

## Оглавление

Задача 1.....	1
Условие .....	1
Решение .....	2
Пояснение к решению .....	2
Задача 2.....	3
Условие .....	3
Решение .....	4
Пояснение к решению .....	5

## Задача 1

### Условие

Костя – фанат московского Спартака. Как-то раз он пришел на известный сектор трибуны В и увидел такую картину: часть фанатов была в футболках игроков, часть была раздета по торсу. Среди фанатов в футболках Костя пошел здороваться только с теми, у которых в номере на спине присутствовала цифра 9 (любимая его цифра). В каком ряду сектора Костя поздоровался с наибольшим количеством людей в футболках?

Формат ввода:

В первой строке файла вводится количество фанатов в футболках  $N$ . В последующих  $N$  строках вводятся пары чисел – номер ряда сектора ( $<500$ ) и номер на футболке ( $<100$ ) фаната.

Формат вывода:

Требуется вывести два числа – номер ряда, в котором Костя поздоровался с наибольшим количеством людей в футболках (если есть несколько рядов с одинаковым и при этом максимальным числом таких людей, вывести любой), и количество таких людей.

Пример ввода:

5

1 9

1 91

2 18

2 20

2 69

Вывод, соответствующий данным из примера ввода:

1 2

Решение

```
1 filepath = 'C:/Users/peter/Curators/1.txt'
2 with open(filepath) as file:
3     n = int(file.readline())
4     rows = {}
5     for line in file:
6         row, number = line.split(' ')
7         if '9' in number:
8             if rows.get(row):
9                 rows[row] += 1
10            else:
11                rows[row] = 1
12    max_key = max(rows, key=lambda x: rows[x])
13    print(max_key, rows[max_key])
14
```

Пояснение к решению

Открываем файл с данными, считываем количество записей в нем. Далее создаем словарь (dictionary), в котором будем хранить данные о каждом ряде сектора, а именно количество людей с цифрой 9 в номере футболки. Словарь – структура данных вида “ключ-значение”, в которой каждому уникальному упорядоченному ключу соответствует значение. Далее построчно считываем файл, разбиваем каждую строку функцией split по пробелу на 2 части – ряд и номер на футболке. Проверяем, входит ли цифра 9 в запись номера. Если да – проверяем, встречался ли нам уже этот ряд. Если встречался – увеличиваем значение, соответствующее ряду, на 1. Если нет – создаем новый ключ в словаре со значением 1 (нужный болельщик в этом ряду встретился нам впервые). Далее с помощью функции max находим ключ, соответствующий максимальному значению. В качестве параметра поиска key мы задаем

функцию, которая по ключу "достает" значение из словаря. Это значит, что функция `max` будет искать по максимальному значению, а не ключу. В конце выводим данные в требуемом формате.

## Задача 2

### Условие

У бездельника Васи есть  $N$  мегабайт мобильного интернета. Он залипает в Youtube, при этом, соответственно, тратится интернет. Бездельник при этом старается посмотреть максимальное количество видео. Когда Вася видит, что уже не может посмотреть ни одного целого видео из предложенных, он переползает в ВК. Там он смотрит мемы в своей любимой группе, на которые тоже уходит интернет, таким же образом стараясь посмотреть максимальное их количество. Когда ВК перестает грузить мемы, Вася наконец-то начинает работать. Сколько МБ мобильного интернета у него останется для рабочих целей в лучшем случае (наибольшее кол-во оставшегося интернета)?

Формат ввода:

В первой строке вводится количество Васиного интернета в МБ  $M$  и количество предложенных Васе соцсетями единиц контента – видео и мемов  $N$ . Далее в случайном порядке вводятся данные о каждой единице контента – ее тип ("video" или "mem") и размер в МБ.

Формат вывода:

Требуется вывести 2 числа – максимально возможное количество оставшихся у Васи МБ и количество просмотренных им мемов.

Пример ввода:

```
6 700
video 401
mem 45
mem 15
video 300
mem 45
video 300
```

Вывод, соответствующий данным из примера ввода:

```
40 2
```

Пояснение к условию:

Васей были просмотрены видео 300 и 300 МБ, далее никакого видео он посмотреть не может и начинает смотреть мемы, смотрит 2 мема – 15 и 45 МБ, больше посмотреть он не может ничего, у него остается 40 МБ интернета. Ответ 10 2, получающийся при случае, когда Вася смотрит мемы 45 и 45 МБ, неверен.

Решение

```
1 filepath = 'C:/Users/peter/Curators/2.txt'
2 with open(filepath) as file:
3     n, memory = [int(s) for s in file.readline().split(' ')]
4     videos, memes = [], []
5     for line in file:
6         content, amount = line.split(' ')
7         amount = int(amount)
8         if content == 'video':
9             videos.append(amount)
10        else:
11            memes.append(amount)
12    videos.sort()
13    memes.sort()
14    curr_memory = int(memory)
15    for video in videos:
16        if (curr_memory - video) >= 0:
17            curr_memory -= video
18        else:
19            break
20    count = 0
21    for meme in memes:
22        if (curr_memory - meme) >= 0:
23            curr_memory -= meme
24            count += 1
25        else:
26            break
27    print(curr_memory, count)
28
```

#### Пояснение к решению

Открываем файл, считываем количество записей в нем и количество Васиного интернета в переменные `n` и `memory`. Создаем два пустых списка – для объемов видео и мемов. Далее построчно считываем файл, добавляя считанный объем в зависимости от типа либо в список `videos`, либо в `memes`. Сортируем оба списка. Создаем копию переменной `memory`, чтобы не потерять изначальное ее значение при изменении (сделать `curr_memory = memory` нельзя, так как в этом случае обе переменные будут ссылаться на один и тот же объект, и при изменении `curr_memory` изменится и `memory`). Далее, пока есть для этого память смотрим видео – вычитаем в порядке возрастания доступные объемы видео. Когда интернет для видео заканчивается (условие в строке 16), перестаем смотреть и выходим из цикла `for` с помощью `break`. Далее аналогично смотрим мемы, при этом считая их количество в переменной `count`. Выводим результаты подсчета. Случай, при котором нельзя просмотреть ни одного мема, решением также предусмотрен.

Пояснение к 3-й строке кода: использован генератор списков, очень удобная фишка Python. С помощью нее из списка (или любого объекта, по которому можно пройти циклом) формируется новый список по правилу, обозначенному разработчиком. В данном случае к каждой строке `s` из объекта `file.readline().split(' ')` (считанная из файла строка, разбитая по пробелу) применяется функция преобразования ее к целому числу `int()`, получается список из чисел. Далее, так как мы знаем, что в этом списке 2 числа, мы “распаковываем” его в 2 отдельные переменные – `n` и `memory`.