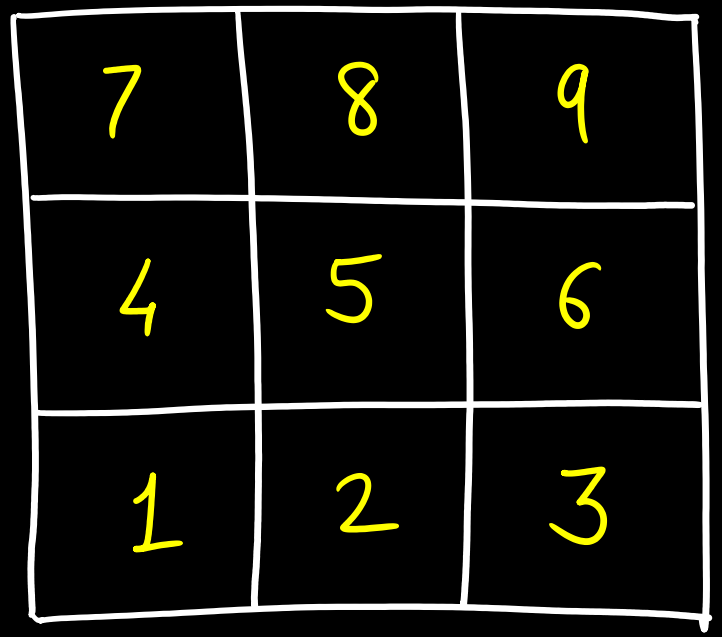
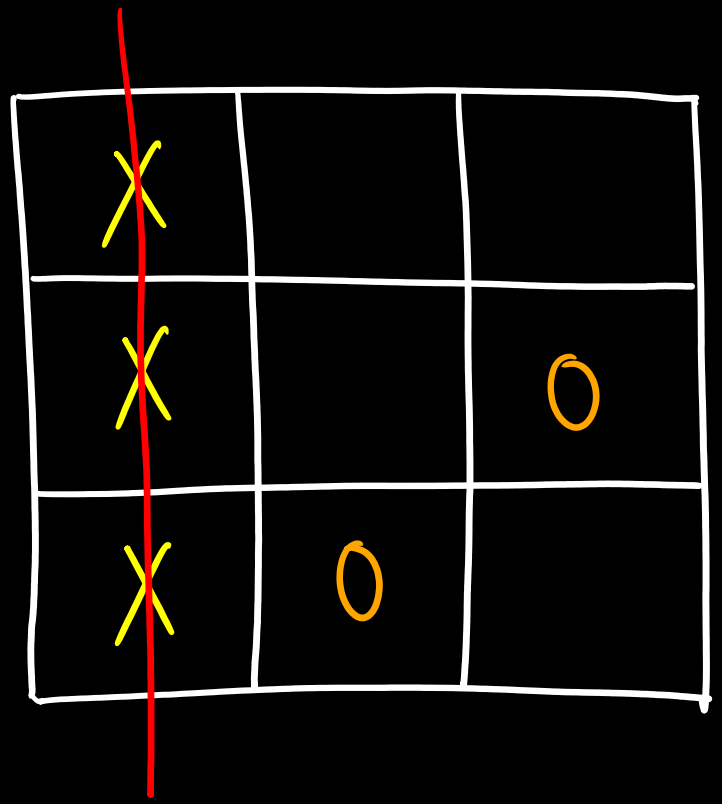
Tic - Tac - Toe :



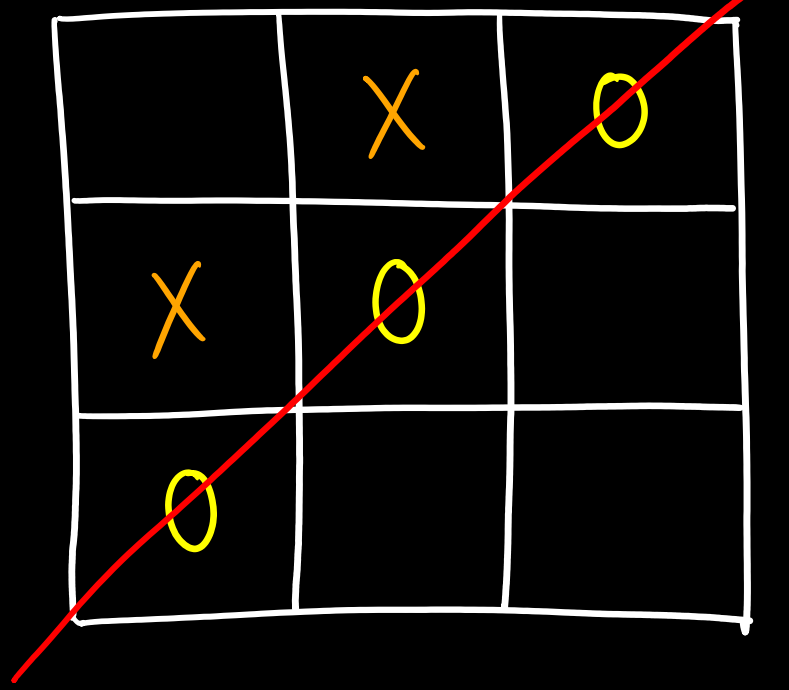
Number keys
&
Board location
mapping

Tic - Tac - Toe :



