

Description

Now you need to implement human moves.

Suppose the bottom left cell has the coordinates (1, 1) and the top right cell has the coordinates (3, 3) like in this table:

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

The program should work in the following way:

1. Get the 3x3 field from the input as in the previous stages,
2. Output this 3x3 field with cells before the user's move,
3. Then ask the user about his next move,
4. Then the user should input 2 numbers that represent the cell on which user wants to make his X or O. (9 symbols representing the field would be on the first line and these 2 numbers would be on the second line of the user input),
5. Then output the table including the user's most recent move.

Do not delete code that checks for table state; it will be useful in the future.

Note that in this stage user moves as X, not O. Keep in mind that the first coordinate goes from left to right and the second coordinate goes from bottom to top. Also, notice that coordinates start with 1 and can be 1, 2 or 3.

But what if the user enters incorrect coordinates? The user could enter symbols instead of numbers or enter coordinates representing occupied cells. You need to prevent all of that by checking a user's input and catching possible exceptions.

The program should also check user input. If the user input is unsuitable, the program should ask him to enter coordinates again.

So, you need to output a field from the first line of the input and then ask the user to enter a move. Keep asking until the user enters coordinate that represents an empty cell on the field and after that output the field with that move. You should output the field only 2 times: before the move and after a legal move.

Examples

The examples below shows how your program should work.

Example 1:

Enter cells: X_X_O____

```
-----  
| X  X |  
|  O  |  
|-----|
```

Enter the coordinates: 1 1

```
-----  
| X  X |  
|  O  |  
| X   |  
|-----|
```

Example 2:

Enter cells: _XX00_OX_

```
-----  
|   X X |  
|  O O  |  
|  O X  |  
|-----|
```

Enter the coordinates: 1 3

```
-----  
| X X X |  
|  O O  |  
|  O X  |  
|-----|
```

Example 3:

Enter cells: _XX00_OX_

```
-----  
|   X X |  
|  O O  |  
|  O X  |  
|-----|
```

Enter the coordinates: 3 1

```
-----  
|   X X |  
|  O O  |  
|  O X X |  
|-----|
```

Example 4:

Enter cells: _XX00_OX_

```
-----  
|   X X |  
|  O O  |  
|  O X  |  
|-----|
```

Enter the coordinates: 3 2

```
-----  
|   X X |  
|  O O X |  
|  O X  |  
|-----|
```

Example 5:

Enter cells: _XX00_0X_

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

Enter the coordinates: 1 1

This cell is occupied! Choose another one!

Enter the coordinates: 1 3

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

Example 6:

Enter cells: _XX00_0X_

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

Enter the coordinates: one

You should enter numbers!

Enter the coordinates: one three

You should enter numbers!

Enter the coordinates: 1 3

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

Example 7:

Enter cells: _XX00_0X_

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

Enter the coordinates: 4 1

Coordinates should be from 1 to 3!

Enter the coordinates: 1 4

Coordinates should be from 1 to 3!

Enter the coordinates: 1 3

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