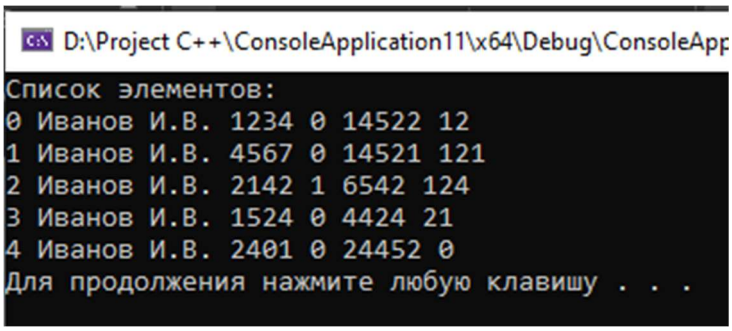
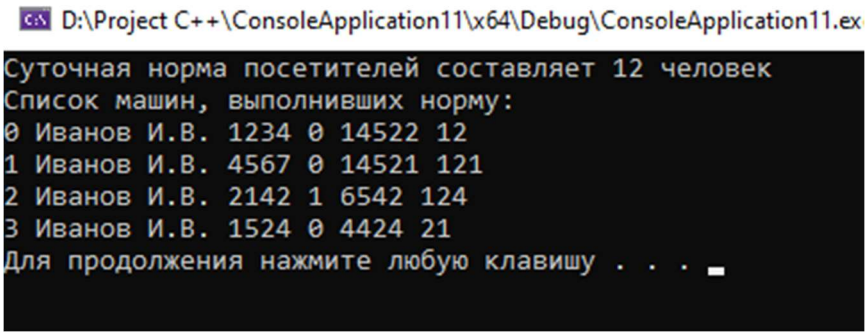
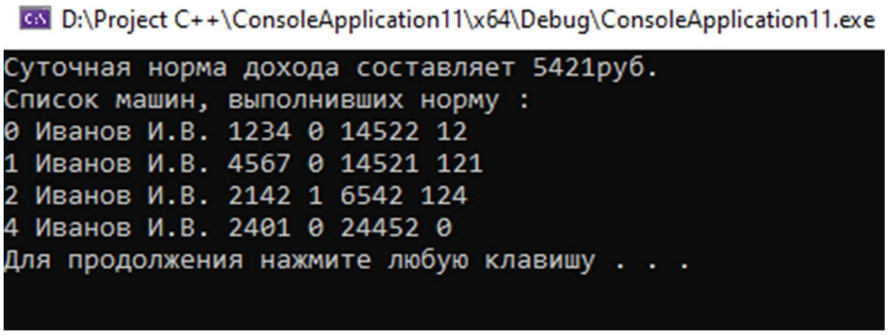
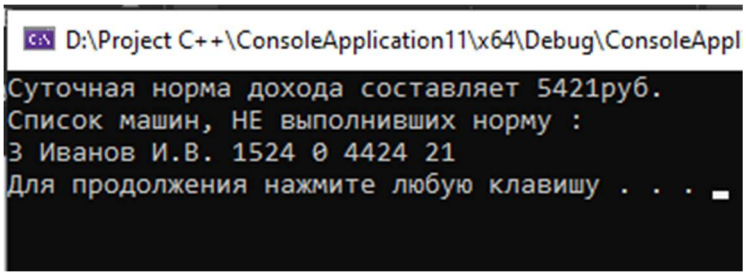
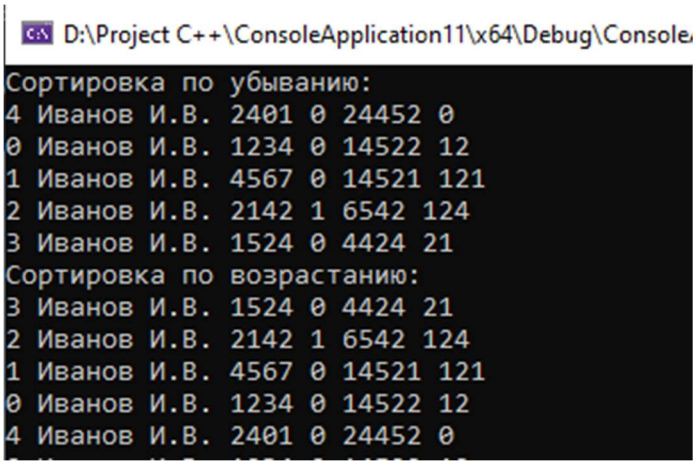

D:\Project C++\ConsoleApplication11\x64\Debug\ConsoleApp
Список элементов:
0 Иванов И.В. 1234 0 14522 12
1 Иванов И.В. 4567 0 14521 121
2 Иванов И.В. 2142 1 6542 124
3 Иванов И.В. 1524 0 4424 21
4 Иванов И.В. 2401 0 24452 0
Для продолжения нажмите любую клавишу . . .


