

Reverse words in a given string

Given a String of length S, reverse the whole string without reversing the individual words in it. Words are separated by dots.

Input:

The first line contains **T** denoting the number of testcases. T testcases follow. Each case contains a string **S** containing characters.

Output:

For each test case, in a new line, output a single line containing the reversed String.

Constraints:

$1 \leq T \leq 100$

$1 \leq |S| \leq 2000$

Example:

Input:

2

i.like.this.program.very.much

pqr.mno

Output:

much.very.program.this.like.i

mno.pqr

Permutations of a given string

Given a string S. The task is to print all permutations of a given string.

Input:

The first line of input contains an integer **T**, denoting the number of test cases. Each test case contains a single string **S** in capital letter.

Output:

For each test case, print all permutations of a given string **S** with single space and all permutations should be in lexicographically increasing order.

Constraints:

$$1 \leq T \leq 10$$

$$1 \leq \text{size of string} \leq 5$$

Example:**Input:**

2

ABC

ABSG

Output:

ABC ACB BAC BCA CAB CBA

ABGS ABSG AGBS AGSB ASBG ASGB BAGS BASG BGAS BGSA BSAG BSGA

GABS GASB GBAS GBSA GSAB GSBA SABG SAGB SBAG SBGA SGAB SGBA

Explanation:

Testcase 1: Given string ABC has permutations in 6 forms as ABC, ACB, BAC, BCA, CAB and CBA .

Longest Palindrome in a String

Given a string S, find the longest palindromic substring in S. Substring of string S: $S[i \dots j]$ where $0 \leq i \leq j < \text{len}(S)$. Palindrome string: A string which reads the same backwards. More formally, S is palindrome if $\text{reverse}(S) = S$. In case of conflict, return the substring which occurs first (with the least starting index).

NOTE: Required Time Complexity $O(n^2)$.

Input:

The first line of input consists number of the testcases. The following T lines consist of a string each.

Output:

In each separate line print the longest palindrome of the string given in the respective test case.

Constraints:

$$1 \leq T \leq 100$$

$$1 \leq \text{Str Length} \leq 10^4$$

Example:**Input:**

1

aaaabbbaa

Output:

aabbbaa

Explanation:

Testcase 1: The longest palindrome string present in the given string is "aabbbaa".

Recursively remove all adjacent duplicates

Given a string **s**, recursively remove adjacent duplicate characters from the string **s**. The output string should not have any adjacent duplicates.

Input:

The first line of input contains an integer **T**, denoting the no of test cases. Then **T** test cases follow. Each test case contains a string **str**.

Output:

For each test case, print a new line containing the resulting string.

Constraints:

$1 \leq T \leq 100$

$1 \leq \text{Length of string} \leq 50$

Example:**Input:**

2

geeksforgeek

acaaabbbacdddd

Output:

gksforgk
acac

Check if string is rotated by two places

Given two strings a and b. The task is to find if a string 'a' can be obtained by rotating another string 'b' by 2 places.

Input:

The first line of input contains an integer T denoting the no of test cases. Then T test cases follow. In the next two line are two string a and b.

Output:

For each test case in a new line print 1 if the string 'a' can be obtained by rotating string 'b' by two places else print 0.

Constraints:

$1 \leq T \leq 50$

$1 \leq \text{length of } a, b < 100$

Example:**Input:**

2
amazon
azonam
geeksforgeeks
geeksgeeksfor

Output:

1
0

Explanation:

Testcase 1: amazon can be rotated anti-clockwise by two places, which will make it as azonam.

Testcase 2: geeksgeeksfor can't be formed by any rotation from the given word geeksforgeeks.

