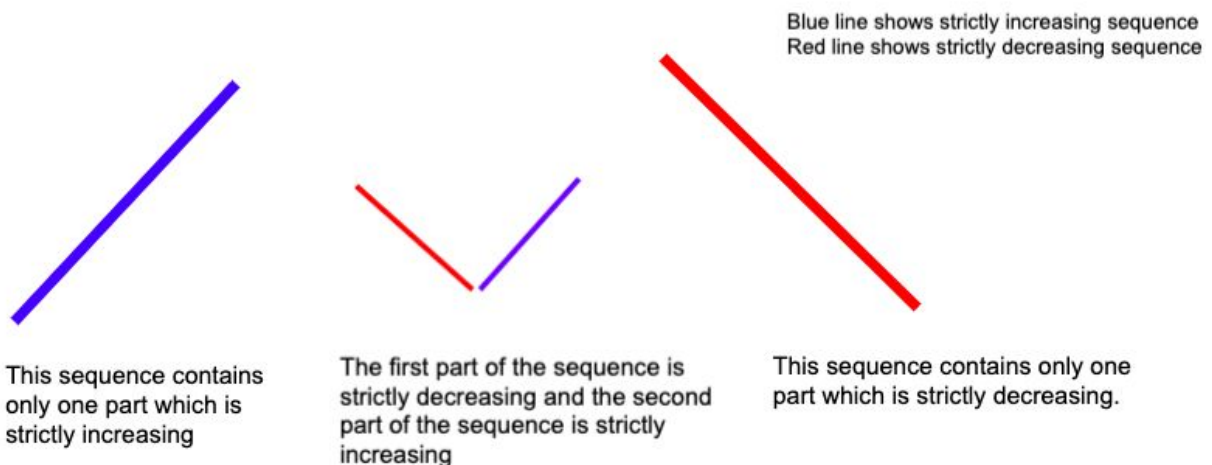# Check Number sequence

---

Problem Description: You are given a sequence of numbers and you have to check whether it is possible to split the sequence into 2 parts such that 1st part is strictly decreasing while the other part is strictly increasing.

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


How to approach?

In this problem, we have these valid sequences:

Blue line shows strictly increasing sequence
Red line shows strictly decreasing sequence

This sequence contains only one part which is strictly increasing

The first part of the sequence is strictly decreasing and the second part of the sequence is strictly increasing

This sequence contains only one part which is strictly decreasing.

So, we need to traverse the whole array once, so for this we need to maintain a count variable and run a loop until this count becomes equal to the number n. In each iteration of this loop, take the current number as input and check whether it is equal to the previous number. If this is true, then we should directly print false, as the sequence is neither strictly increasing or decreasing in that case.

Now, if the current number is less than the previous one, then we will check whether the previous sequence was decreasing or not with the help of a flag *isDec*. If this flag is true that means the previous sequence was decreasing, then we will move to the next number, but if this flag is false that means we found a decrease after a previously increasing sequence, so we should directly print false.

isDec=true

P.

C

Valid Sequence

isDec=false

P.

C

Invalid Sequence
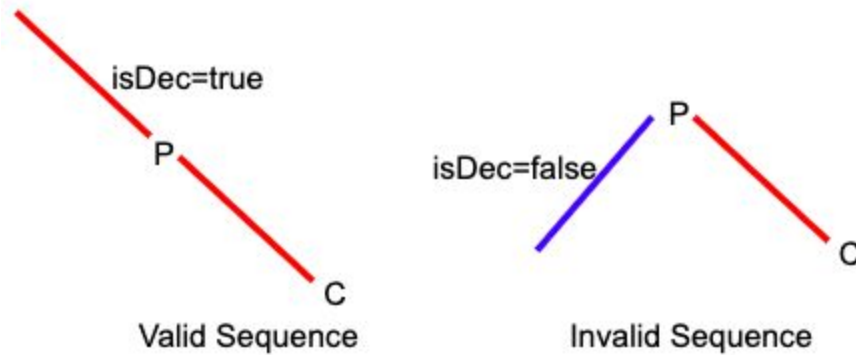
But if the current number is greater than the previous number we will say that if the previous sequence was decreasing i.e. if *isDec* is true, now we have found an increase in the sequence so we will make *isDec* as false. And, if the previous sequence was increasing we will do nothing and continue.

C

isDec=true

isDec=false

P

Valid Sequence

isDec=false

C

isDec=false

P

Valid Sequence
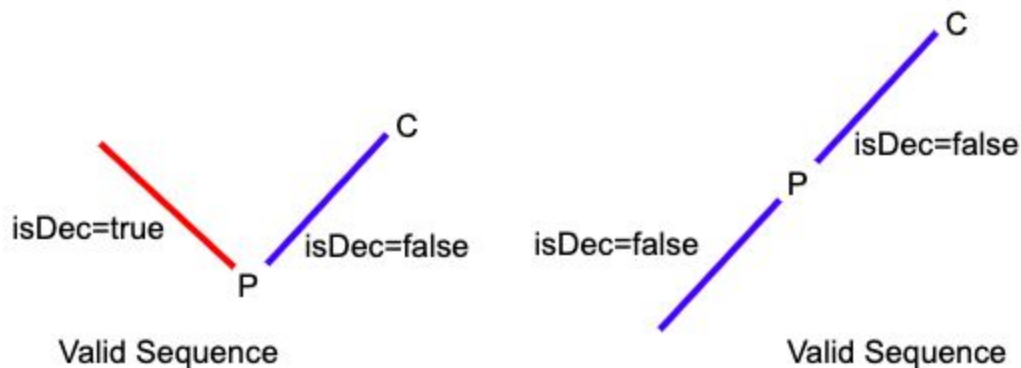
Step by step implementation:
1. Take the number of integers N as input from the user, and then take the 1st number as input from the user, in variable previous.
2. Initialize the number count by 2, and an isDec by true.
3. Run a loop until count becomes equal to N, increment it in each iteration.
4. In each iteration, take the current number as input from the user. Check whether the current number is equal to the previous number, if yes then we print false directly.
5. If the current number is less than the previous number, then we will check if isDec is false, then we will directly print false. If isDec is true, then we will continue.
6. But if the current number is greater than the previous number, then we will check if isDec is true. If isDec is true then make it as false, as we have found an increase in the sequence here. If isDec is false, then we will continue.

7. Make previous number=current number.

Pseudo Code for this problem:
Input=N, prev
Count=2, isDec=true
while count is less than or equal to N:
    Input=curr
    If(curr=prev):
            print(false)
           Return
    If(curr<prev):
            If isDec=false:
                   print(false)
                  Return
      else:
           If isDec = true:
                isDec = false

    prev = current
    Increment count by 1.
print (true)

❏ Let us dry run the code:
   5
   9 8 4 5 6
        ● N=5, prev=9
           count=2, isDec=true

        ● curr=8
           curr<prev and isDec is true:
           prev=curr=8, count=3

        ● curr=4
           curr<prev and isDec is true:
           prev=curr=4, count=4

- curr=5
  curr>prev and isDec is true:
  So isDec=false
  prev=curr=5, count=5

- curr=6
  curr>prev and isDec is false.
  prev=curr, count becomes greater than 5, i.e. equal to 6
  So, print true.

- So, final output:
  true