|  |  |  |
| --- | --- | --- |
|  | |  | | --- | | **Министерство образования и науки Российской Федерации**  **Федеральное государственное бюджетное образовательное учреждение**  **высшего образования**  **«Московский государственный технический университет**  **имени Н.Э. Баумана**  **(национальный исследовательский университет)»**  **(МГТУ им. Н.Э. Баумана)** | |

ФАКУЛЬТЕТ Информатика и системы управления

КАФЕДРА Программное обеспечение ЭВМ и информационные технологии

ОТЧЕТ ПО УЧЕБНО-ВЫЧИСЛИТЕЛЬНОМУ

ПРАКТИКУМУ

### Студент Мансуров Владислав Михайлович

*фамилия, имя, отчество*

### Группа ИУ7-16Б

Тип практики Учебная

### Название

предприятия

|  |  |  |
| --- | --- | --- |
| Студент |  | Мансуров В.М. |
| Руководитель практики | *подпись, дата* | *фамилия, и.о.*  Кузнецова О. В. |
|  | *подпись, дата* | *фамилия, и.о.* |

### Оценка

*2020 г.*

***Содержание***

### Условие задачи 3

### Схема программы… 5

### Описание программы 10

### Текст программы… 16

### Заключение… 18

### Список литературы 18

***Условия задачи***

*«Распределение участка»*

*(Время: 1 сек. Память: 16 Мб)*

В одном маленьком городке начинают работать n крупных компаний. Для начала они хотят поделить между собой n земельных участков. По расчетам экономистов, компания ***i*** может получить с участка ***j*** доход ***a i j*** . Каждая компания, разумеется, хочет получить наибольший доход.

Распределением участков занимается лично мэр города. Распределение происходит следующим образом. Каждая компания сообщает мэру, какой участок ей бы хотелось получить. После этого, если на участок претендует несколько компаний, то он отдается той из них, которая может получить с него наибольший доход (чем больше доход, тем больше налогов), а если таких несколько, то той, у которой наименьший номер (меньшие номера, разумеется, имеют знакомые мэра).

Зная, как происходит распределение, представители компаний решили собраться вместе и решить, кто на какой участок будет претендовать. Чтобы избежать жульничества, они хотят сделать так, чтобы доход компании не мог увеличиться, если она отклонится от выбранного плана, а все остальные будут ему следовать.

Оказалось, однако, что найти такой план не так-то просто. Помогите им сделать это.

***Входные данные:***

В первой строке входного файла INPUT.TXT содержится число n (1 ≤ n ≤ 400). Далее идут n строк по n чисел. *j-е* число в *i-ой* строке - это ***a i j*** , доход i-ой компании с *j-го* участка (1 ≤ ***a i j*** ≤ 109).

***Выходные данные:***

В выходной файл OUTPUT.TXT выведите n чисел. *i-е* число - это участок, на который должна претендовать *i-я* компания. Если ответ не однозначен, выведите любой.

***Ссылка на задачу*** - <https://acmp.ru/index.asp?main=task&id_task=727>

***Пример теста программы:***

|  |  |  |
| --- | --- | --- |
| № | INPUT.TXT | OUTPUT.TXT |
| 1 | 3  5 4 4  5 2 3  2 1 2 | 1 3 2 |
| 2 | 10  11 9 4 2 3 4 5 6 9 10  3 7 4 2 3 100 5 6 9 10  20 9 4 2 3 4 5 6 9 10  3 9 4 2 3 4 5 6 9 10  2 9 45 2 3 60 5 6 9 10  3 9 4 2 3 4 5 6 9 10  1 8 3 2 3 0 5 6 9 10  3 9 4 2 56 4 5 6 9 10  3 6 2 2 3 7 5 6 9 10  9 9 1 2 3 0 5 9 11 10 | 10 6 1 2 3 8 7 5 4 9 |
| 3 | 3  5.5 4.6 4  5.56 2 9.8  8.25 1.0 2.5 | 2 3 1 |

***Блок-схема***

*Начало/Start*

Открываем файлы для чтения и записи:

inputFail **=** **open(**'INPUT.txt'**,** 'r'**)**

ouputFile **=** **open(**'OUTPUT.txt','w'**)**

Выводит сообщение об ошибке, что файла нет.

