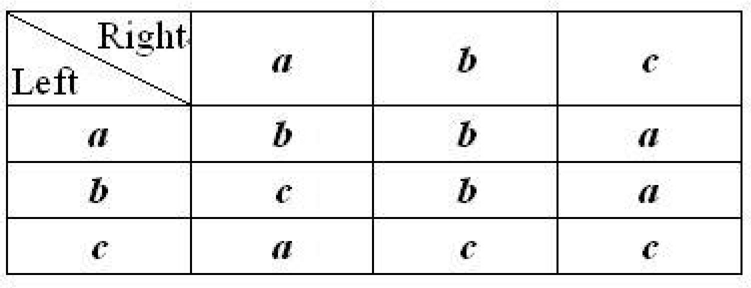
There is a special multiplication operator such that



Right Left	*a*	*b*	*c*
a	*b*	*b*	*a*
b	*c*	*b*	*a*
c	*a*	*c*	*c*

Table of special multiplication operation

Thus $ab = b$, $ba = c$, $bc = a$, $cb = c$, $\ldots$

For example, you are given the string `bbbba` and the character `a`,

```
(b(bb))(ba) = (bb)(ba)   [as bb = b]
            = b(ba)      [as bb = b]
            = bc         [as ba = c]
            = a          [as bc = a]
```

By adding suitable brackets, `bbbba` can produce `a` according to the above multiplication table.

You are asked to **write a program** to show the morphing steps of a string into an expected character, or otherwise, output '`None exist!`' if the given string cannot be morphed as expected.

Input

The first line of the input file gives the number of test cases. Each case consists of two lines. The first line is the starting string which has at most 100 characters. The second line is the target character. All characters in the input are within the range of `a-c`.

Output

For each test case, your output should consist several lines, showing the morphing steps of a string into the character. In case there are more than one solution, always try to start the morphing from the left. Print a blank line between consecutive sets of output.

Sample Input

```
2
bbbba
a
bbbba
a
```

Sample Output

```
bbbba
bbba
bba
bc
a

bbbba
bbba
bba
bc
a
```