- Connect 4 Specs
- Game Details
    - 2D Array of chars to represent board
    - ‘-’ is untaken
    - ‘X’ is human token
    - ‘O’ is computer token

- Get user input for board size (n and m) at least 7x6
    - If n and m are not the minimum keep demanding input from user

- Initialize board variable as nxm (2D) array of chars;
- Loop through and initialize each char as ‘-’;

- Init boolean turn; use to determine which player moves
- Define function game_won(char board[][]) //returns whether game is won
    - Checks if human or computer has 4 in a row
- After player goes
    - Turn = !turn //turn can be true for player 1 and false for player 2
    - Check if game_won(board)
    - Check if board is full by searching for a lack of ‘-’ if !game_won(board) => draw
- Game_won function
    - Diagonal win
        - For every row up to m - 5
            - For every column up to n - 5
                - Check 3 times if the cell up one right one is the same as the current cell
                    - Return win for current boolean player
    - Other diagonal win

            - For every row from 0 to m - 5
                - For every column from 4 to n - 1
                    - Check 3 times if the cell up one left one is the same as the current cell
                        - Return win for current boolean player
        - Horizontal win
            - For every row
                - For every column up to n - 5
                    - If cell is not ‘-’ and all 3 next cells are the same as the current cell
                        - Return win for current boolean player
        - Vertical win
            - For every column
                - For every row up to m - 5
                    - If cell is not ‘-’ and all 3 next cells are the same as the current cell
                        - Return win for current boolean player
- Play turn function
- Play_turn takes an integer for the column and places appropriate marker at the proper place in inputted column
    - Should guard against bad column input and keep demanding input until valid column is given
- Each turn should go as follows:
    - Play_turn
    - Game_won (breaks loop if true)
    - Flip player boolean