Если файл существует

*False True*

numberOfCompany = int(inputFail.readline())

Выводит сообщение об ошибке.

type(numberOfCompany) == int)

*False True*

Выводит число компаний (numberOfCompany)

for i in range

(numberOfCompany)

Создание нулевой квадратной матрицы порядка

numberOfCompany

‘\n’ in row

*True*

*False*

row = row.replace('\n', '')

arrOfRow = row.split(' ')

Выводит сообщение об ошибке.

Рассмотрение переданных из файла элементов и преобразование в integer или float и ловля ошибок или не хватает элементов

for j in range(len(arrOfRow))

*Если ошибок не*

*были найдены*

for i in range(numberOfCompany)

matrixEarningsOfLand, indexOfLand, indexOfCompany = takeLandtoCompany

(matrixEarningsOfLand)

Вызов функции

takeLandtoCompany

(matrixEarningsOfLand)

for i in range(numberOfCompany)

Создание массива с нулями длиной numberOfCompany

Вызов функции

printMatrix

(matrixEarningsOfLand)

j < len(arrOfRow)

i < numberOfCompany

*Конец/Finish*

inputFile.close()

ouputFile.close()

ouputFile.write(str(arrOfAssignedLands[i]) + ' ')

Формирование вывода номера

земли и компании.

Запись данных в файл

j < numberOfCompany

i < numberOfCompany

arrOfAssignedLands [j] = indexOfLand + 1

i == indexOfCompany

*True*

*False*

***Cхема функции***

*takeLandtoCompany*

for j in range(len(matrix))

i == indexOfCompany

Вызов функции

takeLandtoCompany

(matrixEarningsOfLand)

matrix[i][j] = -1

for i in range(n):

for i in range(len(matrix))

maxElement = matrix[i][j]

indexOfCompany = i

indexOfLand = j

matrix[i][j] > maxElement

for j in range(n):



matrix[i][j] = -1

maxElement = matrix[0][0]

indexOfCompany = 0

indexOfLand = 0

***Cхема функции***

*printMatrix*

Вызов функции

printMatrix

(matrixEarningsOfLand)

Формирование вывода каждого элемента матрицы

Вывод каждого элемента

***Описание программы***

На вход данных программа принимает текстовой файл INPUT.txt, но если не находит файл, то выводит сообщение об этом – реализую это с помощью отлова *try/except* ошибки *FileNotFoundError*.

**try:**

inputFail **=** open**(**'INPUT.txt'**,** 'r'**)**

**…**

**except** FileNotFoundError**:**

**print(**'Файл INPUT.txt не был найден.'**)**

Для начала из файла импортируется первая строка и переводится в целое число (integer) и присваивается к *nameOfCompany* (если же не целое число, то выдает сообщение - реализация той же *try/except* ошибки *ValueError*).

**try:**

inputFail **=** open**(**'INPUT.txt'**,** 'r'**)**

numberOfCompany **=** int**(**inputFail**.**readline()**)**

**print(**'Число компаний претендуюших на земли в городе:'**,**numberOfCompany**)**

**…**

**except** FileNotFoundError**:**

**print(**'Файл INPUT.txt не был найден.'**)**

**except** ValueError**:**

**print(**'В файле INPUT.txt первая линия должна быть целым числом!'**)**

Это число будет задавать размер квадратной матрицу данных.

matrixEarningsOfLand **=** [[0]\*numberOfCompany **for** i **in** **range**(numberOfCompany)]

Последующие линии программа через цикл вносит в матрицу *matrixEarningsOfLand*. Так как при взятии линий берется *'\n'* то заменяем на *''* командой *replace.* Затем с помощью команды *split* разделяет строку на отдельные по пробелам и получает массив с данными первой линии.

