# 1166 – Old Sorting

Given an array containing a permutation of **1** to **n**, you have to find the minimum number of swaps to sort the array in ascending order. A swap means, you can exchange any two elements of the array.

For example, let **n = 4**, and the array be **4 2 3 1**, then you can sort it in ascending order in just **1** swaps (by swapping **4** and **1**).

## Input

Input starts with an integer **T (≤ 100)**, denoting the number of test cases.

Each case contains two lines, the first line contains an integer **n (1 ≤ n ≤ 100)**. The next line contains **n** integers separated by spaces. You may assume that the array will always contain a permutation of **1** to **n**.

## Output

For each case, print the case number and the minimum number of swaps required to sort the array in ascending order.

| Sample Input | Output for Sample Input |
|---|---|
| 3<br>4<br>4 2 3 1<br>4<br>4 3 2 1<br>4<br>1 2 3 4 | Case 1: 1<br>Case 2: 2<br>Case 3: 0 |