

```
1  #include <iostream>
2  #include <vector>
3  #include <string>
4  #include <iomanip>
5  #include <algorithm>
6  #include <fstream>
7  #include <map>
8
9  using namespace std;
10
11 class Duration {
12 private:
13     int hour;
14     int min;
15
16 public:
17     Duration(int hour_ = 0, int min_ = 0) {
18         int total = hour_ * 60 + min_;
19         hour = total / 60;
20         min = total % 60;
21     }
22
23     int Hour() const {
24         return hour;
25     }
26
27     int Min() const {
28         return min;
29     }
30 };
31
32 bool operator < (const Duration& lhs, const Duration& rhs) {
33     if (lhs.Hour() == rhs.Hour()) {
34         return lhs.Min() < rhs.Min();
35     }
36     return lhs.Hour() < rhs.Hour();
37 }
38
39 ostream& operator<<(ostream& out, const Duration& dur) {
40     out << setfill('0');
41     out << setw(2) << dur.Hour() << ':'
42         << setw(2) << dur.Min();
43     return out;
44 }
45
46 istream& operator>>(istream& in, Duration& dur) {
47     int hour;
48     in >> hour;
49     in.ignore(1);
50     int min;
51     in >> min;
52     dur = Duration(hour, min);
53     return in;
54 }
55
56 Duration operator + (const Duration& lhs, const Duration& rhs) {
```

```
57     return Duration {lhs.Hour() + rhs.Hour(), lhs.Min() + rhs.Min()};
58 }
59
60
61 int main() {
62     ifstream input("runners.txt");
63     Duration worst;
64     map<Duration, string> all;
65     if (input) {
66         Duration dur;
67         string name;
68         while(input >> dur >> name) {
69             if (worst < dur) {
70                 worst = dur;
71             }
72             all[dur] += (name + " ");
73         }
74     }
75
76     ofstream out("result.txt");
77     for (const auto durationNames: all) {
78         out << durationNames.first << '\t' << durationNames.second << endl;
79     }
80     cout << "Worst runner: " << all[worst] << endl;
81 }
```