1000. Binary Searchable

Total: 99 Accepted: 57

Time Limit: 1sec Memory Limit:32MB

Description

Binary search is a search algorithm usually used on a sorted sequence to quickly find an element with a given value. The algorithm works by repeatedly comparing the sought value with some element from the sequence (the *pivot*), then discarding as much of the sequence as possible based on the comparison and the assumption that the sequence is sorted. Note that in the standard algorithm, the pivot element is as close as possible to the middle of the sequence, but we'll remove that restriction.

The pseudocode for performing binary search on a sequence S, looking for value X is as follows:

```
  1 binary_search(S, X)
2    while S is not empty
3       choose an element from S as the pivot
4       if pivot = X, return true (X was found)
5       else if pivot < X, remove pivot and all elements before it from S
6       else remove pivot and all elements after it from S
7    end while
8    return false (X was not found)
```

In this problem we will evaluate how binary search performs on data that isn't necessarily sorted. An element of S is said to be *binary searchable* if, regardless of how the pivot is chosen on line 3 whenever the line is executed, the above algorithm returns true.

Given a **sequence** containing distinct integers, output the number of elements from the sequence that are binary searchable.

Input

Input may contain many cases.

The first line of each case is the length(between 0 and 100, inclusive) of the sequence. The second line contains the elements in the sequence, seperated by one space.

Output

For each case, output the number of elements from the sequence that are binary searchable, in one line.

Sample Input

Copy

```
3
1 3 2
7
3 2 1 10 23 22 21
6
1 5 7 11 12 18
6
5 4 3 2 1 0
```

Sample Output

Copy

```
1
1
```

```
6
0
```

Submit