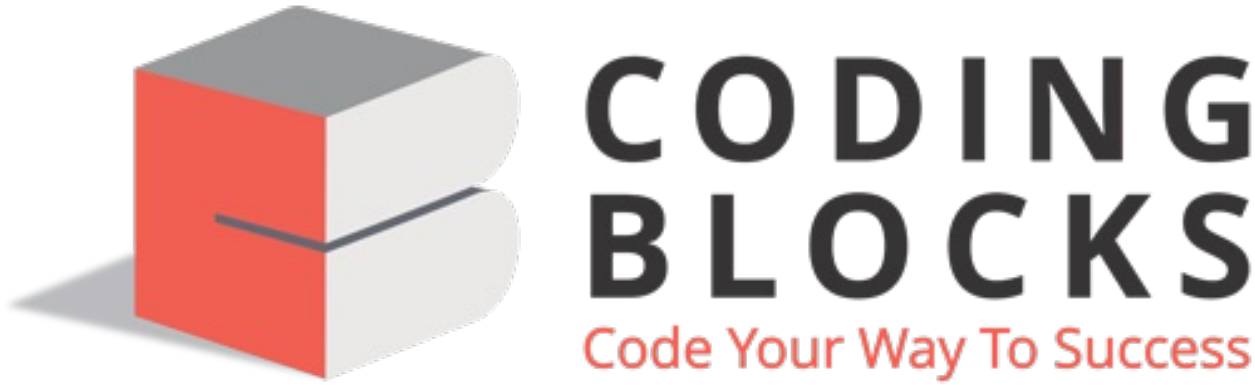

##Increasing Decreasing Sequence - Hint
We have to check whether the given sequence is in such an order that it first decreases (strictly) and then increases (strictly).

We assume our sequence to be in a decreasing order in the beginning. We maintain a flag variable to mark it.

If at any point, we encounter such a point where our sequence starts increasing, we update our flag variable and then from thereon check whether the sequence is strictly increasing.

If however, the sequence again starts decreasing in this part, then the sequence does not satisfy the given condition and we should print 'false' here.

Else if we are able to completely traverse the sequence without violating the given condition, we print true.

Since all comparisons are made between two consecutive elements only at any point of time, we must only store them and discard the rest thus saving space.

####Pseudocode

```
function increasingDecreasingSequence(int n):
    bool isDecreasing = true  //Assume sequence to be decreasing initially
    int prev = inputInteger() //Variable to store prev element
    for i in [1, n-1]:
        int curr = inputInteger()
        if(curr == prev) {
            // Since all sequences must be either strictly increasing
            // or decreasing, two consecutive elements cannot
            // be equal
            return false
        }
        else if(curr > prev) {
            // Sequence started increasing
            isDecreasing = false
        }
        else if(!isDecreasing and curr < prev) {
            // If sequence is increasing and then starts decreasing again,
            // condition is violated
            return false
        }
    prev = curr
}

//If no violations, return true at end
return true



}//End of function
```