

**Флигинский Виктор Михайлович**

<https://vk.com/id361645911>

**Средняя задача (основано на задании 26)**

*Условие:*

Группа учёных занимается изучением поведения молекул газа в закрытом кубе. С помощью специального устройства им удалось зафиксировать положение каждой молекулы в трёхмерной системе координат. Назовём слоем совокупность молекул, имеющих одинаковые координаты по оси  $Y$ . Найдите слой с наибольшим количеством молекул, если подходящих слоёв несколько, выберите тот, у которого координата по оси  $Y$  больше. Определите в этом слое максимальное количество молекул, имеющих одинаковую координату по оси  $X$ . В качестве ответа запишите координату по оси  $Y$  у выбранного слоя и максимальное количество молекул, имеющих одинаковую координату по оси  $X$  в этом слое.

*Входные данные:*

В первой строке входного файла находится число  $N$  - количество молекул. Каждая из следующих  $N$  строк содержит три числа: координата молекулы по оси  $X$ , по оси  $Y$ , по оси  $Z$ , разделённые пробелом.

В ответе запишите два числа: сперва координату по оси  $Y$  у искомого слоя, затем максимальное количество молекул, имеющих одинаковую координату по оси  $X$  в этом слое.

*Пример:*

15

1 1 3

2 4 5

1 3 6

3 4 6

1 3 4

3 6 2

2 3 4  
2 5 1  
3 5 2  
2 5 2  
1 3 1  
2 3 2  
4 3 6  
1 4 2  
4 2 5

При таких входных данных ответ будет 3 3

*Решение:*

1. Открываем файл и считываем n

```
f = open("26.txt")  
n = int(f.readline())
```

2. создаём необходимые переменные. Data – матрица, в каждом элементе которой будут храниться координаты молекулы. Layers – словарь, в качестве ключа будет координата слоя по оси Y, в качестве значения – количество молекул на слое с такой координатой Y. ox – словарь, который будем использовать после нахождения нужного слоя, в качестве ключа будет координата по оси X, в качестве значения – количество молекул с такой координатой X в найденном слое. answer1 и answer2 – переменные для хранения ответа.

```
data = []  
layers = dict()  
ox = dict()  
answer1 = -1  
answer2 = -1
```

3. При помощи цикла из n строк файла считываем координаты каждой молекулы и добавляем их в матрицу data.

```
x, y, z = map(int, f.readline().split())
data.append([x, y, z])
```

4. В этом же цикле можем заполняем словарь layers. Если раньше не встречалась молекула с такой координатой Y (`if y not in layers.keys()`), то добавляем ключ в словарь, а значение по ключу делаем равным 1. Если же раньше встречалась молекула с такой координатой Y, то просто к значению по ключу добавляем единицу.

```
if y not in layers.keys():
    layers[y] = 1
else:
    layers[y] += 1
```

5. Находим максимальное значение среди значений словаря layers и запоминаем его в max\_k.

```
max_k = max(layers.values())
```

6. В цикле проходимся по всем ключам словаря layers и ищем, по какому ключу наибольшее значение, равное max\_k. Полученный ключ будет первым ответом. Запоминаем его в переменную answer1.

```
max_k = max(layers.values())
for key in layers.keys():
    if layers[key] == max_k:
        answer1 = max(answer1, key)
```

7. В цикле проходимся по координатам x, y, z у элементов матрицы data. Если у молекула нам подходит (равен answer1), то изменяем словарь ox. Если раньше не было

молекулы с такой координатой x, то добавляем ключ в словарь, а значение по ключу делаем равным 1. Если же раньше встречалась молекула с такой координатой x, то просто к значению по ключу y добавляем единицу.

```
for x, y, z in data:
    if y == answer1:
        if x not in ox.keys():
            ox[x] = 1
        else:
            ox[x] += 1
```

8. Вторым ответом будет максимальное значение среди значений словаря ox. Запоминаем его.

```
answer2 = max(ox.values())
```

9. Выводим ответ.

```
print(answer1, answer2)
```

Ответ: 193 2