#include<iostream>

#include<string>

#include<cassert>

#include<fstream>

using namespace std;

class good {

private:

string name;

string maker;

double price;

int count;

public:

good(string name = "unnamed", string maker = "unknown", double price = 0, int count = 0) {

this->name = name;

this->maker = maker;

this->price = price;

this->count = count;

}

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

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

is >> other.name >> other.maker >> other.price >> other.count;

return is;

}

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

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

"Maker: " << other.maker << endl <<

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

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

"General price: " << other.price \* other.count << endl << endl;

return out;

}

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

friend good maxValue(good\* arr, int n);

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

};

good maxValue(good\* arr, int n) {

good max = arr[0];

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

if (arr[i].price > max.price)

max = arr[i];

}

return max;

}

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

good temp;

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

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

if (arr[j].maker == arr[j + 1].maker) {

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

temp = arr[j];

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

arr[j + 1] = temp;

}

}

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

temp = arr[j];

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

arr[j + 1] = temp;

}

}

}

}

void main() {

ifstream data;

data.open("data.txt");

int n;

data >> n;

good\* arr = new good[n];

string name, maker;

double price;

int count;

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

data >> name >> maker >> price >> count;

good something(name, maker, price, count);

arr[i] = something;

}

sort(arr, n);

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

cout << arr[i];

}

good max = maxValue(arr, n);

cout << max << endl;

}