

Description

It's time to make our game interactive! Now we're going to add the ability for a user to make a move. To do this, we need to divide the grid into cells. Suppose the top left cell has the coordinates (1, 1) and the bottom right cell has the coordinates (3, 3):

```
(1, 1) (1, 2) (1, 3)
(2, 1) (2, 2) (2, 3)
(3, 1) (3, 2) (3, 3)
```

The program should ask the user to enter the coordinates of the cell where they want to make a move.

In this stage, the user plays as X, not O. Keep in mind that the first coordinate goes from top to bottom and the second coordinate goes from left to right. The coordinates can include the numbers 1, 2, or 3.

What happens if the user enters incorrect coordinates? The user could enter symbols instead of numbers, or enter coordinates representing occupied cells or cells that aren't even on the grid. You need to check the user's input and catch possible exceptions.

Objectives

The program should work as follows:

1. Get the initial 3x3 grid from the input as in the previous stages. Here the user should input 9 symbols representing the field, for example, _XX00_0X_.
2. Output this 3x3 grid as in the previous stages.
3. Prompt the user to make a move. The user should input 2 coordinate numbers that represent the cell where they want to place their X, for example, 1 1.
4. Analyze user input. If the input is incorrect, inform the user why their input is wrong:
Print This cell is occupied! Choose another one! if the cell is not empty.
Print You should enter numbers! if the user enters non-numeric symbols in the coordinates input.
Print Coordinates should be from 1 to 3! if the user enters coordinates outside the game grid.
Keep prompting the user to enter the coordinates until the user input is valid.
5. If the input is correct, update the grid to include the user's move and print the updated grid to the console.

To summarize, you need to output the field 2 times: once before the user's move (based on the first line of input) and once after the user has entered valid coordinates (then you need to update the grid to include that move).

Do not delete the code you wrote in the previous stage! You will need it in the final stage of this project, so now you can just comment out a part of it.

Examples

The greater-than symbol followed by a space (>) represents the user input. Note that it's not part of the input.

Example 1:

```
> X_X_0____
-----
| X  X |
|  0  |
|      |
-----
> 3 1
-----
| X  X |
|  0  |
| X    |
-----
```

Example 2:

```
> _XX00_0X_
-----
|   X X |
| 0 0   |
| 0 X   |
-----
> 1 1
-----
| X X X |
| 0 0   |
| 0 X   |
-----
```

Example 3:

```
> _XX00_0X_
-----
|   X X |
```

```

| 0 0 |
| 0 X |
-----
> 3 3
-----
|   X X |
| 0 0   |
| 0 X X |
-----

```

Example 4:

```

> _XX00_0X_
-----
|   X X |
| 0 0   |
| 0 X   |
-----
> 2 3
-----
|   X X |
| 0 0 X |
| 0 X   |
-----

```

Example 5:

```

> _XX00_0X_
-----
|   X X |
| 0 0   |
| 0 X   |
-----
> 3 1
This cell is occupied! Choose another one!
> 1 1
-----
| X X X |
| 0 0   |
| 0 X   |
-----

```

Example 6:

```
> _XX00_0X_
```

```
-----  
|   X X |  
| 0 0   |  
| 0 X   |  
-----
```

```
> one
```

You should enter numbers!

```
> one one
```

You should enter numbers!

```
> 1 1
```

```
-----  
| X X X |  
| 0 0   |  
| 0 X   |  
-----
```

Example 7:

```
> _XX00_0X_
```

```
-----  
|   X X |  
| 0 0   |  
| 0 X   |  
-----
```

```
> 4 1
```

Coordinates should be from 1 to 3!

```
> 1 4
```

Coordinates should be from 1 to 3!

```
> 1 1
```

```
-----  
| X X X |  
| 0 0   |  
| 0 X   |  
-----
```