

Submit: `quad.c` / `quad.cpp` / `quad.pas`
 Input: `svio`
 Output: `svio`

Time limit: 2 s
 Memory limit: 16 MB
 Points: 100

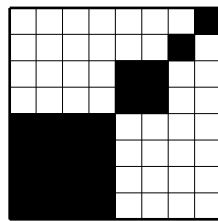
The European Union has recently founded an Extraterrestrial Signals Agency, a new organization whose primary purpose is to communicate with alien civilizations. Since well informed sources have confirmed that aliens really exist, circle around the Earth in their flying saucers¹ and sometimes land and create crop circles, the ESA has decided to try to draw the aliens' attention by drawing large pictures on the face of the Earth, which will be visible from the space.

The pictures are drawn in a $2^k \times 2^k$ grid, each square of the grid is either completely black or completely white. Since the total area of each picture is immense and it contains a lot of large monochromatic regions, they have decided to describe the pictures by the so-called quadrant codes:

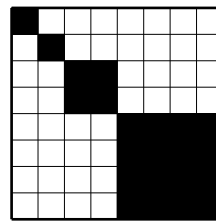
- A completely black picture has code **a**.
- A completely white picture has code **b**.
- Any other picture is encoded by $(q_1q_2q_3q_4)$, where the q_i 's are codes of the picture's quadrants:

| | |
|-------|-------|
| q_1 | q_2 |
| q_3 | q_4 |

For example:



$(a(a(abba)ba)ba)$



$((((baab)aab)aab)$

The ufologists from the ESA have designed a set of pictures and they now need to know how are the pictures visible from a saucer performing complicated manoeuvres. They have already discovered that each manoeuvre causes the picture seen to be transformed in one of the following ways:

- L – rotated counter-clockwise
- R – rotated clockwise
- X – flipped horizontally
- Y – flipped vertically
- N – negated (i.e., black and white swapped).

Your task is to read a quadrant code of a picture and a sequence of transformations T_1, \dots, T_m and produce a quadrant code of the resulting picture (i.e., the original picture transformed first by T_1 , then the result transformed by T_2 , and so on).

Input:

The program's input consists of two (possibly very long) strings read using functions of the `svio` library. The first string contains the quadrant code of the given picture (at most 10 000 000 characters long), the second string contains the sequence of transformations to be applied (again at most 10 000 000 characters).

Output:

The program's output is a single string written using the `svio` library. This string should correspond to the quadrant code of the transformed picture.

Example 1:

| | |
|----------------------|-----------------------|
| <code>quad.in</code> | <code>quad.out</code> |
| <code>a</code> | <code>b</code> |
| <code>LXYRN</code> | |

Example 2:

| | |
|-------------------------------|--------------------------------|
| <code>quad.in</code> | <code>quad.out</code> |
| <code>(a(a(abba)ba)ba)</code> | <code>((((baab)aab)aab)</code> |
| <code>RRR</code> | |

¹ Slowly enough to avoid spilling the sauce...