**Check Number sequence**

**Send Feedback**

You are given S a sequence of n integers i.e. S = s1, s2, ..., sn. Compute if it is possible to split S into two parts : s1, s2, ..., si and si+1, si+2, ….., sn (0 <= i <= n) in such a way that the first part is strictly decreasing while the second is strictly increasing one.

**Note : We say that x is strictly larger than y when x > y.**

**So, a strictly increasing sequence can be 1 4 8. However, 1 4 4 is NOT a strictly increasing sequence.**

That is, in the sequence if numbers are decreasing, they can start increase at one point. And once number starts increasing, they cannot decrease at any point further.

**Sequence made up of only increasing numbers or only decreasing numbers is a valid sequence. So in both the cases, print true.**

**You just need to print true/false. No need to split the sequence.**

**Input format :**

Line 1 : Integer 'n'

Line 2 and Onwards : 'n' integers on 'n' lines(single integer on each line)

**Output Format :**

"true" or "false" (without quotes)

**Constraints :**

0 <= n <= 10^7

**Sample Input 1 :**

5

9

8

4

5

6

**Sample Output 1 :**

true

**Sample Input 2 :**

3

1

2

3

**Sample Output 2 :**

true

**Sample Input 3 :**

3

8

7

7

**Sample Output 3 :**

false

**Explanation for Sample Format 3 :**

8 7 7 is not strictly decreasing, so output is false.

**Sample Input 4 :**

6

8

7

6

5

8

2

**Sample Output 4 :**

false

**Explanation for Sample Input 4 :**

The series is :

8 7 6 5 8 2

It is strictly decreasing first (8 7 6 5). Then it's strictly increasing (5 8). But then it starts strictly decreasing again (8 2). Therefore, the output for this test case is 'false'