**Readme**

------

Team ID: 1025

Team Member:

Baosheng Feng: python code, test program

Jiahang Chen: minimax algorithm, test program, api

Program is in the ai-project2.ipynb.

------

**Environment**:

In this program, we use Jupyter with Python to write the project.

The program is in **ai-project2.ipynb**

To open this document, you need to install Jupyter.

Recommend: You can first install Anaconda and then install Jupyter notebook in Anaconda. Then, you can use Jupyter to open ai-project2.ipynb. To run the program, you should run all the cells in the document.

Python: 3.6

------

**Claims**:

Position: in this program, we will use the ‘position’ to define the position of “X” and “O” in the game. The position start from 1 to n\*n. For the first row, from left to right, it is 1,2,3…n. For the second row, it is n+1, n+2, …, n+n.

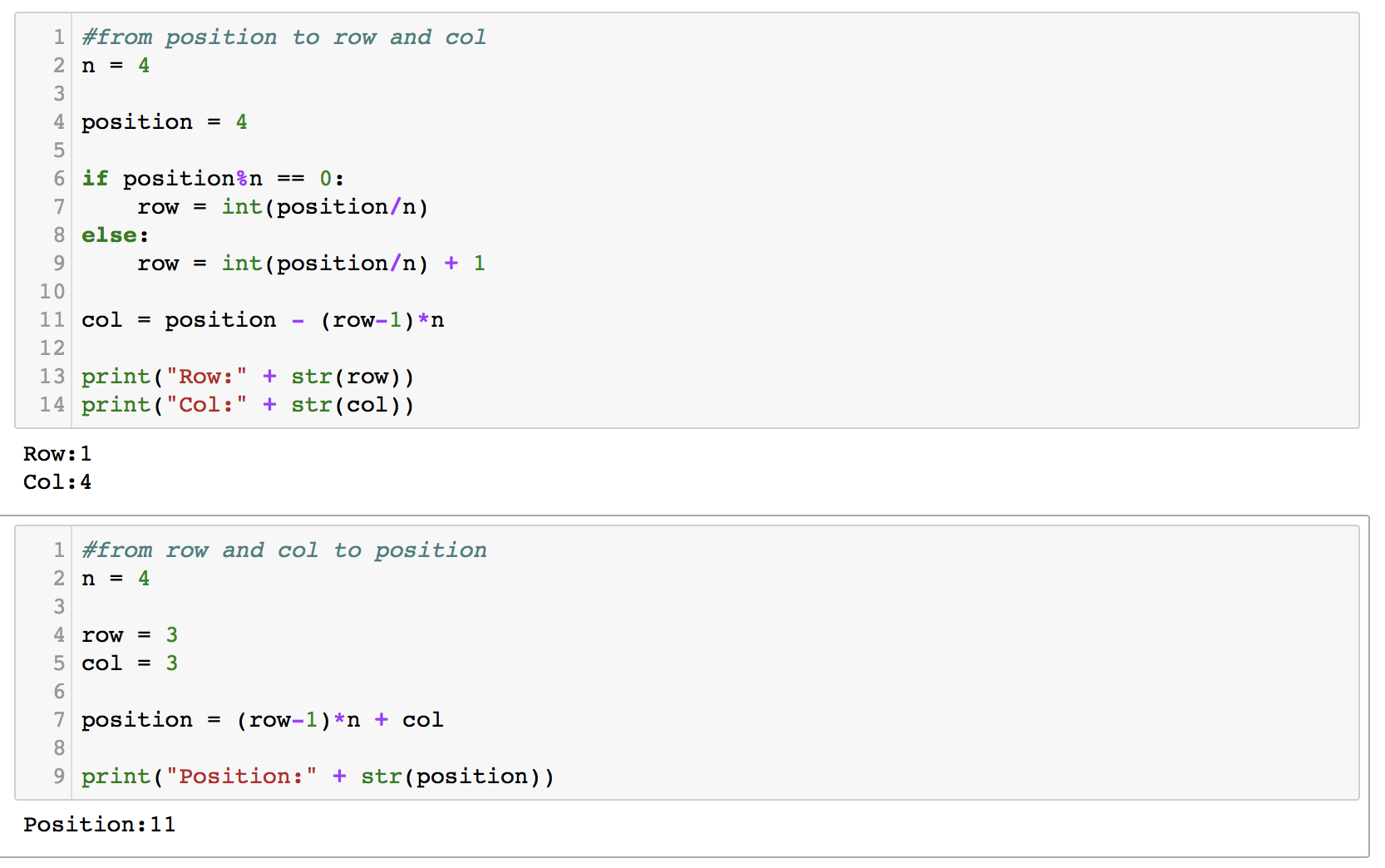
1 2 3 4 … n

n+1, n+2, …, n+n

…

Row & column: in this program, row and col all start from 1 to n

At the end of the program, there are to cell that can help you to transfer position to row&col or from row&col to position. (row, col and position are all start from 1 in this program).



You can change the value of n, position, row, and col in these two cells to get the results you want.

------

**How to use this program**:

Below is the picture that you can set the parameters and get the results.



In the setting area, you can set n, m, “X” player, and “O” player.

In the add moves area, you should add move and run the cell again, then it will print the current game.

We will use function best\_next\_move to find the next move position. In this function, you just need to modify the player, “X” or “O”, who is next.

Here is the screenshot for an example from start to end

1. first, “X” go first, you should run the second cell in the image, it will return the first move for “X”, it is [6,0]



2. then you should add this move to first cell like this. The first cell also print the result. Then, you can run the second cell, it will return the best move for “O”, which is the second move. It is [1,0]



3. Then you can add this move to the first cell and run the first cell. It will print the current game with two steps. Then you can run the second cell, which will return step 3 for “X”, which is [2,0]



4. Repeat the steps above , then you can get all the moves.



5. For the game above, it is ai vs. ai, because we use best\_next\_move to get the move. If you like to player person vs. ai, it is very simple that you can go “X” or “O” move yourself with add game.move to the first cell, then use best\_next\_move function to get the ai move.

6. result for the game

If it is tie. It will return [‘Tie’, 0], like this



If “X” wins, it will return [“X”, 500]

If “O” wins, it will return [“O”, -500]