Rotating Display

`display.c`, `display.cpp`, `display.c11`, `Display.java`, `display.py`

Wendy is finishing her summer job in the lab testing a new 3D printed robot which is being taught to manipulate small objects.

To test the robot's abilities, a simple device called the display is used. It is a thin transparent square array of $N \times N$ square slots. Each slot contains a token, in the shape of an ASCII character (for better recognition), which is held in place in the slot by small magnets. The display can be rotated by 90° around the axis perpendicular to its surface or flipped by 180° around one of its four axes parallel to the display surface.

The robot simulates display rotations and flips by the following process. It removes the tokens from their slots and puts them back into different slots so that the contents of the display looks exactly as if the whole display was rotated or flipped. If, for achieving the desired effect, it is necessary to rotate or to flip particular tokens in their new positions the robot does it as well. The display remains stationary during the whole process.

For example, if the upper left corner of the display contains a token which looks like symbol “`<`” (less than) then after flipping the display around the vertical axis this token is moved to the upper right corner where it looks like the symbol “`>`” (greater than). Then, after left rotation the same token is moved back to the upper left corner where it looks like the symbol “`^`” (caret).

Wendy has programmed the robot to perform a long sequence of successive flips and rotations. To check the correctness of the robot's algorithms, she needs to know in advance how the display should look when the robot finishes its work.

Input Specification

We suppose that a token on the display can be shaped as any of the following so called symmetric characters: “`<`”, “`>`”, “`^`”, “`v`”, “`o`”, “`x`”, “`|`”, “`-`”, “`/`”, “`\`”. When a symmetric character is rotated or flipped it either remains the same or it becomes another symmetric character whose shape is the most similar to the rotated/flipped one.

There are more test cases. Each case starts with a line containing one integer N ($1 \leq N \leq 100$). Next, there are N lines representing the initial state of the display. Each line contains a string which consists of exactly N symmetric characters. Each character represents one slot on the display and the order of symbols in the input corresponds to the order of the tokens on the display. No slot on the display is empty. After N lines, there is a line with a command string specifying the flips and rotations that has to be carried out. A command string consists of command characters, each command character specifies one rotation or flip as follows: “`<`” (rotation left), “`>`” (rotation right), “`-`” (flip around horizontal axis), “`|`” (flip around vertical axis), “`\`” (flip around main diagonal), “`/`” (flip around anti-diagonal). Two successive command characters are separated by single space. The robot has to respect the order of commands in the command string. The number of commands is always positive and at most 10^6.

The decimal ASCII codes of the characters relevant to this problem are 45 (“-”), 47 (“/”), 60 (“<”), 62 (“>”), 92 (“\”), 94 (“^”), 111 (“o”), 118 (“v”), 120 (“x”), 124 (“|”).

Output Specification

For each test case, print N lines specifying the final position and orientation of the tokens on the display. The output format of the display representation is identical to the input format except for the size of the display which should not be printed.

Sample Input

```
3
o^-
/v|
vx^
< |
5
x>-o\
voooo
|ooo/
ooo/v
\o/vv
| \ |
```

Output for Sample Input

```
>-|
x<>
</o
<</o\
</ooo
/ooo|
oooo^
\o-<x
```