**for** i **in** **range**(numberOfCompany):

row = inputFail.readline()

**if** '\n' **in** row**:**

row = row.replace('\n', '')

arrOfRow = row.split(' ')

…

Затем вложенным циклом берем каждый элемент массива с данными линии и преобразует их в float или integer (если встречает string или задана прямоугольная матрица или же не хватает элемента то выводит сообщение об этом).

**for** j **in** **range**(len(arrOfRow)):

**if** **len**(arrOfRow) **!=** numberOfCompany **or** **len**(matrixEarningsOfLand) **!=** numberOfCompany**:**

x\_break **=** 1

**print(**'В файле INPUT.txt матрица данных должны быть квадратной!'**)**

**print(**'Ошибка в'**,** i**+**1**,** 'cтроке:'**,** arrOfRow**)**

**break**

**try:**

**if** float**(**arrOfRow**[**j**])** **==** int**(**float**(**arrOfRow**[**j**])):**

matrixEarningsOfLand**[**i**][**j**]** **=** int**(**float**(**arrOfRow**[**j**]))**

**else:**

matrixEarningsOfLand**[**i**][**j**]** **=** float**(**arrOfRow**[**j**])**

**except** **ValueError:**

x\_break **=** 1

**print(**'В файле INPUT.txt матрица должна состоять из цифр!'**)**

**break**

**if** x\_break **==** 1 **:**

**break**

**…**

x\_break используем для того чтобы при значении 1 программа заканчивалась а при значении 0 она продолжалось

Затем вызываем функцию printMatrix вывода:

**def** **printMatrix**(matrix):

**for** i **in** matrix**:**

**for** j **in** i**:**

**print(**'{:^6g}'**.**format(j),end**=**''**)**

**print()**

**print()**

**…**

**if** x\_break != 1**:**

printMatrix(matrixEarningsOfLand)

**…**

Затем после создания и вывода матрицы данных, она *n* – раз (*n* равно числу компаний) передается в функцию *takeLandtoCompany*, где сначала берется первый элемент матрицы и сравнивается с остальными, то есть ищет максимальный элемент и его ***aij*** - *i* присвоится к i*ndexOfCompany,* а *j* к *indexOfLand.* Через второй цикл вычеркиваем ту строку и столбец (заменяет элементы на -1) и затем передаю матрицу так повторяется пока не пройдет *n* –раз. Для упорядочения полученных данных земель присвоенных к компаниям добавляет в нулевой массив *arrOfAssignedLands* длиной в *n*

**def** **takeLandtoCompany** (matrix):

maxElement = matrix[0][0]

indexOfCompany = 0

indexOfLand = 0

**for** i **in** **range**(n):

**for** j **in** **range**(n):

**if** matrix[i][j] > maxElement:

maxElement = matrix[i][j]

indexOfCompany **=** i

indexOfLand **=** j

**for** i **in** **range**(**len**(matrix)):

**if** i **==** indexOfCompany**:**

**for** j **in** range(len(matrix)):

matrix[i][j] = -1

matrix[i][indexOfLand] = -1

**return** matrix**,** indexOfLand**,** indexOfCompany

…

arrOfAssignedLands = [0 **for** i **in** range(numberOfCompany)]

**for** i **in** **range(**numberOfCompany**):**

matrixEarningsOfLand**,** indexOfLand**,** indexOfCompany = takeLandtoCompany(matrixEarningsOfLand)

**for** i **in** **range**(numberOfCompany):

**if** i **==** indexOfCompany**:**

arrOfAssignedLands [i] **=** indexOfLand **+** 1

Формируем вывод в консоль и открываем файл OUTPUT.txt если он есть перезаписываем, если нет создаем.

**for** i **in** **range**(numberOfCompany):

**print(**'"'**,** i**+**1**,**'"-ая компания получает "'**,** arrOfAssignedLands[i]**,** '"-ую землю.'**)**

ouputFile **=** **open(**'OUTPUT.txt'**,**'w'**)**

**for** i **in** arrOfAssignedLands **:**

ouputFile.write(str(i) + ' '**)**

outputFile.close()

**Тесты программы:**

*Данные INPUT.txt:* *Данные OUTPUT.txt :*

3 1 3 2

5 4 4

5 2 3

2 1 2

*В консоли:*

Число компаний претендующих на земли в городе: 3

Матрица данных о доходе компаний на земли a\_ij,

где i - номер компании j - номер земли.

