UCF Local Contest — September 1, 2012

Palindrome Maker

filename: `palindrome`

A "palindromic integer sequence" is a sequence that is the same when written forwards or backwards, i.e., of the form {a1, a2, a3, …. , a3, a2, a1}. Some examples of palindromic integer sequences are {78, 91, 78}, {100}, {10, 20, 20, 10}, and {5, 5, 5, 5}. But {1, 2, 3, 1} and {10, 20} are not palindromic sequences.

You are given an integer sequence. You want to convert the sequence into a palindromic integer sequence by a series of insertions. You can convert {1, 2, 3, 1} to a palindromic sequence by inserting a 2 at the fourth position which will become {1, 2, 3, 2, 1}, and you can convert {10, 20} to palindromic sequence by inserting 20 at the first position or inserting 10 at the last position.

The Problem:

Given an integer sequence, determine the minimum number of insertions required to convert it into a palindromic sequence. It is guaranteed that the result (number of insertions) will always be less than 100.

The Input:

The first input line contains a positive integer, *n*, indicating the number of integer sequences to check. The sequences are on the following *2n* lines, two lines per sequence. First line contains *s* ($1 \leq s \leq 10{,}000$), the size of the sequence. Second line contains *s* integers (separated by a single space), providing the sequence. The integers in a sequence are between 1 and 50 inclusive.

The Output:

For each sequence, first output "`Sequence #i: `" where *`i`* is the sequence number, starting with 1. Then, output a single integer denoting the minimum number of insertions required to convert the sequence into a palindromic sequence. Leave a blank line after the output for each test case. Follow the format illustrated in Sample Output.

(Sample Input/Output on the next page)

Sample Input:

```
3
4
1 2 3 1
4
1 2 3 4
3
1 2 1
```

Sample Output:

```
Sequence #1: 1

Sequence #2: 3

Sequence #3: 0
```