B. Tic-Tac-Toe

time limit per test: 2 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

Two bears are playing tic-tac-toe via mail. It's boring for them to play usual tic-tac-toe game, so they are a playing modified version of this game. Here are its rules.

The game is played on the following field.

Players are making moves by turns. At first move a player can put his chip in any cell of any small field. For following moves, there are some restrictions: if during last move the opposite player put his chip to cell with coordinates (x_l, y_l) in some small field, the next move should be done in one of the cells of the small field with coordinates (x_l, y_l). For example, if in the first move a player puts his chip to lower left cell of central field, then the second player on his next move should put his chip into some cell of lower left field (pay attention to the first test case). If there are no free cells in the required field, the player can put his chip to any empty cell on any field.

You are given current state of the game and coordinates of cell in which the last move was done. You should find all cells in which the current player can put his chip.

A hare works as a postman in the forest, he likes to foul bears. Sometimes he changes the game field a bit, so the current state of the game could be unreachable. However, after his changes the cell where the last move was done is not empty. You don't need to find if the state is unreachable or not, just output possible next moves according to the rules.

Input

First 11 lines contains descriptions of table with 9 rows and 9 columns which are divided into 9 small fields by spaces and empty lines. Each small field is described by 9 characters without spaces and empty lines. character "x" (ASCII-code 120) means that the cell is occupied with chip of the first player, character "o" (ASCII-code 111) denotes a field occupied with chip of the second player, character "." (ASCII-code 46) describes empty cell.

The line after the table contains two integers x and y ($1 \le x, y \le 9$). They describe coordinates of the cell in table where the last move was done. Rows in the table are numbered from up to down and columns are numbered from left to right.

It's guaranteed that cell where the last move was done is filled with "x" or "o". Also, it's guaranteed that there is at least one empty cell. It's not guaranteed that current state of game is reachable.

Output

Output the field in same format with characters "!" (ASCII-code 33) on positions where the current player can put his chip. All other cells should not be modified.

Examples

input
... ... x.. ... . 6 4
output

```
... ... ...
... ... ...
... ... ...

... ... ...
... ... ...
... x.. ...

!!! ... ...
!!! ... ...
!!! ... ...
```

input

```
xoo x.. x..
ooo ... ...
ooo ... ...

x.. x.. x..
... ... ...
... ... ...

x.. x.. x..
... ... ...
... ... ...
7 4
```

output

```
xoo x!! x!!
ooo !!! !!!
ooo !!! !!!

x!! x!! x!!
!!! !!! !!!
!!! !!! !!!

x!! x!! x!!
!!! !!! !!!
!!! !!! !!!
```

input

```
o.. ... ...
... ... ...
... ... ...

... xxx ...
... xox ...
... ooo ...

... ... ...
... ... ...
... ... ...
5 5
```

output

```
o!! !!! !!!
!!! !!! !!!
!!! !!! !!!

!!! xxx !!!
!!! xox !!!
!!! ooo !!!

!!! !!! !!!
!!! !!! !!!
```

```
!!! !!! !!!
```

Note

In the first test case the first player made a move to lower left cell of central field, so the second player can put a chip only to cells of lower left field.

In the second test case the last move was done to upper left cell of lower central field, however all cells in upper left field are occupied, so the second player can put his chip to any empty cell.

In the third test case the last move was done to central cell of central field, so current player can put his chip to any cell of central field, which is already occupied, so he can move anywhere. Pay attention that this state of the game is unreachable.