Gnome Sort

Filename: `gnome`

The Dutch garden gnome (*tuinkabouter*) is the bane of any gardener's existence. These mischievous little creatures delight in wrecking artistic arrangements of flowerpots by arranging every line of flowerpots in non-decreasing order of size. Recently, researchers have succeeded in identifying the sorting algorithm used by the gnomes. For their next experiment, they intend to catch a gnome in the act of sorting, and so they have given you the job of writing a simulator that performs the gnome sort[1].

In order to sort a line of flowerpots, a gnome starts by standing in front of the leftmost pot in the line, and performs a series of steps. The rules are as follows:

- If he is at the leftmost pot, he takes a step to the right.
- If the pot in front of him is not smaller than the pot to his left, he takes a step to the right.
- If the pot in front of him is smaller than the pot to his left, he swaps the two and takes a step to the left.
- If there is no pot in front of him, he is done, and yells "Sorted!"

The Problem:

Given an arrangement of flowerpots, simulate the steps used by the gnomes in the Gnome Sort.

The Input:

The input contains descriptions of gardens, which each contain flowerpots to sort. The first line of each garden description contains a single positive integer, n ($n \leq 500$), the number of flowerpots to be sorted. On the following line, there are n integers (not necessarily distinct) indicating the sizes of the flowerpots from left to right. End of input will be indicated by a garden with 0 flowerpots. This case should not be processed.

The Output:

At the beginning of the output for each garden, you must print "`Garden c:`" on a line by itself where c is the garden number (starting from 1). On the following lines, you must print the sequence of swaps performed by the gnome, in the form: "`The gnome swaps the pots at positions a and b.`" where a and b are the 1-based positions of the pots the gnome is swapping. Ensure that a is always the smaller of the two positions. Each swap should be printed on a line by itself. The output for each garden should be ended by the word "`Sorted!`" on a line by itself. Leave a blank line after the output for each garden.

[1] Interesting tidbit: There is, in fact, a real sorting algorithm called gnome sort, and it works in precisely the way this problem describes. Well, except for the gnomes.

Sample Input:

```
5
5 4 3 2 1
3
1 2 3
6
1 7 3 2 9 8
0
```

Sample Output:

```
Garden 1:
The gnome swaps the pots at positions 1 and 2.
The gnome swaps the pots at positions 2 and 3.
The gnome swaps the pots at positions 1 and 2.
The gnome swaps the pots at positions 3 and 4.
The gnome swaps the pots at positions 2 and 3.
The gnome swaps the pots at positions 1 and 2.
The gnome swaps the pots at positions 4 and 5.
The gnome swaps the pots at positions 3 and 4.
The gnome swaps the pots at positions 2 and 3.
The gnome swaps the pots at positions 1 and 2.
Sorted!

Garden 2:
Sorted!

Garden 3:
The gnome swaps the pots at positions 2 and 3.
The gnome swaps the pots at positions 3 and 4.
The gnome swaps the pots at positions 2 and 3.
The gnome swaps the pots at positions 5 and 6.
Sorted!
```