Assignment: Build a Tic Tac Toe Backend

Objective

Create a clear, well-structured backend for Tic Tac Toe that manages game state, enforces move legality, and determines game outcomes (win or draw). This foundational exercise helps solidify concepts like data structures, functions, and control flow.

Part 1: Board Initialization

1. **Function:** `new_game()`
   Task: Return a fresh, empty 3×3 game board or any other data structure you choose.

Part 2: Player Representation

Define a simple way to distinguish players. For example:

- Player 'X'
- Player 'O'
  You may represent this with strings, enums (if available in your language), or constants.

Part 3: Making Moves

2. **Create a function for making moves**
   Tasks:
   - Check if `(row, col)` is inside the board and currently empty.
   - If valid, update the board with the player's symbol and return `True`.
   - Otherwise, return `False`.

Part 4: Winner (and Draw) Detection

3. **Function:** `check_winner(board)`
   Task: Check for a win or draw. Return:
   - `'X'` if X wins

    - `'O'` if O wins
    - `'Draw'` if the board is full without a winner
    - `None` if the game is still ongoing

Part 5: Game Flow (Optional Basic Loop)

Optionally, implement a game loop (e.g., in a `play()` function) that:

1. Starts a new game.
2. Alternates players calling `make_move()` until `check_winner()` returns a result.
3. Announces the winner or a draw at the end.