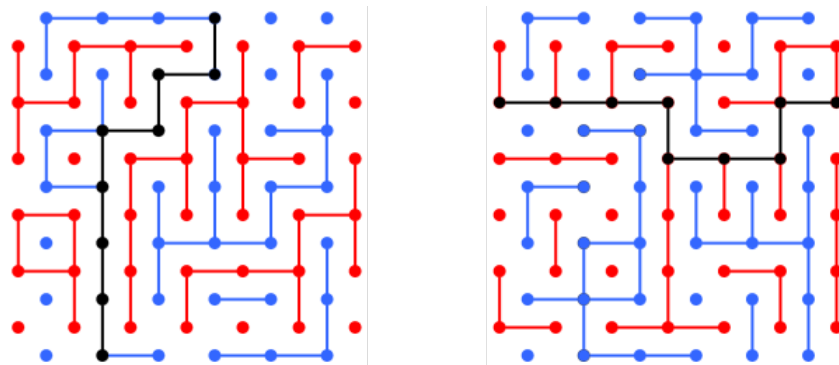


# PipeLayer



Pipelayer is a game played on two colored, overlapping grids of size 6x7 and 7x6 that are slightly offset from each other. The image below shows the two grids that are colored red and blue.

- Player 1 & Player 2 play on the red and blue grids respectively.
- The players take turns moving by connecting two dots of their own color.
- A player can only connect dots that are horizontally or vertically adjacent and also only dots of their own color.
- No move may draw across another move.
- To win, a player must make an unbroken connection from one side of the board to the other in the longer direction for his color of dots. For the red player this is from left to right and for the blue player, it is from top to bottom.



The first image shows a win state for the blue player and the second image shows a win state for the red player. The winning path for each is shown in black.

In this problem, two overlapping grids 6x7 and 7x6 are represented as a single 13x13 grid with red dots **r** at the left and right end and the blue dots **b** at the top and bottom end of the board. An empty slot is represented by a '.' which can be replaced by '|' or '-' to connect the dots above and below or the dots towards the left and the right side of the empty slot respectively.

The grid follows the four rules given below. All indices are 0-indexed.

1. Positions (0, 0), (0, 12), (12, 0) and (12, 12) are marked as '\*' (invalid)
2. Positions (i, j) where i is odd and j is even will always be a red dot.
3. Positions (i, j) where i is even and j is odd will always be a blue dot.
4. Positions (i, j) where both (i, j) are even or (i, j) are odd can be '.' (empty, ascii value: 46) or '|' (vertical pipe, ascii value: 124) or '-' (horizontal pipe, ascii value:45).

## Example Input

```
r
*b.b.b.b.b.b*
r|r.r.r.r.r.r
.b.b.b.b.b.b.
r|r.r.r.r.r.r
.b-b-b.b.b.b.
r.r.r.r.r.r.r
.b.b.b.b.b.b.
r-r-r.r.r.r.r
.b.b|b.b.b.b.
r.r.r.r.r.r.r
.b.b.b.b.b.b.
r.r.r.r.r.r.r
*b.b.b.b.b.b*
```

## Example Output

7 9

The board results in the following state after the above move.

```
*b.b.b.b.b.b*
r|r.r.r.r.r.r
.b.b.b.b.b.b.
r|r.r.r.r.r.r
.b-b-b.b.b.b.
r.r.r.r.r.r.r
.b.b.b.b.b.b.
r-r-r.r.r.r.r
.b.b|b.b.b.b.
r.r.r.r.r.r.r
.b.b.b.b.b.b.
r.r.r.r.r.r.r
*b.b.b.b.b.b*
```

## Explanation

It is red's turn, so the empty slot will be replaced by - to connect the two red dots at (7,8) and (7,10) horizontally.

The function **nextMove** takes a character player ('r' or 'b') and a 13x13 board array as input. Complete the function to print (x,y) (2 space-separated integers). The empty slot is replaced by '|' or '-' which connects the current player's dots horizontally or vertically. The top left co-ordinate is (0,0) and the bottom right co-ordinate is (12,12).

The initial board configuration is:

```
*b.b.b.b.b.b*
r.r.r.r.r.r.r
.b.b.b.b.b.b.
r.r.r.r.r.r.r
.b.b.b.b.b.b.
r.r.r.r.r.r.r
.b.b.b.b.b.b.
r.r.r.r.r.r.r
.b.b.b.b.b.b.
r.r.r.r.r.r.r
.b.b.b.b.b.b.
r.r.r.r.r.r.r
*b.b.b.b.b.b*
```