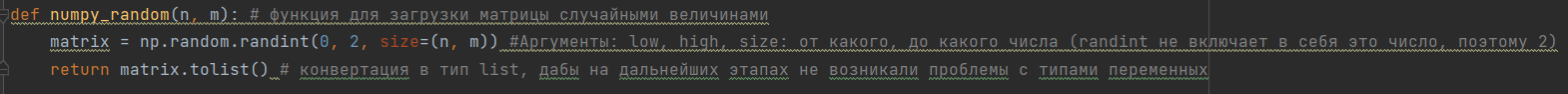
Для задания размеров матрицы, значения столбца *m* и элементов матрицы используем библиотеку *random*. Матрицу создадим с помощь библиотеки *numpy*.

*Листинг 1. Исходные данные*



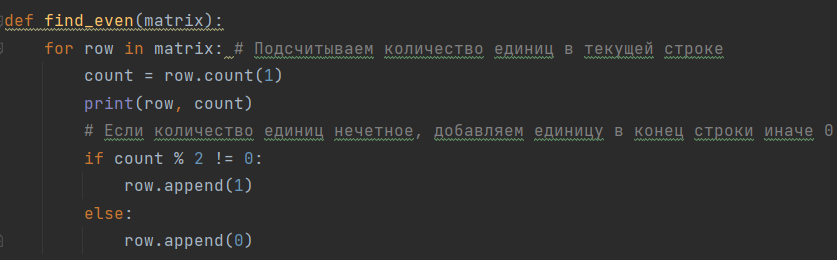
Для записи исходных значений в файл используем функцию *open*, с помощью которой открываем файл и метод *write* для записи. Для закрытия файла используем метод *close*.

*Листинг 2. генерация исходных данных*

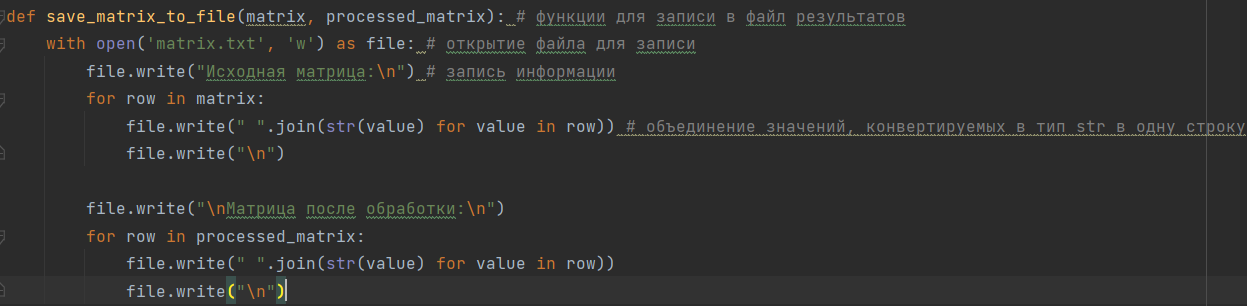


Для добавления столбца с нужным кол-во единиц, нужно найти строчки в которых кол-во единиц нечётное количество. Это задачу решает функция find\_even

*Листинг 3. реализация функции find\_even*



*Листинг 4. Запись результата*

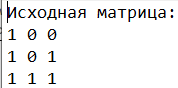


**Описание возникших затруднений**

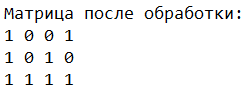
При выполнении работы затруднений не возникло.

**Тесты**

Тест. Входные данные



Тест. Выходные данные



**Выводы о проделанной работе**

Выполняя данную работу, мы научились работать с двумерными списками (матрицами) и изменять их.

**Приложение**

Полный код программы:

import copy

import random

import numpy as np

n = int(input("Введите N: ")) # ввод размерности матрицы

m = int(input("Введите M: "))

matrix = [] # list для элементов массива

def no\_numpy\_random(matrix):

for \_ in range(n):

row = []

for \_ in range(m):

value = random.randint(0, 1)

row.append(value)

matrix.append(row)

def numpy\_random(n, m): # функция для загрузки матрицы случайными величинами

matrix = np.random.randint(0, 2, size=(n, m)) #Аргументы: low, high, size: от какого, до какого числа (randint не включает в себя это число, поэтому 2)

return matrix.tolist() # конвертация в тип list, дабы на дальнейших этапах не возникали проблемы с типами переменных

def find\_even(matrix):

for row in matrix: # Подсчитываем количество единиц в текущей строке

count = row.count(1)

print(row, count)

# Если количество единиц нечетное, добавляем единицу в конец строки иначе 0

if count % 2 != 0:

row.append(1)

else:

row.append(0)

def print\_matrix(matrix): #функция для отображения в консоле матрицы в виде таблицы

for row in matrix: # цикл строки в матрице

for value in row: # цикл значений в строке

print(value, end=" ")

print()

def save\_matrix\_to\_file(matrix, processed\_matrix): # функции для записи в файл результатов

with open('matrix.txt', 'w') as file: # открытие файла для записи

file.write("Исходная матрица:\n") # запись информации

for row in matrix:

file.write(" ".join(str(value) for value in row)) # объединение значений, конвертируемых в тип str в одну строку

file.write("\n")

file.write("\nМатрица после обработки:\n")

for row in processed\_matrix:

file.write(" ".join(str(value) for value in row))

file.write("\n")

matrix = numpy\_random(n, m) # загрузка в матрицу случайных значений

print\_matrix(matrix) # вывод матрицы

res\_matrix = copy.deepcopy(matrix) #изменения, внесенные в `res\_matrix`, не будут влиять на `matrix` c помощью модуля copy и функции deepcopy

find\_even(res\_matrix) # вызов функции по анализу матрицы на предмет чётных кол-во 1

print\_matrix(res\_matrix)

save\_matrix\_to\_file(matrix, res\_matrix) # запись результатов в файл