

```

1  #include <cstdio>
2  #include <vector>
3  #include <algorithm>
4  #include <climits>
5
6  #define abs(a) ((a) > 0)? (a) : -(a)
7  #define max(a, b) ((a > b)? (a) : (b))
8  #define min(a, b) ((a < b)? (a) : (b))
9
10 using namespace std;
11
12 class Number {
13 public:
14     int val;
15     int pos;
16
17     Number(int val, int pos) {
18         this->val = val;
19         this->pos = pos;
20     }
21 };
22
23 struct Comparator {
24     bool operator() (const Number& n1, const Number& n2) {
25         return n1.val < n2.val;
26     }
27 };
28
29 void printData(vector<Number>& vec);
30
31 int main() {
32     //printf("%d %d %d %d\n", abs(5), abs(1 - 7), max(-1, 2), max(2, -1));
33     int N, val;
34     scanf("%d", &N);
35     vector<Number> vec;
36     vec.reserve(N+10);
37     for(int i = 0; i < N; ++i) {
38         scanf("%d", &val);
39         vec.push_back(Number(val, i));
40     }
41
42     std::sort(vec.begin(), vec.end(), Comparator());
43     //printData(vec);
44     vec.push_back(Number(-2, -1)); // This is a 'sentinel' to make last element
45                                     handling easier.
46
47     int current = -1;
48     int max_i = -1;
49     int min_i = INT_MAX;
50     int max_pos = -1;
51     int min_pos = INT_MAX;
52     for(int i = 0; i < N; ++i) {
53         Number num = vec.at(i);
54         if(num.val != current) {
55             current = num.val;
56             min_i = max_i = i;
57             min_pos = max_pos = num.pos;
58         } else {
59             max_i = i;
60             min_pos = min(num.pos, min_pos);
61             max_pos = max(num.pos, max_pos);
62         }
63
64         if(vec.at(i+1).val != current) { // summarize disposition
65             int diff1 = abs(min_i - max_pos);
66             int diff2 = abs(max_i - min_pos);
67             printf("%d %d\n", current, max(diff1, diff2));
68         }
69     }
70     return 0;
71 }
72

```

```

73 void printData(vector<Number>& vec) {
74     for(Number& n: vec)
75         printf("%d ", n.val);
76     printf("\n");
77 }
78
79 /**
80 12
81 4 3 2 1 5 1 2 1 3 6 5 4
82
83 1 7
84 2 3
85 3 5
86 4 8
87 5 6
88 6 2
89
90 =====
91 10
92 1 1 1 1 3 3 2 2 2 2
93
94 1 3
95 2 5
96 3 5
97
98 =====
99 9
100 9 8 7 6 5 4 3 2 1
101
102 1 8
103 2 6
104 3 4
105 4 2
106 5 0
107 6 2
108 7 4
109 8 6
110 9 8
111 */
112

```