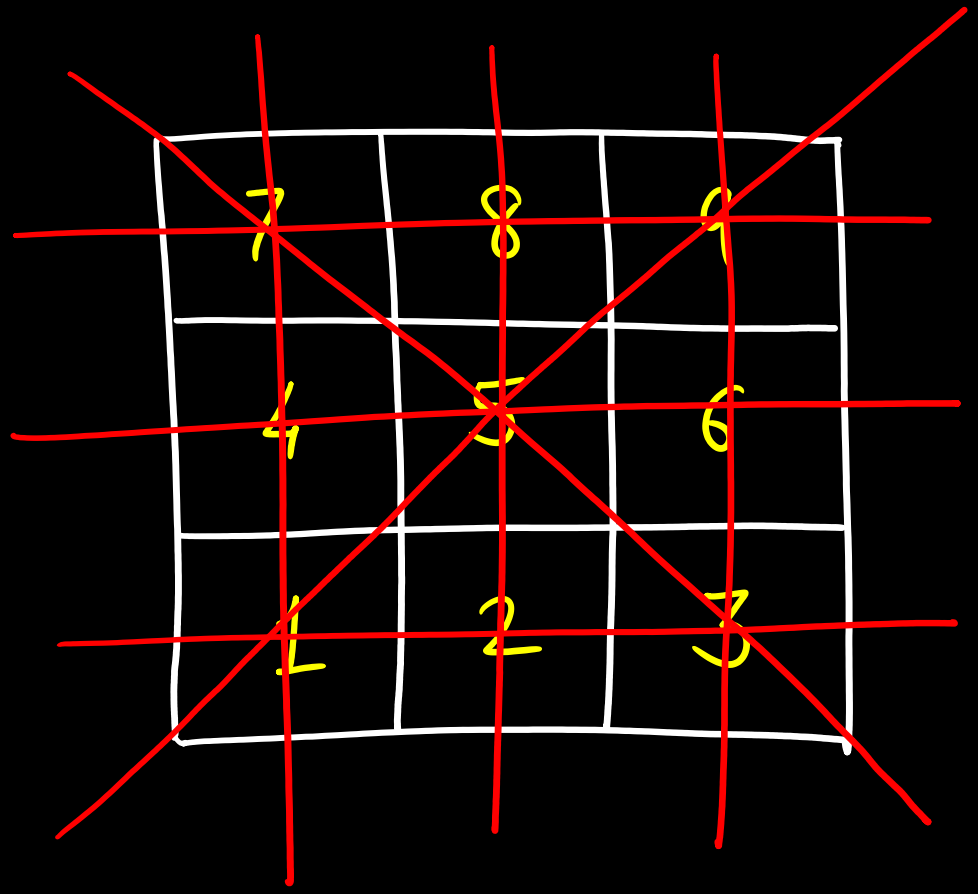
Player with mark = 'X' won the match.

Tic - Tac - Toe :



Player with mark = 'O' won the match.

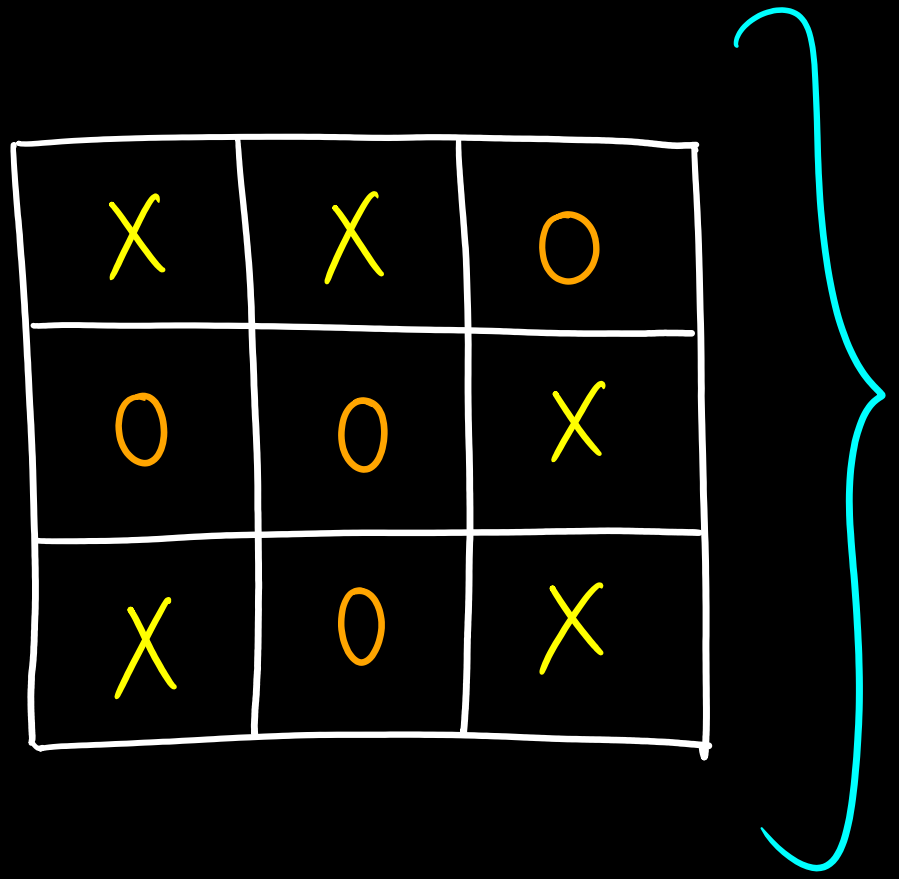
Winning Cases:



