Given a string of characters, we can permute the individual characters to make new strings. We can then order these strings into alphabetical order.

For example the string ‘abba’ gives rise to the following 6 distinct permutations in alphabetical order.

```
aabb 1
abab 2
abba 3
baab 4
baba 5
bbaa 6
```

Of these 6 permutations, only 2 are palindromes (A string that reads the same when read backwards). These are ‘`abba`’ and ‘`baab`’.

Given a string, you have to find out the nth palindrome in the sorted list of all permutations. For the above case ‘`abba`’ is the 1-st and ‘baab’ is the 2-nd palindrome.

Input

The first line of input gives the number of test cases. Each case contains a string, consisting of lowercase letters only, followed by a space separated positive integer n ($n < 2^{31}$). The length of the string will be at most 30.

Output

For each case, output the case number followed by the nth palindrome, but if the total number of palindromes is less than n output ‘`XXX`’ without the quotes. Follow the sample for exact format.

Sample Input

```
3
abba 1
abba 2
abba 3
```

Sample Output

```
Case 1: abba
Case 2: baab
Case 3: XXX
```