

## Lab: Tic Tac Toe

In the game of tic tac toe, players alternate placing Xs and Os on a board until one player gets three of their symbol in a row, column, or diagonal. We will store the tic tac toe board as a 2D list of integers, where a zero means a space on the board is empty, a 1 means it is filled with an X, and a -1 means it is filled with a O. This will make it easier later on to detect wins.

Write these functions for tic-tac-toe, assuming a board is stored as a 3 by 3 grid of numbers.

```
def print_board(board):
```

Prints a 3 by 3 tic tac toe board using X's and O's. (don't return anything)

```
def make_move(board, row, col, xo):
```

Makes a move on the board. The xo parameter will be either 1 for X, or -1 for O. (returns nothing) *[We don't need to return anything because when a list (1D or 2D) is passed to a function as an argument, modifying it directly will change the variable in the calling function.]*

```
def add_row(board, r):
```

Calculates AND RETURNS the sum of all the items in row r of the board.

```
def add_column(board, c):
```

Calculates AND RETURNS the sum of all the items in column c of the board.

```
def x_wins(board):
```

Returns True if player X has won the game, False otherwise.

Hint: use calls to add\_row and add\_column to check each row and column. (You'll also need to check the diagonals separately).

```
def o_wins(board):
```

Returns True if player O has one the game, False otherwise.

```
def tie(board):
```

Returns True if there's a tie (all squares are filled and nobody has won), False otherwise.

Write a main function to play the game. This function should

- Let players alternate turns to place Xs and Os. X should go first.
- Prevent players from placing Xs and Os in spots already occupied.
- Figure out if someone has won and which player it is.
- Figure out if there is a tie and end the game (don't keep prompting for moves if there is a tie).