

Wordle

На вход программы приходит:

- сначала одно слово из пяти заглавных латинских букв (загаданное),
- потом целое число - количество попыток его угадать (не более шести),
- потом сами попытки (каждая - слово из пяти букв, тоже латинских и заглавных).

Требуется вывести на экран, как выглядит табличка с результатами. Для каждой буквы в каждой из попыток угадать выводится, есть ли такая буква в загаданном слове вообще и на этом ли месте.

Выводить следующими символами:

- - (буквы нет);
- g (green, буква есть, положение правильное);
- y (yellow, буква есть, положение неправильное).

Например:

Ввод	Результат
PROSE 3 START PRIME PROSE	y--y- gg--g ggggg

Грузовики

(Полная режиссёрская версия.)

Реализуйте класс, описывающий грузовую машину. Базовый прототип класса:

```
#include <string>

class Truck {
public:
    // Конструктор:
    // - capacity - грузоподъемность
    // - speed - скорость
    // - number - номер (да, это строка)
    Truck(unsigned int capacity, unsigned int speed, const std::string& number);

    // Дополнительные методы и операторы, которые потребуются
};
```

Класс должен:

- уметь взаимодействовать с потоками вывода, стандартный оператор вывода для класса Truck должен выводить сначала номер машины, затем через пробел грузоподъемность и скорость;
- быть базово совместим с STL, должна работать стандартная сортировка вектора из Truck.

При сравнении двух машин для сортировки должна использоваться следующая логика:

- "меньше" считается та машина, у которой меньше грузоподъемность,
- при равной грузоподъемности "меньше" считается та машина, у которой меньше скорость,
- при равных грузоподъемности и скорости сравниваются номера машин по алфавиту.

Для тестирования можете использовать следующий базовый пример:

```
#include <iostream>
#include <string>
#include <vector>
#include <algorithm>

using namespace std;

int main()
{
    vector<Truck> data;
    data.push_back(Truck(10, 10, "a720po"));
    data.push_back(Truck(100, 10, "a721aa"));
    data.push_back(Truck(100, 20, "e721op"));
    data.push_back(Truck(10, 10, "o720po"));

    sort(data.begin(), data.end());

    for(Truck& c : data) {
        cout << c << endl;
    }

    return 0;
}
```

Данный пример должен вывести на экран следующее:

```
a720po 10 10
o720po 10 10
```

a721aa 100 10
e721op 100 20

Мешок котов

Напишите класс, который является мешком с котами.

Коты выглядят вот так:

```
#include <string>
```

```
struct Cat {  
    std::string name; // Имя  
    unsigned int weight; // Вес (в граммах)  
};
```

Прототип требуемого класса:

```
class BagOfCats {  
public:  
    // Конструктор и деструктор, если нужны  
  
    // Сложить кота в мешок  
    void putCat(const Cat& c);  
  
    // Распечатать статистику по содержимому мешка.  
    // Требования и пример вывода ниже.  
    void printStats() const;  
};
```

Внимание: могут быть коты с одинаковыми именами и весом.

Требования к статистике:

- вывод в `std::cout`
- сгруппировать котов по именам, каждую группу вывести в новой строке
- для каждой группы вывести через пробел (а) имя, (б) сколько котов с таким именем, (в) их общий вес
- при выводе группы сортировать (а) сначала по количеству котов по убыванию, (б) при совпадении количества -- по весу по убыванию (гарантируется, что совпадения веса групп до грамма не бывает).

Для базового тестирования можете использовать такой пример:

```
int main()  
{  
    Cat cats[] = {"Barsic", 1500}, {"Murzik", 1200}, {"Gav", 500}, {"Murzik", 1800}};  
  
    BagOfCats bag;  
  
    for(const auto& cat : cats)  
        bag.putCat(cat);  
  
    bag.printStats();  
  
    return 0;  
}
```

На экране должно получиться:

```
Murzik 2 3000  
Barsic 1 1500  
Gav 1 500
```

Наблюдение за небом

На стандартный поток ввода приходит множество точек. Каждая точка имеет координаты X и Y, а также яркость. Все значения - целые числа. Точки могут повторяться. Если у точек совпали координаты - это повтор одной и той же точки (гарантируется, что яркость при этом тоже совпадёт).

Считайте все пришедшие точки, отфильтруйте следующим образом:

- оставить все точки с яркостью больше или равной 10;
- оставить все точки, расстояние от которых до начала координат не превышает 100;
- оставить все точки, которые повторились не менее 2 раз;
- все остальные точки отбросить.

Достаточно, чтобы точка отвечала одному любому условию. То есть, например, тусклые и далёкие точки, замеченные 2 раза, оставляем.

После фильтрации выведите точки в формате "X Y яркость" (см. примеры ниже). При выводе сортировать следующим образом:

- сначала по яркости от более ярких к менее ярким;
- при совпадении яркости -- по количеству повторов от более частых к более редким;
- при совпадении яркости и количества повторов -- по расстоянию от более близких к началу координат к более далёким.

Формат ввода

В отдельной строке целое число N - количество точек. Затем N строк, в каждой через пробел X, Y, яркость.

Формат вывода

Каждая точка в отдельной строке, в формате "X Y яркость". Выводить без повторов.

Например:

Ввод	Результат
7 100 200 50 5 5 25 10 10 1 100 300 1 100 300 1 150 150 5 20 20 1	100 200 50 5 5 25 100 300 1 10 10 1 20 20 1