


[Contests](#)
[Virtual
Contests](#)
[Problems](#)
[Submit](#)
[Runs Status](#)
[Rank List](#)
[Forum](#)

1720. Brainman

Time Limit: 1.0 Seconds Memory Limit: 65536K

Total Runs: 6243 Accepted Runs: 2071

Raymond Babbitt drives his brother Charlie mad. Recently Raymond counted 246 toothpicks spilled all over the floor in an instant just by glancing at them. And he can even count Poker cards. Charlie would love to be able to do cool things like that, too. He wants to beat his brother in a similar task.

Here's what Charlie thinks of. Imagine you get a sequence of N numbers. The goal is to move the numbers around so that at the end the sequence is ordered. The only operation allowed is to swap two adjacent numbers. Let us try an example:

```
Start with: 2 8 0 3
swap  (2 8) 8 2 0 3
swap  (2 0) 8 0 2 3
swap  (2 3) 8 0 3 2
swap  (8 0) 0 8 3 2
swap  (8 3) 0 3 8 2
swap  (8 2) 0 3 2 8
swap  (3 2) 0 2 3 8
swap  (3 8) 0 2 8 3
swap  (8 3) 0 2 3 8
```

So the sequence (2 8 0 3) can be sorted with nine swaps of adjacent numbers. However, it is even possible to sort it with three such swaps:

```
Start with: 2 8 0 3
swap  (8 0) 2 0 8 3
swap  (2 0) 0 2 8 3
swap  (8 3) 0 2 3 8
```

The question is: What is the minimum number of swaps of adjacent numbers to sort a given sequence? Since Charlie does not have Raymond's mental capabilities, he decides to cheat. Here is where you come into play. He asks you to write a computer program for him that answers the question. Rest assured he will pay a very good prize for it.

Input

The first line contains the number of scenarios.

For every scenario, you are given a line containing first the length N ($1 \leq N \leq 1000$) of the sequence, followed by the N elements of the sequence (each element is an integer in $[-1000000, 1000000]$). All numbers in this line are separated by single blanks.

Output

Start the output for every scenario with a line containing "Scenario #i:", where i is the number of the scenario starting at 1. Then print a single line containing the minimal number of swaps of adjacent numbers that are necessary to sort the given sequence. Terminate the output for the scenario with a blank line.

Sample Input

```
4
4 2 8 0 3
10 0 1 2 3 4 5 6 7 8 9
6 -42 23 6 28 -100 65537
5 0 0 0 0 0
```

Sample Output

Scenario #1:
3

Scenario #2:
0

Scenario #3:
5

Scenario #4:
0

Source: *TUD Programming Contest 2003*

[Submit](#) [List](#) [Runs](#) [Forum](#) [Statistics](#)

[Tianjin University Online Judge](#) v1.3.0

Maintance:Fxz. Developer: [SuperHacker](#), [G.D.Retop](#), [Fxz](#)