Every business can make use of a good accountant and, if they're not big on following the law, sometimes a bad one. Bad accountants try to make more money for their employers by fudging numbers without getting caught.

Sometimes a bad accountant wants to make a number larger, and sometimes smaller. It is widely known that tax auditors will fail to notice two digits being swapped in any given number, but any discrepancy more egregious will certainly be caught. It's also painfully obvious when a number has fewer digits than it ought to, so a bad accountant will never swap the first digit of a number with a 0.

Given a number, how small or large can it be made without being found out?

**Input**

Input begins with an integer **T**, the number of numbers that need tweaking. Each of the next **T** lines contains a number **N**.

**Output**

For the *i*th number, print a line containing "Case #*i*: " followed by the smallest and largest numbers that can be made from the original number, using at most a single swap and following the rules above.

**Constraints**

1 ≤ **T** ≤ 100   
0 ≤ **N** ≤ 999999999   
**N** will never begin with a leading 0 unless **N** = 0