Problem A

"Our high speed camera failed at the most inappropriate moment," said the director of the ZOO. "This sequence with the falcon hurtling towards the ground at 250 km/h is absolutely stunning. I had hopes that we could use the last frame as a promotion picture, it would look great with the autumn trees in the background. But the falcon is too high, even in this very last frame caught by the camera before it broke."

"Cut out the falcon from the picture in Photoshop and just move it downwards," said the falconer. "It's a routine photo manipulation."

"That would be unnatural," objected the director. "We cannot show the public such obviously doctored pictures."

"On the contrary, that would be quite natural," replied the falconer. "Look, the falcon in such speed does not change its orientation so much, its shape in the picture remains virtually the same in a few consecutive frames. So if you move it down artificially it would still be a very good approximation of the natural situation which really occurred during the filming."

After some hesitation, the director agreed with the proposition.

You are given two last frames of the camera with the silhouette of the falcon in both frames. The background in the frames is identical, only the silhouette of the falcon is at a different position in both frames. The falcon is moving at a constant speed and the time between consecutive camera frames is also constant. Your task is to reconstruct the missing next frame in which the position of the falcon silhouette is changed according to its speed and to the speed of the camera. The background in the new frame should be the same as the background in the previous two frames.

Input Specification

There are more test cases. Each test case starts with a line containing two integers M, N ($2 \leq M, N \leq 1\,000$) and a printable ASCII character C enclosed in single quotes. The values on the line are separated by spaces. Next, there are M lines, one empty line, and other M lines. The first M lines represent the first frame, the last M lines represent the second frame. Each nonempty line contains string of exactly N printable ASCII characters. Each character represents one pixel of the original frame. Each frame contains a complete silhouette of the falcon. In both frames all silhouette pixels are represented by the character C and all pixels which do not belong to the silhouette are represented by characters other than C. The pixels of the silhouettes in both frames do not overlap even partially, in other words, no coordinates of a pixel of the silhouette in the first frame are the same as the coordinates of any pixel of the silhouette in the second frame. The shapes of the silhouettes in both frames are identical. The silhouette in any frame can be shifted by some number of pixels horizontally and/or vertically so that its position exactly matches the position of the silhouette in the other frame. The silhouettes

do not rotate. For various technical reasons the silhouette image might not be connected, it may comprise of more disconnected regions in the frame.

A printable ASCII character is an element of the subset of ASCII characters starting with the exclamation mark character (`'!'`, ASCII code 33 in decimal) and ending with the tilde character (`'~'`, ASCII code 126 in decimal).

There is a blank line between successive cases. The input is terminated by a line containing "`0 0 ' '`".

Output Specification

For each test case, print a picture frame consisting of M lines with N characters each. The frame should represent the result of exact extrapolation of the falcon's movement based on the two input frames. If the silhouette image in the second input frame is shifted horizontally and vertically by some number of pixels relatively to the first input frame then the silhouette image in the result frame should be shifted horizontally and vertically by the same number of pixels relatively to the second frame. It is possible that the falcon's silhouette might appear in the frame picture only partially or it may not appear there at all. Print one empty line after each case.

Sample Input

```
2 2 'X'
X^
--

.X
--

3 12 'A'
ABABABABABAC
BABABABABABB
ABABABABABAB

BABABABABABA
BBABABABABAB
BABABABABABA

6 26 '>'
../||\........OO......\|/.
>//||\\.....OOOOOO....-O-.
/>>>|\\\....OOOOOO..../|\.
..>||........OOOO.........
...||.........||..........
||||||||||||||||||||||||||

../||\.....>..OO......\|/.
.//||\\.....>>>OOO....-O-.
///||\\\....O>OOOO..../|\.
...||........OOOO.........
...||.........||..........
||||||||||||||||||||||||||

0 0 ' '
```

Output for Sample Input

```
.^
--

BBABABABABAC
BBBABABABABA
BBABABABABAB

../||\........OO......\>>>
.//||\\.....OOOOOO....-O>.
///||\\\....OOOOOO..../|\.
...||........OOOO.........
...||.........||..........
||||||||||||||||||||||||||
```