

# Palindrome Maker

One day little girl Lilith was playing with strings. She made a string X which is a palindrome! Suddenly her favorite dog “hehway” came and ate some of the characters from X but not all! Lilith built another palindrome but naughty “hehway” came again and destroyed it! Watching this happening again and again Lilith’s boyfriend Mada thought that he can give her a machine as birthday present that automatically converts any string to palindrome by adding some characters. The machine will be connected with an infinite supply of lowercase Latin characters(‘a’-‘z’).

The machine also has two amazing features :

1. The machine always adds minimum number of characters to the input string in order to convert it to a palindrome.
2. If there are multiple ways then it always produces the lexicographically smallest palindrome.

We all know that machines are not perfect! So, there is no way to guaranty that the machine generated palindrome is the original palindrome that Lilith made in the first place! But, “Mada” thought that it will at least help Lilith to forget the sorrow of losing her favorite palindromes to her favorite dog!

So, when you enter a string X into the machine, the machine’s output looks like N Y, where N is number of new characters that it added to X to make Y. And Y is the lexicographically minimum palindrome produced by adding 0 or more characters with X.

Can you help “Mada” with the algorithm of the machine?

**Input :**

The first line of input determines the number of test cases ( $1 \leq T \leq 2000$ ). Each of the next T lines contains a string S ( $1 \leq |S| \leq 100$ ). S is a string which is left after Lilith’s dog “hehway” ate some of the characters(possibly none!) of the original string. S will be the input of the machine.

**Output :**

Print the machine’s output in each line along with the case number! See Sample test cases for I/O format and explanation for more clarification about the problem.

Sample Input	Sample Output
4	Case 1: 1 aba
ab	Case 2: 0 aba
aba	Case 3: 3 abcdcba
abcd	Case 4: 2 abcba
bca	

**Explanation :**

First input "ab" => We can add a 'b' to its beginning which will make it "bab" we can also add an 'a' to its end which will make it "aba". In either way we are adding one extra character but "aba" is lexicographically smaller than "bab". So our result is "aba". We can also add "ba" to the end of "ab" which will result into "abba" which is also a palindrome, but in that case we would need two characters instead of one. We need to ensure minimal character insertion at first and then try to minimize the result lexicographically.

See this way, (a) + ab + (ba + a) results into "aabbaa" which is lexicographically smaller than "aba" but it needs 4 characters to build that palindrome where "aba" needs only 1. So, "aba" is the desired result!

**In second case the string is already a palindrome!**

**Look at the fourth case. We are allowed to add character at any position of the given string not only just at beginning or end!**