

```
1: import java.io.BufferedReader;
2: import java.io.InputStreamReader;
3: import java.util.*;
4: import java.util.stream.Collectors;
5:
6: /**
7:  * Created by tym12 on 10/22/16.
8:  */
9: public class Samples {
10:
11:     static class Data {
12:         int time;
13:         int value;
14:     }
15:
16:     static class Instruction {
17:         String op;
18:         String func;
19:         int time;
20:     }
21:
22:     public static void main(String[] args) {
23:
24:         //Scanner scanner = new Scanner(System.in);
25:
26:         // write your code here
27:
28:         BufferedReader bufferedReader = new BufferedReader(new InputStreamReader(System.in));
29:         String nrS;
30:         try {
31:             do {
32:                 nrS= bufferedReader.readLine();
33:                 //nrS = scanner.nextLine();
34:                 int n = Integer.parseInt(nrS);
35:
36:                 // >>> data
37:                 ArrayList<Data> data = new ArrayList<>(n);
38:                 for (int i = 0; i <n ; i++) {
39:                     String text = bufferedReader.readLine();
40:                     //String text = scanner.nextLine();
41:
42:                     String[] split = text.split(" ");
43:                     Data next = new Data(){
44:                         time = Integer.parseInt(split[0]);
45:                         value = Integer.parseInt(split[1]);
46:                     };
47:
48:                     data.add(next);
```

```
49:         }
50:
51:         // >>> instrukce
52:         nrS= bufferedReader.readLine();
53:         //nrS = scanner.nextLine();
54:         int in = Integer.parseInt(nrS);
55:         for (int i = 0; i <n ; i++) {
56:             String text = bufferedReader.readLine();
57:             //String text = scanner.nextLine();
58:
59:             String[] split = text.split(" ");
60:             Instruction instruction = new Instruction(){
61:                 op = split[0];
62:                 func = split[1];
63:                 time = Integer.parseInt(split[2]);
64:             };
65:
66:             process(data, instruction);
67:         }
68:
69:     }
70:     while(nrS != null);
71: } catch (Exception e) {
72:     //
73: }
74:
75: }
76:
77: private static void process(List<Data> data, Instruction i) {
78:     int matches = 0;
79:
80:     boolean first = true;
81:     for (Data next : data) {
82:         if (first) {
83:             first = false;
84:             continue;
85:         }
86:
87:         IntSummaryStatistics stats = data.stream()
88:             .filter(d -> d.time < next.time)
89:             .filter(d -> (next.time - d.time) < i.time)
90:             .mapToInt(d -> d.value)
91:             .summaryStatistics();
92:
93:         double val = 0;
94:         if (i.func.equals("min")) {
95:             val = stats.getMin();
96:         }
```

```
97:         } else if (i.op.equals("max")) {
98:             val = stats.getMax();
99:         } else {
100:             val = stats.getAverage();
101:         }
102:
103:         if (i.op.equals("lt")) {
104:             if (next.value < val)
105:                 matches++;
106:         } else if (i.op.equals("gt")) {
107:             if (next.value > val)
108:                 matches++;
109:         }
110:
111:     }
112:     System.out.println(matches);
113: }
114:
115: private static Set<Character> toSet(String text) {
116:     return text.chars().mapToObj(c -> (char) c).collect(Collectors.toSet());
117: }
118:
119: }
120:
```