There are 8 winning cases.

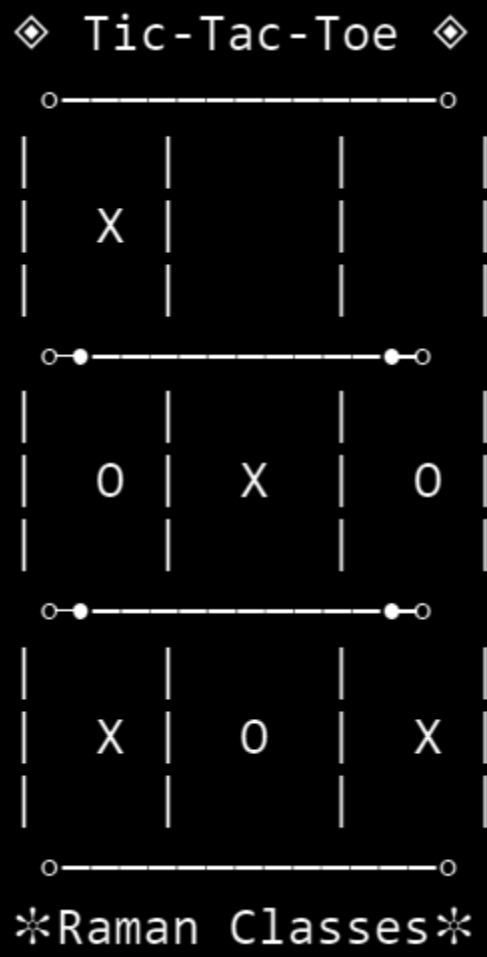
Check for winning condition every time after taking input.

Match Tie :



There is a tie.
Neither player 1 nor player 2 won.

Glimpse of the Board:



Tic - Tac - Toe steps:

Step 1: Create board for playing the game.

Step 2: Beautify your board by copy-pasting symbols from coolsymbol.com website.

Step 3: Use IPython.display.clear_output() in order to clear the previously displayed board output so that you can display the board with newly taken inputs.

Step 4: Display board using some dummy inputs.

Step 5: Take input from user & store it in the number list and call display() function.

Do this 9 times.

Step 6: Take inputs from the user & show X's & O's alternatively on the respective board positions.

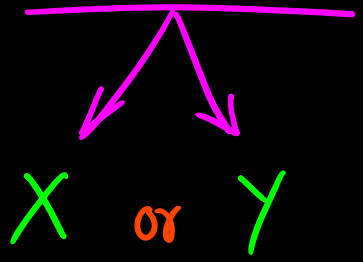
→ Number ∈ [1, 9]

Step 7: Write logic for winning condition.

Step 8: After every input, check the winning condition. There are 8 winning cases & if any one of these cases is true, we should print who won the game.

Step 9: Write match tie logic.

Step 10: Give player 1 freedom to choose the marker.



Based on player1_mark, decide player2_mark.

Step 11: Ask player names and modify the output as per the names.

Step 12: Check if player enters valid number or not.

→ If some mark is already present at some location (index from number list), we should not accept input till user enters a valid location from the board.

→ Player mark should be entered in the unfilled slot.

→ Do not overwrite a filled slot with the player's mark.

Step 13: Check for the correctness of user input while player1 is selecting a mark for himself/herself.

Step 14: Show game statistics.

- No. of matches played
- No. of matches won by player1
- No. of matches won by player2.
- No. of match ties

Step 15: Store the game statistics in a CSV file.

Step 16: Create beautiful plots showcasing individual game statistics or overall game statistics.

Step 17: Create a menu to show game statistics.