

Your task is, given an integer  $N$ , to make a palindrome (word that reads the same when you reverse it) of length at least  $N$ . Any palindrome will do. Easy, isn't it? That's what you thought before you passed it on to your inexperienced team-mate. When the contest is almost over, you find out that that problem still isn't solved. The problem with the code is that the strings generated are often not palindromic. There's not enough time to start again from scratch or to debug his messy code. Seeing that the situation is desperate, you decide to simply write some additional code that takes the output and adds just enough extra characters to it to make it a palindrome and hope for the best. Your solution should take as its input a string and produce the smallest palindrome that can be formed by adding zero or more characters at its end.

## Input

Input will consist of several lines ending in EOF. Each line will contain a non-empty string made up of upper case and lower case English letters ('A'-'Z' and 'a'-'z'). The length of the string will be less than or equal to 100,000.

## Output

For each line of input, output will consist of exactly one line. It should contain the palindrome formed by adding the fewest number of extra letters to the end of the corresponding input string.

## Sample Input

```
aaaa  
abba  
amanaplanacanal  
xyz
```

## Sample Output

```
aaaa  
abba  
amanaplanacanalpanama  
xyzyx
```