

Binary Trees

Source: <http://acm.tju.edu.cn/toj/showp2801.html>

Maybe you have taken the course called Data Structure before. You should be familiar with the fact that no parenthesis is needed in the suffix expression. But have you ever wondered why?

We know that an expression can be uniquely represented by a binary tree. The suffix expression is the post-order traversal of this tree. Given the post-order traversal of a binary tree, each of whose non-leaf nodes has exactly two children, we can reconstruct the original binary tree if we are told which nodes in the post-order traversal are non-leaves. This means that no parenthesis is needed in the suffix expression.

Here is your task. Given a string S which represents the post-order traversal of a binary tree T in which all of the non-leaf nodes have exactly two children, and given which nodes in S are non-leaves, you must reconstruct the binary tree and output the in-order traversal of T .

Each node of T has a label, which is a letter('a'-'z', 'A'-'Z'). A lowercase letter('a'-'z') means the corresponding node is a leaf, an uppercase letter('A'-'Z') means the corresponding node is a non-leaf.

Input

The input contains an integer on the first line, which indicates the number of test cases. Each test case contains one string S on a line, the post-order traversal of a binary tree.

Output

For each test case, output on a line one string which is the in-order traversal of the corresponding binary tree. There can be no white spaces in your output.

Notes

- The post-order traversal of a binary tree is to visit the left subtree of the root first, then the right subtree and finally the root.
- The in-order traversal of a binary tree is to visit the left subtree of the root first, then the root, and finally the right subtree.

Constraints

- S contains letters('a'-'z', 'A'-'Z') only and don't contain any white spaces.
- The length of S is between 3 and 999, inclusive.
- The length of S is an odd number.
- S will be the post-order traversal of a binary tree.

Sample Input

2
bcA
efBghCA

Sample Output

bAc
eBFgCh