Question - 1

Ques

In a warehouse, a manager would like to maintain a list of unique *items* in the stock, according to the following rules:
--> Items should be sorted in non-decreasing order of their frequency in the stock.
--> In case, there are more than one items having same frequency, then they must be sorted in non-decreasing order according to the item-values.

Input Format:
The first line contains an integer *n*, the total number of *items* present in the stock.
The next *n* lines each contain an integer, item-value.
Output Format:
List of uniquely sorted items [One in each new line]
Constraints

- 1 ≤ n ≤ 1000
- 1 ≤ item-value ≤ 1000

Sample Test Case 1:
Input:
7 <-- n=7
4 <-- array of item-values = [4, 1, 5, 1, 4, 2, 4]
1
5
1
4
2
4
Output:
2
5
1
4
Explanation:

- There are 2 values that occur **once**: [2, 5].
- There is 1 value that occur **twice**: [1].
- There is 1 value that occurs **thrice**: [4].
- The array of items sorted by frequency and then by item-value in ascending order is *[2, 5, 1, 4].*

Sample Test Case 2:
Input:
6
4
5
6
5
4
3
Output:

3
6
4
5

Explanation:

n = 6
items = [4, 5, 6, 5, 4, 3]

- There are 2 values that occur **once**: [3, 6].
- There are 2 values that occur **twice**: [4, 5].
- The array of items sorted by frequency and then by item-value in ascending order is *[3, 6, 4, 5].*

Question - 2

Ques

You are given an integer array of **N** elements, whose elements are sorted in ascending order. But, there may exist two indices **i, j** (both inclusive) in this array, such that the sequence of elements in between these indices are inverted [reverse sorted]. If we reverse the elements in between the indices i and j (both inclusive), then the array becomes completely sorted.
You need to identify the starting-index *i.e.* **i**, and ending-index *i.e.* **j** in the given array.

****NOTE:***
If no such inversion exists, then starting-index and ending-index, both must display -1 as their respective outputs.

Input Format:
First-line has N: Number of Elements of the array
Next, N lines have the sequence of N elements of the array
Output Format:
Starting-Index in first-line
Ending-Index in second-line
Constraints:

1. Array Index Values start from 0
2. 1<= N <= 1000.
3. 0<= Array-elements <= 1000.

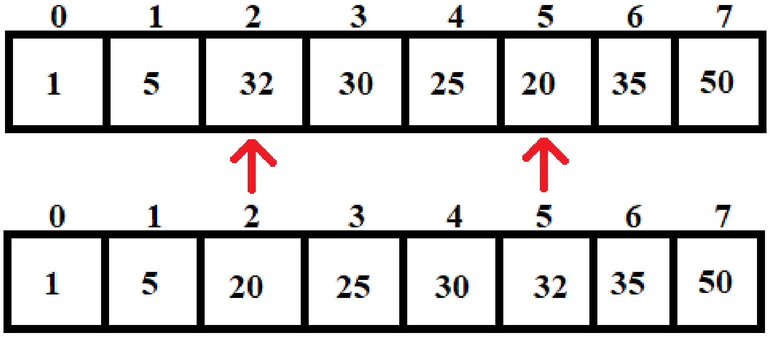
Sample Case 1:
Input:
8   <- N [Number of inputs for the array]
1   <- array values [1, 5, 32, 30, 25, 20, 35, 50]
5
32
30
25
20
35
50

Output:
2 <- Starting-Index
5 <- Ending-Index

Explanation:
If we invert the sequence starting from index 2 and ending at index 5, i.e., from [32,30,25,20] to [20,25,30,32], we will get a completely sorted array [1,5,20,25,30,32,35,50].

/

So, the output is 2 as starting-index in first-line and 5 as ending-index in second-line. Below figure represents the visualization for the same.



Sample Case 2:
Input:
6
11
22
33
44
55
66
Output:
-1
-1
Explanation:
N=6 and array values are [11, 22, 33, 44, 55, 66]
As the sequence is already sorted, no such inversion exists.
So, the output is -1 as the starting-index in first-line and -1 as ending-index in the second line.

/