**Revolving Digits**

**Time Limit: 2000/1000 MS (Java/Others)    Memory Limit: 65536/32768 K (Java/Others)**

**Problem Description**

One day Silence is interested in revolving the digits of a positive integer. In the revolving operation, he can put several last digits to the front of the integer. Of course, he can put all the digits to the front, so he will get the integer itself. For example, he can change 123 into 312, 231 and 123. Now he wanted to know how many different integers he can get that is less than the original integer, how many different integers he can get that is equal to the original integer and how many different integers he can get that is greater than the original integer. We will ensure that the original integer is positive and it has no leading zeros, but if we get an integer with some leading zeros by revolving the digits, we will regard the new integer as it has no leading zeros. For example, if the original integer is 104, we can get 410, 41 and 104.

**Input**

The first line of the input contains an integer T (1<=T<=50) which means the number of test cases.

For each test cases, there is only one line that is the original integer N. we will ensure that N is an positive integer without leading zeros and N is less than 10^100000.

**Output**

For each test case, please output a line which is "Case X: L E G ", X means the number of the test case. And L means the number of integers is less than N that we can get by revolving digits. E means the number of integers is equal to N. G means the number of integers is greater than N.

**Sample Input**

1

341

**Sample Output**

Case 1: 1 1 1