

# Append Sort

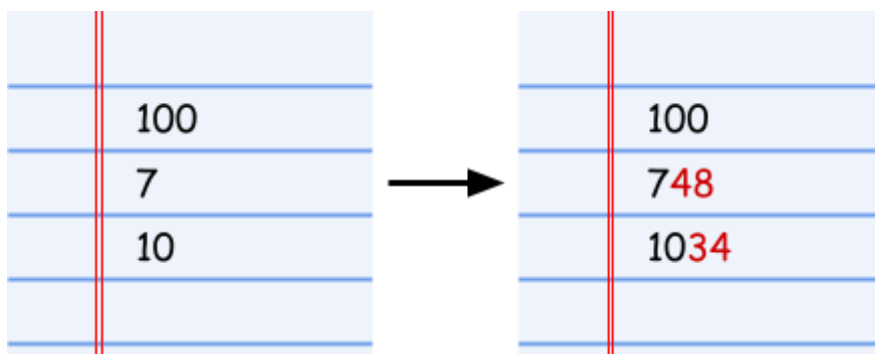
## Problem

We have a list of integers  $X_1, X_2, \dots, X_N$ . We would like them to be in strictly increasing order, but unfortunately, we cannot reorder them. This means that usual sorting algorithms will not work.

Our only option is to change them by appending digits 0 through 9 to their right (in base 10). For example, if one of the integers is 10, you can turn it into 100 or 109 with a single append operation, or into 1034 with two operations (as seen in the image below).

Given the current list, what is the minimum number of single digit append operations that are necessary for the list to be in strictly increasing order?

For example, if the list is 100, 7, 10, we can use 4 total operations to make it into a sorted list, as the following image shows.



## Input

The first line of the input gives the number of test cases,  $T$ .  $T$  test cases follow. Each test case is described in two lines. The first line of a test case contains a single integer  $N$ , the number of integers in the list. The second line contains  $N$  integers  $X_1, X_2, \dots, X_N$ , the members of the list.

## Output

For each test case, output one line containing `Case #x: y`, where  $x$  is the test case number (starting from 1) and  $y$  is the minimum number of single digit append operations needed for the list to be in strictly increasing order.

## Limits

Time limit: 10 seconds.

Memory limit: 1 GB.

$1 \leq T \leq 100$ .

## Test Set 1 (Visible Verdict)

$2 \leq N \leq 3$ .

$1 \leq X_i \leq 100$ , for all  $i$ .

## Test Set 2 (Visible Verdict)

$$2 \leq N \leq 100.$$

$$1 \leq X_i \leq 10^9, \text{ for all } i.$$

### Sample

#### Sample Input

```
4
3
100 7 10
2
10 10
3
4 19 1
3
1 2 3
```

#### Sample Output

```
Case #1: 4
Case #2: 1
Case #3: 2
Case #4: 0
```

In Sample Case #1, the input is the same as in the example given in the problem statement. As the image shows, the list can be turned into a sorted list with 4 operations. Notice that the last two integers need to end up with at least 3 digits (requiring at least 3 append operations in total). If all of the final numbers had exactly three digits, the second would be larger than the third because it starts with a 7 instead of a 1. This means we cannot do it with fewer than 4 operations.

In Sample Case #2, notice that the list needs to be in strictly increasing order, so we have to do at least one operation. In this case, any valid append operation to the second integer works.

In Sample Case #3, we can use two append operations to get the list to 4, 19, 1**93**.

In Sample Case #4, the given list is already in strictly increasing order, so no operations are necessary.