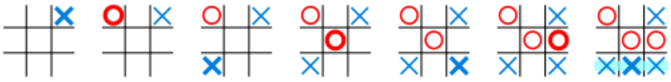
Points: 0.00 Rank: 129203

Tic tac toe

Problem Submissions Leaderboard Discussions

Tic-tac-toe is a pencil-and-paper game for two players, X (ascii value 88) and O (ascii value 79), who take turns marking the spaces in a 3×3 grid. The player who succeeds in placing three respective marks in a horizontal, vertical, or diagonal row wins the game. Empty space is represented by _ (ascii value 95), and the X player goes first.

Here is an example game won by the first player, X:



The function nextMove takes in a char player, and the 3x3 board as an array. Complete the function to print 2 space separated integers *r* and *c* which denote the row and column that will be marked in your next move. The top left position is denoted by (0,0).

How does it work?
Your code is run alternately with the opponent bot for every move.

Example input:

```
X
___
___
_XO
```

Example output:

```
1 0
```

Explanation:
The board results in the following state after the above move

```
___
X__
_XO
```

Submissions: 4640
Max Score: 10
Difficulty: Advanced

Rate This Challenge:
☆☆☆☆☆

More

Python 3

```
#CHETAN NAG.R 1BM22AI037
def nextMove(player, board):
    def checkWin(b, p):
        for r in range(3):
            if all(b[r][c] == p for c in range(3)):
                return True
        for c in range(3):
            if all(b[r][c] == p for r in range(3)):
                return True
        if all(b[i][i] == p for i in range(3)):
            return True
        if all(b[i][2 - i] == p for i in range(3)):
            return True
        return False

    opponent = 'O' if player == 'X' else 'X'

    # Check for winning move for player
    for r in range(3):
        for c in range(3):
            if board[r][c] == '_':
                board[r][c] = player
                if checkWin(board, player):
                    print(r, c)
                    return
                board[r][c] = '_'

    # Check for blocking move for opponent
    for r in range(3):
        for c in range(3):
            if board[r][c] == '_':
                board[r][c] = opponent
                if checkWin(board, opponent):
                    print(r, c)
                    return
                board[r][c] = '_'

    # Prefer position (1, 0) if available
    if board[1][0] == '_':
        print(1, 0)
        return

    # Check for center position if available
    if board[1][1] == '_':
        print(1, 1)
        return

    # Check for corner positions
    for r, c in [(0, 0), (0, 2), (2, 0), (2, 2)]:
        if board[r][c] == '_':
            print(r, c)
            return

    # Check for remaining positions
    for r, c in [(0, 1), (1, 0), (1, 2), (2, 1)]:
        if board[r][c] == '_':
            print(r, c)
            return

player = 'X'
board = [
    ['_', '_', '_'],
    ['_', '_', '_'],
    ['_', 'X', 'O']
]

nextMove(player, board)
```

Line: 1 Col: 1

Upload Code as File

Run Code Submit Code

Game 1 Game 2

JudgeBot vs chetan_ai22

X		
O		
		X

JudgeBot won the game

◀◀ ▶ ■ ▶▶

Download moves as file

Player: 2

Input

```
O
X__
O__
__X
```

Output

```
1 0
```

Error

Interview Prep | Blog | Scoring | Environment | FAQ | About Us | Support | Careers | Terms Of Service | Privacy Policy |