D:\Project C++\ConsoleApplication11\x64\Debug\ConsoleApplication11.ex
Суточная норма посетителей составляет 12 человек
Список машин, выполнивших норму:
0 Иванов И.В. 1234 0 14522 12
1 Иванов И.В. 4567 0 14521 121
2 Иванов И.В. 2142 1 6542 124
3 Иванов И.В. 1524 0 4424 21
Для продолжения нажмите любую клавишу . . .


D:\Project C++\ConsoleApplication11\x64\Debug\ConsoleApplication11.exe
Суточная норма дохода составляет 5421руб.
Список машин, выполнивших норму :
0 Иванов И.В. 1234 0 14522 12
1 Иванов И.В. 4567 0 14521 121
2 Иванов И.В. 2142 1 6542 124
4 Иванов И.В. 2401 0 24452 0
Для продолжения нажмите любую клавишу . . .


D:\Project C++\ConsoleApplication11\x64\Debug\ConsoleAppl
Суточная норма дохода составляет 5421руб.
Список машин, НЕ выполнивших норму :
3 Иванов И.В. 1524 0 4424 21
Для продолжения нажмите любую клавишу . . .


D:\Project C++\ConsoleApplication11\x64\Debug\Console
Сортировка по убыванию:
4 Иванов И.В. 2401 0 24452 0
0 Иванов И.В. 1234 0 14522 12
1 Иванов И.В. 4567 0 14521 121
2 Иванов И.В. 2142 1 6542 124
3 Иванов И.В. 1524 0 4424 21
Сортировка по возрастанию:
3 Иванов И.В. 1524 0 4424 21
2 Иванов И.В. 2142 1 6542 124
1 Иванов И.В. 4567 0 14521 121
0 Иванов И.В. 1234 0 14522 12
4 Иванов И.В. 2401 0 24452 0

```
0 Иванов И.В. 1234 0 14522 12
1 Иванов И.В. 4567 0 14521 121
2 Иванов И.В. 2142 1 6542 124
3 Иванов И.В. 1524 0 4424 21
4 Иванов И.В. 2401 0 24452 0
Введите id записи которую нужно удалить >> 4
Удаление прошло успешно!
Для продолжения нажмите любую клавишу . . .
```

```
D:\Project C++\ConsoleApplication11\x64\Debug\ConsoleAp
Список элементов:
0 Иванов И.В. 1234 0 14522 12
1 Иванов И.В. 4567 0 14521 121
2 Иванов И.В. 2142 1 6542 124
3 Иванов И.В. 1524 0 4424 21
Для продолжения нажмите любую клавишу . . .
```

```cpp
#include <iostream>
#include <string>
#include <Windows.h>
#include <vector>
#include <algorithm>
#include <map>

using namespace std;

const int dailyRateVisitors = 12; // Суточная норма по кол-ву посетителей
const int minProfit = 5421; // Минимальна прибыль

struct Model {
    string id;
    string name;        // имя ответственного
    string number;      // номер машины
    string OnOff;       // свободна
    string profit;      // доход с места
    string numberUsers; // количество пользователей за сутки
};

class Controller {
private:
    map <int, Model> modelMap;
    map <int, Model>::iterator it;
    int sizeMap = -1;

    void ShowModel(Model _model) {
        cout << _model.id << " ";
        cout << _model.name << " ";
        cout << _model.number << " ";
        cout << _model.OnOff << " ";
        cout << _model.profit << " ";
        cout << _model.numberUsers << "\n";
    }

public:
    Controller() { }

    void Add() {
        cout << "Заполните поля игровой станции:\n";
        Model newModel; newModel.id = to_string(sizeMap + 1);
```

```
        cout << "id >> " << newModel.id << "\n";
        cout << "name >> "; getline(cin, newModel.name);
        cout << "number >> "; getline(cin, newModel.number);
        cout << "OnOff >> "; getline(cin, newModel.OnOff);
        cout << "profit >> "; getline(cin, newModel.profit);
        cout << "numberUsers >> "; getline(cin, newModel.numberUsers);

        modelMap[++sizeMap] = newModel;

        cout << "\nДобавление нового элемента прошло успешно!\n";
        system("pause");
        system("cls");
    }

    void Show() {
        it = modelMap.begin();
        cout << "Список элементов:" << endl;
        while (it != modelMap.end()) {
            ShowModel(it->second);
            it++;
        }
        system("pause");
        system("cls");
    }

    void Delete() {
        if (sizeMap == -1) { cout << "Список пуст\n\n"; return; }
