Задание 26 (№2) Сибигатулин Рамиль https://vk.com/infmat100

Условие:

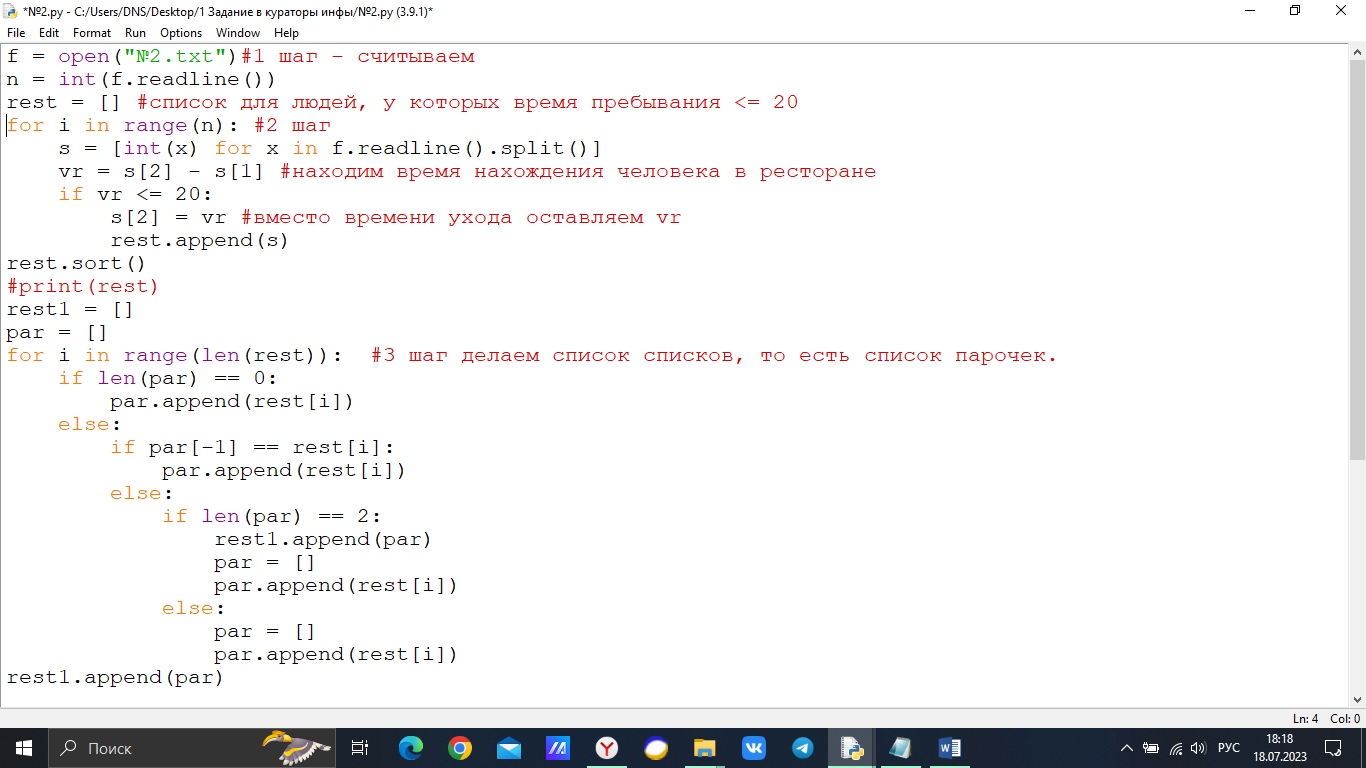
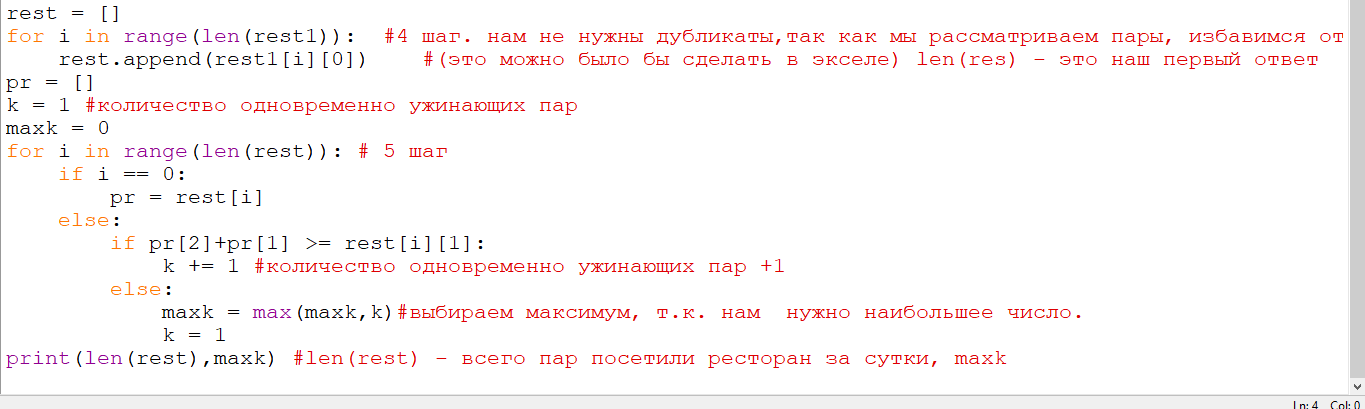
Местный Загс решил перевыполнить план и выследить своих потенциальных клиентов, ужинающих в самом престижном ресторане Клод Моне. С этой целью сотрудник Загса получил некую информацию от персонала ресторана, представленную в входном файле. Требуется определить, сколько влюбленных парочек посетили ресторан, если по наблюдениям официантов, такие всегда выдерживали максимум 20 минут, а после уходили в более уединенное место для продолжения свидания. Именно это условие исключает возможность того, ужинали родственники или просто друзья, ведь те оставались подольше. Также нужно определить максимальное количество парочек, которые были в ресторане одновременно.

