Problem F. SSeeeeiinngg DDoouubbllee

Time limit 1000 ms
Mem limit 262144 kB

A *palindrome* is a string that reads the same backward as forward. For example, the strings `z`, `aaa`, `aba`, and `abccba` are palindromes, but `codeforces` and `ab` are not.

The *double* of a string s is obtained by writing each character twice. For example, the double of `seeing` is `sseeeeiinngg`.

Given a string s, rearrange its double to form a palindrome. Output the rearranged string. It can be proven that such a rearrangement always exists.

Input

The first line of input contains t ($1 \leq t \leq 1000$) — the number of test cases.

The only line of each test case contains a single string s ($1 \leq |s| \leq 100$) consisting only of lowercase English letters.

Note that the sum of $|s|$ over all test cases is not bounded.

Output

For each test case, output a palindromic string of length $2 \cdot |s|$ that is a rearrangement of the double of s.

Examples

Input	Output
4 a sururu errorgorn anutforajaroftuna	aa suurruurruus rgnororerrerorongr aannuuttffoorraajjaarrooffttuunnaa

Note

In the first test case, the double of `a` is `aa`, which is already a palindrome.

In the second test case, the double of `sururu` is `ssuurruurruu`. If we move the first `s` to the end, we get `suurruurruus`, which is a palindrome.

In the third test case, the double of `errorgorn` is `eerrrroorrggoorrnn`. We can rearrange the characters to form `rgnororerrerorongr`, which is a palindrome.