331 Mapping the Swaps

Sorting an array can be done by swapping certain pairs of adjacent entries in the array. This is the fundamental technique used in the well-known bubble sort. If we list the identities of the pairs to be swapped, in the sequence they are to be swapped, we obtain what might be called a swap map. For example, suppose we wish to sort the array A whose elements are 3, 2, and 1 in that order. If the subscripts for this array are 1, 2, and 3, sorting the array can be accomplished by swapping A2 and A3, then swapping A1 and A2, and finally swapping A2 and A3. If a pair is identified in a swap map by indicating the subscript of the first element of the pair to be swapped, then this sorting process would be characterized with the swap map 2 1 2.

It is instructive to note that there may be many ways in which swapping of adjacent array entries can be used to sort an array. The previous array, containing 3 2 1, could also be sorted by swapping A1 and A2, then swapping A2 and A3, and finally swapping A1 and A2 again. The swap map that describes this sorting sequence is 1 2 1.

For a given array, how many different swap maps exist? A little thought will show that there are an infinite number of swap maps, since sequential swapping of an arbitrary pair of elements will not change the order of the elements. Thus the swap map 1 1 1 2 1 will also leave our array elements in ascending order. But how many swap maps of minimum size will place a given array in order? That is the question you are to answer in this problem.

Input

The input data will contain an arbitrary number of test cases, followed by a single '0'. Each test case will have a integer n that gives the size of an array, and will be followed by the n integer values in the array.

Output

For each test case, print a message similar to those shown in the sample output below. In no test case will n be larger than 5.

Sample Input

```
2 9 7
2 12 50
3 3 2 1
3 9 1 5
0
```

Sample Output

```
There are 1 swap maps for input data set 1.
There are 0 swap maps for input data set 2.
There are 2 swap maps for input data set 3.
There are 1 swap maps for input data set 4.
```