5 4 4

5 2 3

2 1 2

" 1 "-ая компания получает " 1 "-ую землю.

" 2 "-ая компания получает " 3 "-ую землю.

" 3 "-ая компания получает " 2 "-ую землю.

Данные записаны в файл OUTPUT.txt как – 1 3 2

*Данные INPUT.txt:* *Данные OUTPUT.txt :*

f Пустой

5 4 4

5 2 3

2 1 2

*В консоли:*

В файле INPUT.txt первая линия должна быть одно целое число!

*Данные INPUT.txt:* *Данные OUTPUT.txt :*

3 Пустой

f 4 4

5 2 3

2 1 a

*В консоли:*

Число компаний претендующих на земли в городе: 3

В файле INPUT.txt матрица должна состоять из цифр!

*Данные INPUT.txt:* *Данные OUTPUT.txt :*

3 Пустой

5 4 4

5 2 3

2 1

*В консоли:*

Число компаний претендующих на земли в городе: 3

В файле INPUT.txt матрица данных должны быть квадратной!

Ошибка в 3 cтроке: 2 1

*Данные INPUT.txt:* *Данные OUTPUT.txt :*

3 2 3 1

5.5 4.6 4

5.56 2 9.8

8.25 1.0 2.5

*В консоли:*

Число компаний претендующих на земли в городе: 3

Матрица данных о доходе компаний на земли a\_ij,

где i - номер компании j - номер земли.

5.5 4.6 4

5.56 2 9.8

8.25 1 2.5

" 1 "-ая компания получает " 2 "-ую землю.

" 2 "-ая компания получает " 3 "-ую землю.

" 3 "-ая компания получает " 1 "-ую землю.

Данные записаны в файл OUTPUT.txt как – 2 3 1

*Данные INPUT.txt:* *Данные OUTPUT.txt :*

3 1 2 3

0 0 0

0 0 0

0 0 0

*В консоли:*

Число компаний претендующих на земли в городе: 3

Матрица данных о доходе компаний на земли a\_ij,

где i - номер компании j - номер земли.

0 0 0

0 0 0

0 0 0

" 1 "-ая компания получает " 1 "-ую землю.

" 2 "-ая компания получает " 2 "-ую землю.

" 3 "-ая компания получает " 3 "-ую землю.

