ESO207
Data Structures and Algorithms
Indian Institute of Technology, Kanpur

Date of Submission: Aug 25, 2019

Programming Assignment

1

Instructions.

- Solution of Programming Assignments(PA) are accepted through a contest created on Hacker Rank. Individual submissions via emails will not be graded.
- HackerRank will accept solutions only in following programming languages - C, C++, Java.
- Grading of PAs is entirely automatic. Submissions made by UserIDs shared with us on Moodle through Google Forms will only be graded.
- Deadline for not receiving any late submission penalties is *25th Aug*. You may choose to delay your submission by **24 hrs.** with a **penalty of 10%** and by **48 hrs.** with a **penalty of 25%**.
- Use of algorithm header file like $< algorithm.h >$ and primitive functions for operations like sorting, searching and implementing data structures constitutes plagiarism. This course is on Data Structure and Algorithm; hence, we expect students to write their code rather than using predefined function.
- Please keep in mind the *Programming Anti-Plagiarism Policy* mentioned on Moodle while making submissions.

Problem 1: Element Search - Rotated Sorted Array

A right circular rotation of an array $A[0 \cdots n-1]$ is shifting of i^{th} element to $(i+1)\ mod\ (n-1)$ position.

A rotated sorted array is a sorted array with distinct elements which is right circular shifted multiple times. For instance array $\{3,4,5,0,1,2\}$ is right circular shift of $\{0,1,2,3,4,5\}$ by 3 times.

Given a rotated sorted array and a target element. Write a program to return the index of the target element in the given array. Return -1 if the target element is not found to be in the array.

Input Format

First line of each input is a positive integer t - number of test cases. Each input consists of 3 lines.

- n - the size of the array.
- n space seperated integers of input array.
- k - target element

Output Format

Output the index of k in input array. Return -1 if the element is not found in the array.

Constraints

- $1 \leq n \leq 10^6$
- $1 \leq a_i \leq 10^{15}$

Example

Input	Output
2 7 4 5 6 7 0 1 2 6 7 8 9 10 0 2 5 6 0	2 3

Explanation

Indexing of elements starts with 0. Hence, the index of target element 6 is 2 and that of 0 is 3.

Problem 2: Counting Inversion

Given an array $\{a_1, a_2, \ldots, a_n\}$ the number of inversion in the array is the number of pairs (a_i, a_j) such that $a_i > a_j$ and $i < j$. For example, number of inversions in array $\{1, 3, 4, 2\}$ is 2 - $(3, 2)$, $(4, 2)$.

Write a program which takes an array as input and outputs number of inversions in that array.

Input Format

First line of each input is a positive integer t - number of test cases. Each test case contains two lines -

- n - size of input array.
- n space separated integers of input array.

Output Format

For each test case output the number of inversions.

Constraints

- $1 \leq t \leq 5$
- $1 \leq n \leq 10000$
- $1 \leq a_i \leq 10^{15}$

Example

Input	Output
3 5 1 1 1 1 1 5 1 2 3 4 5 5 5 4 3 2 1	0 0 10

Explanation

In input 3 on first line signifies the number of following arrays on which we want to count inversion. The first array is all elements same and the second is sorted, therefore no inversions in both. In the last array, every unordered pair a inversion hence 10.