

Our Solution(s)Run Code

Solution 1Solution 2

```
1 // Copyright © 2020 AlgoExpert, LLC. All rights reserved.
2
3 using namespace std;
4
5 bool breaksDirection(int direction, int previousInt, int currentInt) {
6     int difference = currentInt - previousInt;
7     if (direction > 0)
8         return difference < 0;
9     return difference > 0;
10 }
11
12 // O(n) time | O(1) space - where n is the length of the array
13 bool isMonotonic(vector<int> array) {
14     if (array.size() <= 2)
15         return true;
16
17     int direction = array[1] - array[0];
18     for (int i = 2; i < array.size(); i++) {
19         if (direction == 0) {
20             direction = array[i] - array[i - 1];
21             continue;
22         }
23
24         if (breaksDirection(direction, array[i - 1], array[i]))
25             return false;
26     }
27
28     return true;
29 }
30
```

Your SolutionsRun Code

Solution 1Solution 2Solution 3

```
1 using namespace std;
2
3 bool isMonotonic(vector<int> array) {
4     // Write your code here.
5     return false;
6 }
7
```

```

1  #!/usr/bin/env python
2  """Test Case #1, Case 1
3  """
4  def test_case_1():
5      """Test Case #1, Case 1
6      """
7      # Test Case #1, Case 1
8      # Test Case #1, Case 1
9      # Test Case #1, Case 1
10     # Test Case #1, Case 1
11     # Test Case #1, Case 1
12     # Test Case #1, Case 1
13     # Test Case #1, Case 1
14     # Test Case #1, Case 1
15     # Test Case #1, Case 1
16     # Test Case #1, Case 1
17     # Test Case #1, Case 1
18     # Test Case #1, Case 1
19     # Test Case #1, Case 1
20     # Test Case #1, Case 1
21     # Test Case #1, Case 1
22     # Test Case #1, Case 1
23     # Test Case #1, Case 1
24     # Test Case #1, Case 1
25     # Test Case #1, Case 1
26     # Test Case #1, Case 1
27     # Test Case #1, Case 1
28     # Test Case #1, Case 1
29     # Test Case #1, Case 1
30     # Test Case #1, Case 1
31     # Test Case #1, Case 1
32     # Test Case #1, Case 1
33     # Test Case #1, Case 1
34     # Test Case #1, Case 1
35     # Test Case #1, Case 1
36     # Test Case #1, Case 1
37     # Test Case #1, Case 1
38     # Test Case #1, Case 1
39     # Test Case #1, Case 1
40     # Test Case #1, Case 1
41     # Test Case #1, Case 1
42     # Test Case #1, Case 1
43     # Test Case #1, Case 1
44     # Test Case #1, Case 1
45     # Test Case #1, Case 1
46     # Test Case #1, Case 1
47     # Test Case #1, Case 1
48     # Test Case #1, Case 1
49     # Test Case #1, Case 1
50     # Test Case #1, Case 1
51     # Test Case #1, Case 1
52     # Test Case #1, Case 1
53     # Test Case #1, Case 1
54     # Test Case #1, Case 1
55     # Test Case #1, Case 1
56     # Test Case #1, Case 1
57     # Test Case #1, Case 1
58     # Test Case #1, Case 1
59     # Test Case #1, Case 1
60     # Test Case #1, Case 1
61     # Test Case #1, Case 1
62     # Test Case #1, Case 1
63     # Test Case #1, Case 1
64     # Test Case #1, Case 1
65     # Test Case #1, Case 1
66     # Test Case #1, Case 1
67     # Test Case #1, Case 1
68     # Test Case #1, Case 1
69     # Test Case #1, Case 1
70     # Test Case #1, Case 1
71     # Test Case #1, Case 1
72     # Test Case #1, Case 1
73     # Test Case #1, Case 1
74     # Test Case #1, Case 1
75     # Test Case #1, Case 1
76     # Test Case #1, Case 1
77     # Test Case #1, Case 1
78     # Test Case #1, Case 1
79     # Test Case #1, Case 1
80     # Test Case #1, Case 1
81     # Test Case #1, Case 1
82     # Test Case #1, Case 1
83     # Test Case #1, Case 1
84     # Test Case #1, Case 1
85     # Test Case #1, Case 1
86     # Test Case #1, Case 1
87     # Test Case #1, Case 1
88     # Test Case #1, Case 1
89     # Test Case #1, Case 1
90     # Test Case #1, Case 1
91     # Test Case #1, Case 1
92     # Test Case #1, Case 1
93     # Test Case #1, Case 1
94     # Test Case #1, Case 1
95     # Test Case #1, Case 1
96     # Test Case #1, Case 1
97     # Test Case #1, Case 1
98     # Test Case #1, Case 1
99     # Test Case #1, Case 1
100    # Test Case #1, Case 1

```

Run or submit code when you're ready.