#include<iostream>

#include<string>

#include<fstream>

using namespace std;

class train{

private:

int number; // номер потяга

string station; // станція прибуття

double price; // ціна квитка

int count; // кількість проданих квитків

public:

// конструктор за замувчуванням/параметрами

train(int number = 0, string station = "unknown", double price = 0, int count = 0) {

this->number = number;

this->station = station;

this->price = price;

this->count = count;

}

// конструктор копіювання

train(const train& other) {

this->number = other.number;

this->station = other.station;

this->price = other.price;

this->count = other.count;

}

/////////////

//перегрузка вводу

friend istream& operator >>(istream& is, train& other) {

is >> other.number >> other.station >> other.price >> other.count;

return is;

}

//перегрузка виводу

friend ostream& operator <<(ostream& out, train& other) {

out << "Name: " << other.number << endl <<

"Maker: " << other.station << endl <<

"Price: " << other.price << endl <<

"Count: " << other.count << endl <<

"General price " << other.generalPrice() << endl << endl;

return out;

}

//перегрузка < за кількістю проданих квитків

bool operator <(const train& other) {

return this->count < other.count;

}

//перегрузка -(зменшення вартості квитка на вказану кількість)

void operator -(int n) {

this->price -= n;

}

/////////////

// метод вираховування загальної вартості квитків

double generalPrice() {

return price \* count;

}

friend train minValue(train\* arr, int n);

friend void sort(train\* arr, int n);

friend void showInfo(train\* arr, int n, string station);

};

// знаходження потягу з мінімальною вартістю квитка

train minValue(train\* arr, int n) {

train min = arr[0];

for (int i = 1; i < n; i++) {

if (arr[i].price < min.price)

min = arr[i];

}

return min;

}

// сортування за загальною виручкою

void sort(train\* arr, int n) {

train temp;

for (int i = 0; i < n; i++) {

for (int j = 0; j < n - i - 1; j++) {

if (arr[j].generalPrice() > arr[j + 1].generalPrice()) {

temp = arr[j];

arr[j] = arr[j + 1];

arr[j + 1] = temp;

}

}

}

}

//виведення у файл інформації про всі потяги що прямують до заданої станції

void showInfo(train\* arr, int n, string station) {

for (int i = 0; i < n; i++) {

if (arr[i].station == station) {

cout << arr[i];

}

}

}

void main() {

ifstream data;

data.open("data.txt");

int n;

data >> n;

train\* arr = new train[n];

int number;

string station;

double price;

int count;

for (int i = 0; i < n; i++) {

data >> number >> station >> price >> count;

train something(number, station, price, count);

arr[i] = something;

}

//cout << arr[0];

//arr[0] - 5;

//cout << arr[0];

//cout << arr[0];

//cout << arr[1];

//bool result = arr[0] < arr[1];

//cout << result;

//sort(arr, n);

//for (int i = 0; i < n; i++) {

// cout << arr[i];

//}

//cout << "info about train to lviv: " << endl;

//showInfo(arr, n, "lviv");

//train min = minValue(arr, n);

//cout << "train with minimal price:" << endl << min << endl;

//cout << arr[0];

//cin >> arr[0];

//cout << arr[0];

delete[] arr;

}