Objective: Practice on 2D Arrays

You have been provided with a template and some of the more complex functions necessary to create a fully working version of the game Connect Four using 2D Arrays. Your first task is to complete a basic version of the game by implementing the remaining functionality described in Task 1 below. You are then required to implement the advanced functionality described in Task 2 to complete a a fully functional version of the game.

If you are unaware of Connect 4, look it up. This version is designed for two human players, not user vs computer. Players pieces are X or O.

All of the functions in the template have all their necessary arguments so you only need to worry about writing the code that goes inside the functions.

Task#1:
Marks: 3
Summary: Implement the basic functionality of Connect Four

Details: In this task, you will write just enough code to make the Connect Four game playable. Your code should initialize the game board, then repeatedly ask each player in turn which column they would like to drop their game token into. After each turn you should print the board to the terminal. Using the template on the 1E3 website, add the following functions to the game:

• main() – Complete the main function, specifically the game loop which perpetuates the game until one player wins (does this sound like a while loop or a for loop?). This should also assign each player their turn.
  o This should print an updated board to the terminal after each player makes their move
  o Prompt user for input
  o Drop the piece in the appropriate column
  o Check whether that move results in a win
• init_board() – Creates the 2D array used to store the pieces on the board.
• drop_piece() – Allows a player to drop a piece into the specified column of the 2D array, positioning their piece on top of any existing pieces in the column. It must record the player's dropped piece in the array and return the row it ends up in to the calling function.

Task#2:
Marks: 2
Summary: Extend the game by preventing a player from being able to add a piece to a full column or a full game board. Add Game Over sequence.

Details: Add the print_game_over() function to check if the board is full via the board_full() function and if so, print the message “STALEMATE”. Extend the game loop in the main() function to also check if the board is full via the board_full() function.

It is rare that nobody will win at a game of Connect Four, but when writing code we need to think about the edge cases. Complete the following functions:

• print_game_over() – Prints the board to the screen via the print_board() function, the message “Game Over” and the name of the winning player, or “STALEMATE” if nobody wins.
• column_full() – Checks to see if the column the user selected is full
• board_full() – Checks to see if the game board is full
• Add finishing touches to the game, e.g. instructions, end game sequence error messages
  o We are looking for a complete game for full marks