Practice Arena

Practice problems aimed to improve your coding skills.

- PRACTICE-02_SCAN-PRINT
- PRACTICE-03_TYPES
- LAB-PRAC-02_SCAN-PRINT
- LAB-PRAC-01
- PRACTICE-04_COND
- BONUS-PRAC-02
- LAB-PRAC-03_TYPES
- PRACTICE-05_COND-LOOPS
- LAB-PRAC-04_COND
- LAB-PRAC-05_CONDLOOPS
- PRACTICE-07_LOOPS-ARR
- LAB-PRAC-06_LOOPS
- LAB-PRAC-07_LOOPS-ARR
- LABEXAM-PRAC-01_MIDSEM
- PRACTICE-09_PTR-MAT
- LAB-PRAC-08_ARR-STR
- PRACTICE-10_MAT-FUN
- LAB-PRAC-09_PTR-MAT
- LAB-PRAC-10_MAT-FUN
- PRACTICE-11_FUN-PTR
- LAB-PRAC-11_FUN-PTR
- LAB-PRAC-12_FUN-STRUC
- LABEXAM-PRAC-02_ENDSEM
- LAB-PRAC-13_STRUC-NUM
- LAB-PRAC-14_SORT-MISC
  - Predecessor and Successor
  - Insertion Sort
  - Link a List
  - The United Sums of Arrays
  - Bubble Sort
  - Pretty Queues Revisited
  - Just About Sorted
  - Brick Sort
  - All My Descendants
  - Mr C likes a Majority
  - Cocktail Sort
  - All My Descendants - Part II

Just About Sorted

LAB-PRAC-14_SORT-MISC

Just About Sorted [20 marks]

Problem Statement
Mr C took a lot of pain to sort an array in non-decreasing order. However, one of the mischievous clones came and messed up the array by rotating the array right by a few elements. For example, if the nice original sorted array was
1 2 3 4 5
then rotating it right by one location would give the array
5 1 2 3 4
Rotating the original sorted array two locations to the right would give the array
4 5 1 2 3
Rotating the original sorted array four locations to the right would give the array
2 3 4 5 1
We will not consider rotations by five locations since that would give back the original array. Thus, if the array has n elements, then the clone could have rotated it right by 0 locations (i.e. no change), 1 location, 2 locations, ... n-1 locations.

The first line of the input will give you n, the number of elements in the array. In the next line we will give you n integers, separated by a space. These will constitute the array which was sorted and then rotated to the right by a certain number of locations. In the third line of the input we will give you a query integer q.

In the first line of the output, print by how many locations to the right did the clone rotate the original sorted array. Remember that if there are n elements in the array, then this number can be between 0 and n-1 (both included). In the second line of the output print the smallest index at which the query element occurs, if it does occur in the given (possibly rotated) array, else print -1 to indicate that the query element is not present in the given array at all.

You would see that both the above problems are very easy to solve using O(n) time. The challenge is to solve these using O(log n) time using techniques we learnt in binary search (i.e. divide and conquer).

Caution

1. You have to print the location of the query element in the array that is given to you, i.e. the array that was possibly rotated after being sorted. Do not give the location of the query element in the original sorted array.
2. If the query element appears twice in the given (possibly rotated) array, print the smallest index at which it occurs.
3. The array elements may be positive, negative or zero. The query element may also be positive, negative or zero.
4. The array may contain repeated elements. However, we promise you that there will be at least two distinct elements in the array i.e. the original sorted array will never look like 6 6 6 6 6 6 6 since rotations do not make sense in such an array. However, the original sorted array may look like 6 6 6 6 6 7.
5. Be careful about extra/missing lines and extra/missing spaces in your output.

EXAMPLE:
INPUT
5
3 4 5 1 2
5

OUTPUT:
3
2

Explanation: The original sorted array was 1 2 3 4 5 and it was rotated 3 locations to the right to get 3 4 5 1 2. The element 5 is present in the given array but at index 2.

Grading Scheme:
Total marks: **[20 Points]**

There will be partial grading in this question. There are two lines in your output. Printing each line correctly, in the correct order, carries 50% weightage. Each visible test case is worth 2 points and each hidden test case is worth 4 points. There are 2 visible and 4 hidden test cases.

Please remember, however, that when you press Submit/Evaluate, you will get a green bar only if all parts of your answer are correct. Thus, if your answer is only partly correct, Prutor will say that you have not passed that test case completely, but when we do autograding afterwards, you will get partial marks.

Start Solving! (/editor/practice/6291)