Данные записаны в файл OUTPUT.txt как - 1 2 3

***Полный текст программы***

# inputFail - открытие файла INPUT.txt для чтения

# outputFail - открытие файла OUTPUT.txt для перезаписи или создания

# numberOfCompany - количество компания взятых с первой линнии INPUT.txt

# matrixEarningsOfLand - квадратная матрица данных о заработке компания за земли

# maxElement - максимальный элемент matrixEarningsOfLand

# indexOfCompany - индекс компания от 0

# indexOfLand - индекс земель от 0

# x\_break - переменная для выхода из цикла при выполнения условия при = 1

# arrOfAssignedLands - массив упорядоченных элементов присвоенных земель по индексу компаний

# row – линия без \n

#

**def** **takeLandtoCompany** **(**matrix**):**

maxElement **=** matrix**[**0**][**0**]**

indexOfCompany **=** 0

indexOfLand **=** 0

**for** i **in** range**(**numberOfCompany**):**

**for** j **in** **range(**numberOfCompany**):**

**if** matrix**[**i**][**j**]** **>** maxElement**:**

maxElement **=** matrix**[**i**][**j**]**

indexOfCompany **=** i

indexOfLand **=** j

**for** i **in** **range(len(**matrix**)):**

**if** i **==** indexOfCompany**:**

**for** j **in** **range(len(**matrix**)):**

matrix**[**i**][**j**]** **=** -1

matrix**[**i**][**indexOfLand**]** **=** -1

**return** matrix**,** indexOfLand**,** indexOfCompany

**def** **printMatrix(**matrix**):**

**for** i **in** matrix**:**

**for** j **in** i**:**

**print(**'{:^6g}'**.**format**(**j**),**end**=**''**)**

**print()**

**print()**

**try:**

inputFail **=** **open(**'INPUT.txt'**,** 'r'**)**

outputFile **=** **open(**'OUTPUT.txt'**,**'w'**)**

numberOfCompany **=** **int(**inputFail**.**readline**())**

matrixEarningsOfLand **=** **[[**0**]\***numberOfCompany **for** i **in** range**(**numberOfCompany**)]**

x\_break **=** 0

n **=** numberOfCompany

**print(**'Число компаний претендующих на земли в городе:'**,**numberOfCompany**)**

**for** i **in** range**(**numberOfCompany**):**

row **=** inputFail**.**readline**()**

**if** '\n' **in** row**:**

row **=** row**.**replace**(**'\n'**,** ''**)**

arrOfRow **=** row**.**split**(**' '**)**

**for** j **in** **range(**len**(**arrOfRow**)):**

**if** **len(**arrOfRow**)** **!=** numberOfCompany **or** **len(**matrixEarningsOfLand**)** **!=** numberOfCompany**:**

x\_break **=** 1

**print(**'В файле INPUT.txt матрица данных должны быть квадратной!'**)**

**print(**'Ошибка в'**,** i**+**1**,** 'cтроке:'**,** arrOfRow**)**

**break**

**try:**

**if** **float(**arrOfRow**[**j**])** **==** **int(float(**arrOfRow**[**j**])):**

matrixEarningsOfLand**[**i**][**j**]** **=** **int(float(**arrOfRow**[**j**]))**

**else:**

matrixEarningsOfLand**[**i**][**j**]** **=** **float(**arrOfRow**[**j**])**

**except** ValueError**:**

x\_break **=** 1

**print(**'В файле INPUT.txt матрица должна состоять из цифр!'**)**

**break**

**if** x\_break **==** 1 **:**

**break**

**if** x\_break **!=** 1**:**

**print(**' Матрица данных о доходе компаний на земли a\_ij,\nгде i - номер компании j - номер земли.'**)**

printMatrix**(**matrixEarningsOfLand**)**

arrOfAssignedLands **=** [0 **for** i **in** **range**(numberOfCompany)] # cоздаем нулевой массив

**for** i **in** **range(**numberOfCompany**):**

matrixEarningsOfLand**,** indexOfLand**,** indexOfCompany **=** takeLandtoCompany**(**matrixEarningsOfLand**)**

**for** i **in** **range**(numberOfCompany)**:**

**if** i **==** indexOfCompany**:**

arrOfAssignedLands [i] = indexOfLand + 1

**for** i **in** arrOfAssignedLands **:**

**print(**'"'**,** i**+**1**,**'"-ая компания получает "'**,** arrOfAssignedLands[i], '"-ую землю.'**)**

outputFile.write(str(i) **+** ' ')

**print(**Данные записаны в файл OUTPUT.txt как -', \*arrOfAssignedLands**)**

**except** **ValueError:**

**print(**'В файле INPUT.txt первая линия должна быть целым числом!'**)**

**except** **FileNotFoundError:**

**print(**'Файл INPUT.txt не был найден.'**)**

outputFile.close()

inputFail.close()

*З****аключение***

В написании программы, я научился работать с файлами, читать и также записывать, вытаскивать данные с файла и проводить с ними различные операции, записывать в новый файл. Также рассматривать все возможные случае ошибок при получение файла с данными, так к примеру, если в файле ошибка, то программа выведет ошибку, здесь я применил навыки работы с try/except и вместо ошибок выводил сообщение об ошибке.

***Список литературы:***

* Конспекты лекций
* Официальный сайт Python - <https://www.python.org>
* Сайт задачи - <https://acmp.ru/>