

# Problem I: A Smash Hit

Filename: `smashing`

Time limit: 1 `second`

Your backyard movie studio has been hard at work on its latest monster movie. In the spirit of many classic monster movies, you want to have a scene where the monster (played by one of your friends in costume) rampages through town (a collection of tiny paper-mache houses, in a straight line) destroying buildings and generally causing chaos. Since movie watchers only notice the big details, and you want to avoid overwhelming your audience (and save some of the houses for later movies), you've instructed your friend to only knock over the tallest building. Unfortunately, the monster costume you created has quite poor visibility, so your friend will knock over any building if it's the tallest that your friend has touched so far on the rampage. To help decide where to start the rampage, you need to write a program that determines how many buildings will be destroyed for a given starting location and direction. Your friend will always start at one of the buildings in town, and will always move in the same direction as the rampage was started in.

## Problem

For each query, find the number of buildings your friend will knock over if the rampage moves left, followed by the number of buildings your friend will knock over if the rampage moves right. Your friend can only knock over buildings directly in the path of the rampage.

## Input

Input will begin with a single positive integer, **c**, indicating that **c** test cases follow. The input for each case will begin with a single integer, **b**, representing the number of buildings. The next line will contain **b** integers representing the height of each building. The heights may be referred to as  $h_i$  ( $1 \leq i \leq b$ ). The following line will contain a single integer, **q**, representing the number of queries. The next **q** lines will contain a single integer **s**, representing the index of the house where the rampage will start.

## Output

For each query, output one line containing the number of buildings smashed if your friend moves to the left, followed by a single space, followed by the number of buildings smashed if your friend rumpages to the right.

## Input Bounds and Corresponding Credit

40 Points	60 Points
<ul style="list-style-type: none"><li>• <math>1 \leq c \leq 20</math></li><li>• <math>1 \leq b \leq 1,000</math></li><li>• <math>1 \leq h_i \leq 2,000,000,000</math></li><li>• <math>1 \leq q \leq 1,000</math></li><li>• <math>1 \leq s \leq b</math></li></ul>	<ul style="list-style-type: none"><li>• <math>1 \leq c \leq 35</math></li><li>• <math>1 \leq b \leq 10^6</math></li><li>• <math>1 \leq h_i \leq 2,000,000,000</math></li><li>• <math>1 \leq q \leq 10,000</math></li><li>• <math>1 \leq s \leq b</math></li></ul>

## Samples

Input	Output
1	2 3
5	1 2
4 2 1 3 7	
2	
2 1	

**Sample Explanation:** In the query, we start at building 2, which has height 2. If we go to the left, we first knock over this building, followed by knocking over the building with height 4, since  $4 > 2$ . If we go to the right, we first knock over the building of height 2, skip the building with height 1 (since 1 isn't greater than 2), then knock over the building with height 3, since  $3 > 2$ , and finally knock over the building with height 7, since  $7 > 3$ .

In the second query, moving left, we only knock over the leftmost building with height 4. When we move right, we knock over the building with height 4 first, then skip the next three buildings which are all of less height than 4, and end by knocking over the building with height 7.