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 № 14

Выполнил:

студент группы ИВТАПбд-21

Мухаметзянов Т. А.

Проверил:

преподаватель кафедры

«Вычислительная техника»

Исхаков И.И.

Ульяновск, 2023

**Задание по варианту**

Необходимо реализовать 14 вариант задания, а именно: обработать матрицу по варианту. Выполнить обработку элементов прямоугольной матрицы A, имеющей N строк и M столбцов. Разделить элементы матрицы на элемент матрицы с наибольшим значением

**Требования к программе:**

1. Ввод элементов списка должен быть доступен путем автоматической генерации. Необходимо использовать библиотеку numpy. Результаты выполнения должны сохраняться в файл (исходные данные и результат обработки).
2. Исходный код должен быть откомментирован
3. Необходимо реализовать правильную декомпозицию программы на методы.

**Описание реализации**

**Функция обработки матрицы:**

def matrixVariable():  
 matrix = np.random.randint(int(input("Enter low number: ")), int(input("Enter high number: ")),  
 size=(int(input("Enter N: ")), int(input("Enter M: "))))  
 # maxNumber in matrix  
 maxMatrix = numpy.amax(matrix)  
 # divide matrix[i] on maxNumber  
 newMatrix = np.divide(matrix, maxMatrix)  
 # save newMatrix and initial matrix  
 print('Initial matrix:', matrix, 'MaximumEl:', maxMatrix, sep='\n')  
 np.round(newMatrix, 2)  
 print(saveTxt(matrix, newMatrix))  
 saveNpz(matrix, newMatrix)

С помощью random.randint задаем интервал целочисленных значений нашей будущей матрицы, также указывается кол-во столбцов и строк матрицы.

Нахождение максимума матрицы с помощью функции amax из библиотеки numpy, которая в качестве аргументов принимает матрицу.

Деление всех элементов матрицы с помощью функции divide из бибилотеки матрицы, которая в качестве аргументов принимает матрицу и делитель.

Округление элементов матрицы до двух знаков после запятой с помощью функции round, из библиотеки np, в качестве аргументов принимает матрицу и число равное кол-во знаков после запятой.

Функция сохранения в .npz файл:

def saveNpz(initial, result):  
 np.savez\_compressed('file.npz', matrix=initial, newMatrix=result, fmt="%.2f")  
 finalRes = np.load('file.npz')  
 old = finalRes['matrix']  
 new = finalRes['newMatrix']  
 np.round(new, 2)  
 print("\nOld:", old, "New:", new, sep="\n")

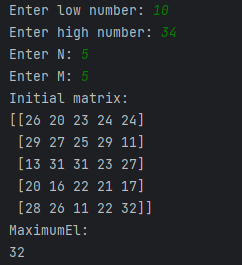
Функция сохранения в .txt файл:

def saveTxt(initial, result):  
 numpy.savetxt("file1.txt", initial, fmt="%.1d")  
 numpy.savetxt("file2.txt", result, fmt="%.2f")  
 open("result.txt", "w").write(open("file1.txt", "r").read() + "\n\n" + open("file2.txt", "r").read())  
 os.remove("file1.txt")  
 os.remove("file2.txt")  
 content = numpy.loadtxt("result.txt")  
 return content

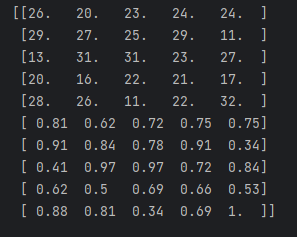
**Описание возникших затруднений**

Работа c библиотекой numpy была произведена впервые, возникли небольшие трудности из-за отсутствия русскоязычной документации, но прибегнув к возможностям интернета задача была реализована. Также возникли трудности при сохранении двух матриц (исходной и финальной), т.к сохранить 2 матрицы в .txt файл можно лишь разделив их на два файла сохранить в новый, прочитав их тех двух файлов. Поэтому реализовано 2 метода сохранения матриц в файл - .txt и .npz.

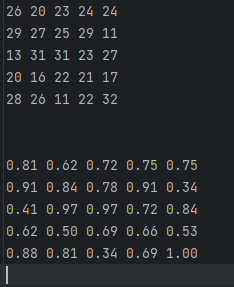
**Результаты работы программы**



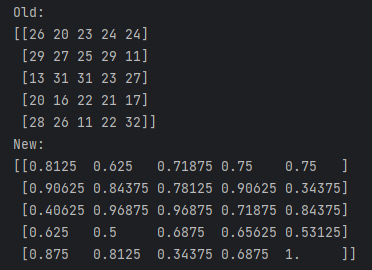
**1 рис. Ввод данных и начальные значения**



**2 рис. Вывод двух матриц в одной матрице из .txt файла**



**3 рис. Матрицы в файле**



**4 рис. Вывод двух матриц из .npz файла**

**Код программы**

import os  
import numpy  
import numpy as np  
  
  
def saveTxt(initial, result):  
 numpy.savetxt("file1.txt", initial, fmt="%.1d")  
 numpy.savetxt("file2.txt", result, fmt="%.2f")  
 open("result.txt", "w").write(open("file1.txt", "r").read() + "\n\n" + open("file2.txt", "r").read())  
 os.remove("file1.txt")  
 os.remove("file2.txt")  
 content = numpy.loadtxt("result.txt")  
 return content  
  
  
def saveNpz(initial, result):  
 np.savez\_compressed('file.npz', matrix=initial, newMatrix=result, fmt="%.2f")  
 finalRes = np.load('file.npz')  
 old = finalRes['matrix']  
 new = finalRes['newMatrix']  
 np.round(new, 2)  
 print("\nOld:", old, "New:", new, sep="\n")  
  
  
def matrixVariable():  
 matrix = np.random.randint(int(input("Enter low number: ")), int(input("Enter high number: ")),  
 size=(int(input("Enter N: ")), int(input("Enter M: "))))  
 # maxNumber in matrix  
 maxMatrix = numpy.amax(matrix)  
 # divide matrix[i] on maxNumber  
 newMatrix = np.divide(matrix, maxMatrix)  
 # save newMatrix and initial matrix  
 print('Initial matrix:', matrix, 'MaximumEl:', maxMatrix, sep='\n')  
 np.round(newMatrix, 2)  
 print(saveTxt(matrix, newMatrix))  
 saveNpz(matrix, newMatrix)  
  
  
try:  
 matrixVariable()  
except Exception as e:  
 print(e)