        it = modelMap.begin();
        while (it != modelMap.end()) { ShowModel(it->second); it++; }

        cout << "Введите id записи которую нужно удалить >> ";
        int choice; cin >> choice; cin.clear(); // cin.ignore();

        it = modelMap.find(choice);
        if (it == modelMap.end()) { cout << "Нет записи с таким ID!\n"; return; }
        modelMap.erase(it);
        cout << "Удаление прошло успешно!\n";

        system("pause");
        system("cls");
    }

    void NormCompleted() {
        if (sizeMap == -1) { cout << "Список пуст\n\n"; return; }

        cout << "Суточная норма посетителей составляет " << dailyRateVisitors << "
человек\nСписок машин, выполнивших норму:\n";
        bool errorNotFound = true;
        it = modelMap.begin();
        while (it != modelMap.end()) {
            if (stoi(it->second.numberUsers) >= dailyRateVisitors) {
                ShowModel(it->second);
                errorNotFound = false;
            }
            it++;
        } if (errorNotFound) {
            cout << "Ни одна машина не выполнила суточную норму\n";
        }
        system("pause");
        system("cls");
    }

    void operator()(bool even) {
        if (even) {
            cout << "Суточная норма дохода составляет " << minProfit << "руб.\n";
```

```
            bool errorNotFound = true;
            cout << "Список машин, выполнивших норму : \n";
            it = modelMap.begin();
            while (it != modelMap.end()) {
                if (stoi(it->second.profit) >= minProfit) {
                    ShowModel(it->second);
                    errorNotFound = false;
                }
                it++;
            } if (errorNotFound) { cout << "Отсутсмвуют машины, выполнившие
норму\n"; }
        } else {
            cout << "Суточная норма дохода составляет " << minProfit << "руб.\n";
            bool errorNotFound = true;
            cout << "Список машин, НЕ выполнивших норму : \n";
            it = modelMap.begin();
            while (it != modelMap.end()) {
                if (stoi(it->second.profit) < minProfit) {
                    ShowModel(it->second);
                    errorNotFound = false;
                }
                it++;
            } if (errorNotFound) { cout << "Все машины выполнили суточную норму\n";
}
        }
        system("pause");
        system("cls");
    }

    void SortVector() {
        if (sizeMap == -1) { cout << "Список пуст\n\n"; return; }
        vector <Model> model;

        it = modelMap.begin();
        while (it != modelMap.end()) { model.push_back(it->second);  it++; }

        cout << "Сортировка по убыванию:\n";
        sort(model.begin(), model.end(), [](Model a, Model b) { return
stoi(a.profit) > stoi(b.profit); }); // Убывание
        for (auto i : model) { ShowModel(i); }

        cout << "\nСортировка по возрастанию:\n";
        sort(model.begin(), model.end(), [](Model a, Model b) { return
stoi(a.profit) < stoi(b.profit); }); // Возрастание
        for (auto i : model) { ShowModel(i); }
        system("pause");
        system("cls");
    }
};

void startData(Controller& _controller) {
    _controller.Add();
    _controller.Add();
    _controller.Add();
    _controller.Add();
    _controller.Add();
}

int main() {
    SetConsoleCP(1251);
    SetConsoleOutputCP(1251);

    Controller _controller;
    startData(_controller); // добавить запись
```

```
    _controller.Show();
    _controller.NormCompleted();

    _controller(true);  // найти характеристики по значению больше заданной
    _controller(false); // найти характеристики по значению меньше заданной

    _controller.SortVector();
    _controller.Delete();
    _controller.Show();
}
```