В первой строке входного файла содержится N – общее число людей, посетивших ресторан, а в следующих N строк содержится по 3 числа: номер столика, время прихода от начала суток, время ухода от начала суток (оба вторых числа - время в минутах, не превышающее 1440 и не меньшее 480).

Решение: (Программа)

1. Открываем файл, считываем число N.
2. Создаём список rest и проходимся циклом по файлу, считывая каждую строчку в список s, обновляющийся на каждом шагу. У каждого посетителя заменяем время ухода на время пребывания в ресторане, для того, чтобы понять, кто находился в ресторане <= 20 и только таких посетителей добавляем в список rest. Теперь у нас есть только те посетители, которые нам могут подойти. Отсортируем наш список.
3. Но нужно проверить, сколько людей сидели за одним столиком, ведь если их было двое, то это влюбленная парочка и она нам подходит, но это может быть один, а может быть три и больше человек, которые пришли в одно время, за один столик и, просидев меньше 20 минут, просто ушли. Про это в условии ничего не написано, а значит такое может быть. Создаем список rest1, в который будем добавлять только парочек. Проходимся циклом по списку rest и добавляем посетителей с одинаковыми данными в вспомогательный список par. После проверяем длину списка par, если она == 2, то добавляем par в rest1, а иначе, просто забываем эти данные. В конце нужно не забыть, после всего цикла, добавить последний par в rest1, потому что он не пройдет по циклу до конца.
4. Теперь нужно избавиться от кучи вложенных списков и удалить из списка rest1 дубликаты. Я это делаю программой, но можно и в экселе. Я создаю список rest снова, который будет содержать данные о парах, следовательно, его длина и будет первым ответом – количество всех пар.
5. Теперь осталось найти максимальное число пар, ужинающих одновременно, для этого создаем переменную К, которая будет считать текущее количество пар в зале и на каждом шагу сравниваем с maxk, переменной, которая была создана заранее и содержит самый максимум кол-во пар. Проходимся по списку rest, запоминая прошлый элемент в переменную pr. Нужно, чтобы время прихода нового было меньше либо равно времени ухода прошлого, тогда к += 1, иначе сравнимаем с maxk и к = 1
6. Print(len(rest),maxk) – ответ.

**Код ниже:**

****

f = open("№2.txt")#1 шаг - считываем

n = int(f.readline())

rest = [] #список для людей, у которых время пребывания <= 20

for i in range(n): #2 шаг

s = [int(x) for x in f.readline().split()]

vr = s[2] - s[1] #находим время нахождения человека в ресторане

if vr <= 20:

s[2] = vr #вместо времени ухода оставляем vr

rest.append(s)

rest.sort()

#print(rest)

rest1 = []

par = []

for i in range(len(rest)): #3 шаг делаем список списков, то есть список парочек.

if len(par) == 0:

par.append(rest[i])

else:

if par[-1] == rest[i]:

par.append(rest[i])

else:

if len(par) == 2:

rest1.append(par)

par = []

par.append(rest[i])

else:

par = []

par.append(rest[i])

rest1.append(par)

rest = []

for i in range(len(rest1)): #4 шаг. нам не нужны дубликаты,так как мы рассматриваем пары, избавимся от них так.

rest.append(rest1[i][0]) #(это можно было бы сделать в экселе) len(res) - это наш первый ответ

pr = []

k = 1 #количество одновременно ужинающих пар

maxk = 0

for i in range(len(rest)): # 5 шаг

if i == 0:

pr = rest[i]

else:

if pr[2]+pr[1] >= rest[i][1]:

k += 1 #количество одновременно ужинающих пар +1

else:

maxk = max(maxk,k)#выбираем максимум, т.к. нам нужно наибольшее число.

k = 1

print(len(rest),maxk) #len(rest) - всего пар посетили ресторан за сутки, maxk