Build-A-Pair

Problem

You want to build a pair of positive integers. To do that, you are given a list of decimal digits to use. You must use every digit in the list exactly once, but you get to choose which ones to use for the first integer and which ones to use for the second integer. You also get to choose the order of the digits within each integer, except you cannot put a zero as the most significant (leftmost) digit in either integer. Note that you cannot choose just a zero for one integer either, because it would not be positive.

For example, you could be given the list $[1, 0, 2, 0, 4, 3]$. Two of the valid pairs you can build are $(200, 143)$ and $(3, 12400)$. The following pairs, on the other hand, are not valid:

- $(0102, 34)$: has a leading zero.
- $(0, 12340)$: has a non-positive integer.
- $(10, 243)$ and $(12300, 47)$: the list of digits in each of these pairs is not exactly equal to the given list of digits.

Given the list of digits to use, what is the minimum absolute difference between the two built integers that can be achieved?

Input

The first line of the input gives the number of test cases, T. T lines follow. Each line describes a test case with a single string of digits D. Each character of D is a digit you must use.

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 possible absolute difference between the two integers built from D according to the rules above.

Limits

Time limit: 5 seconds.
Memory limit: 1 GB.
$1 \le T \le 100$.
Each character of D is a decimal digit.
At least two characters of D are not `0`.

Test Set 1 (Visible Verdict)

$2 \le$ the length of $D \le 8$.

Test Set 2 (Visible Verdict)

$2 \le$ the length of $D \le 36$.

Sample

The optimal pair of integers to build are 31 and 24 for Sample Case #1, 10 and 10 for Sample Case #2, 700 and 80 for Sample Case #3, and 89 and 90 for Sample Case #4.