Functions Challenge 34:
Head to Head Tic-Tac-Toe App

Description:
You are responsible for writing a program that will allow two users to play a game of tic tac toe. Your program should follow the standard rules in which two players alternate turns putting their pieces, X or O, on a board. If a player has three pieces in a row, either vertically, horizontally, or diagonally, they are declared the winner. You will represent the tic tac toe board using the integers 1 through 9 for the 9 spaces on the board. An empty spot on the board will be represented by an underscore "_". For example, if a player would like to put a piece in the center of the board they would enter 5 as their move.

Tic-Tac-Toe
~~~~~~~~~
|| 1 || 2 || 3 ||
~~~~~~~~~
|| 4 || 5 || 6 ||
~~~~~~~~~
|| 7 || 8 || 9 ||
~~~~~~~~~

Tic-Tac-Toe
~~~~~~~~~
|| _ || _ || _ ||
~~~~~~~~~
|| _ || _ || _ ||
~~~~~~~~~
|| _ || _ || _ ||
~~~~~~~~~

Step By Step Guide:
Defining your functions

- Define a function draw_board() which takes one parameter; a list holding all of the current pieces on the tic tac toe board.
    - To draw the tic tac toe board for the current state of the game use a print statement to print out each individual row of the tic tac toe board.
    - The elements that take up the spots on the board should either be X, O, or _ which indicates that the spot is empty.
    - Print each element from the list at its correct location on the board.
    - There is no return value for this function.

- Define a function get_player_input() which takes two parameters, the player's character and a list holding all of the current pieces on the tic tac toe board.

    - Use a while loop to loop until the player the player gives an appropriate move.
    - Get user input for where they would like to place their piece.
    - If the given move is a space on the board:
        - If the space on the board does not currently have a piece there:
            - Return the move.
        - Else, inform the player that the spot has already been chosen.
    - Else, inform the user that their move is not a spot on the board.
    - You are looping through this function until you hit the return statement which will return the users move.

- Define a function place_char_on_board() which takes three parameters; the player's character, the player's desired move, and a list holding all of the current pieces on the tic tac toe board.
    - To simulate putting the players character on the board at the correct spot put the player's character in the list holding all the current pieces on the tic tac toe board at the index that corresponds to the spot chosen by the player.
    - There is no return for this function as we are directly changing our list.

- Define a function is_winner() which takes two parameters; A list holding all of the current pieces on the tic tac toe board and the player's character.
    - There are eight different ways you can win tic tac toe; 3 vertical, 3 horizontal, and 2 diagonal.
    - Use compound conditional statements to check for victory.
    - For example, if a player has their piece at position 1 and position 2 and position 3 then they have won.
    - Return a Boolean, True or False value based on whether or not a player has won.

Your main code:

- Create a variable player_1 and set it equal to "X".
- Create a variable player_2 and set it equal to "O".

- Create a list called c_list and set it equal to ['_']*9.
    - This will create a list of nine "_" which represent empty spots on the tic tac toe board.
    - As you play the game we will replace a "_" with the character of a player.
    - This is the list that will hold all of our moves throughout the game.
- Create a list called n_list and set it equal to the string values ["1", "2", "3", … "8", "9"].
    - This list will give a visual representation for the player as to how to enter their moves.
    - For example, 1, 2, and 3 correspond to the first row, 4, 5, and 6 correspond to the second row, and 7, 8, and 9 correspond to the final row.

- To begin the game draw a board with the numbers representing each board position by calling the draw_board() function and passing the n_list as an argument.

- Next, draw the current state of the board by calling the draw_board() function and passing the c_list as an argument.

- Use a while loop to play the game until there is a winner.
- Each iteration of your loop should do the following, first for player 1 then for player 2:
    - Get the player's move by calling the get_player_input() function.
    - Update the c_list by calling the place_char_on_board() function.
    - Draw the number board by calling the draw_board() function.
    - Draw the current state of the game board by calling the draw_board() function.
    - Check to see if there is a winner by calling the is_winner() function.
        - If there is print a statement declaring who won.
        - Break the while loop.
    - Check to see if there are no more empty spots on the board.
        - If so, declare a tie.

- Use at least 2 comments to describe sections of your code.
- "Chunk" your code so that is readable.
- Use appropriate and informative variable names.
- Format your output as below.

Example Output:

```
 Tic-Tac-Toe
~~~~~~~~~
|| 1 || 2 || 3 ||
~~~~~~~~~
|| 4 || 5 || 6 ||
~~~~~~~~~
|| 7 || 8 || 9 ||
~~~~~~~~~

 Tic-Tac-Toe
~~~~~~~~~
|| _ || _ || _ ||
~~~~~~~~~
|| _ || _ || _ ||
~~~~~~~~~
|| _ || _ || _ ||
~~~~~~~~~
X: Where would you like to place your piece (1 - 9): 5

 Tic-Tac-Toe
~~~~~~~~~
|| 1 || 2 || 3 ||
~~~~~~~~~
```

|| 4 || 5 || 6 ||
~~~~~~~~~
|| 7 || 8 || 9 ||
~~~~~~~~~

Tic-Tac-Toe
~~~~~~~~~
|| _ || _ || _ ||
~~~~~~~~~
|| _ || X || _ ||
~~~~~~~~~
|| _ || _ || _ ||
~~~~~~~~~
O: Where would you like to place your piece (1 - 9): 5
That spot has already been chosen. Try again.
O: Where would you like to place your piece (1 - 9): 15
That is not a spot on the board. Try again.
O: Where would you like to place your piece (1 - 9): 3

Tic-Tac-Toe
~~~~~~~~~
|| 1 || 2 || 3 ||
~~~~~~~~~
|| 4 || 5 || 6 ||
~~~~~~~~~
|| 7 || 8 || 9 ||
~~~~~~~~~

Tic-Tac-Toe
~~~~~~~~~
|| _ || _ || O ||
~~~~~~~~~
|| _ || X || _ ||
~~~~~~~~~
|| _ || _ || _ ||
~~~~~~~~~
X: Where would you like to place your piece (1 - 9): 6

Tic-Tac-Toe
~~~~~~~~~
|| 1 || 2 || 3 ||
~~~~~~~~~
|| 4 || 5 || 6 ||
~~~~~~~~~
|| 7 || 8 || 9 ||
~~~~~~~~~

~~~~~~~~~

Tic-Tac-Toe
~~~~~~~~~
|| _ || _ || O ||
~~~~~~~~~
|| _ || X || X ||
~~~~~~~~~
|| _ || _ || _ ||
~~~~~~~~~

O: Where would you like to place your piece (1 - 9): 9

Tic-Tac-Toe
~~~~~~~~~
|| 1 || 2 || 3 ||
~~~~~~~~~
|| 4 || 5 || 6 ||
~~~~~~~~~
|| 7 || 8 || 9 ||
~~~~~~~~~

Tic-Tac-Toe
~~~~~~~~~
|| _ || _ || O ||
~~~~~~~~~
|| _ || X || X ||
~~~~~~~~~
|| _ || _ || O ||
~~~~~~~~~

X: Where would you like to place your piece (1 - 9): 4

Tic-Tac-Toe
~~~~~~~~~
|| 1 || 2 || 3 ||
~~~~~~~~~
|| 4 || 5 || 6 ||
~~~~~~~~~
|| 7 || 8 || 9 ||
~~~~~~~~~

Tic-Tac-Toe
~~~~~~~~~
|| _ || _ || O ||
~~~~~~~~~
|| X || X || X ||
~~~~~~~~~

```
~~~~~~~~~
|| _ || _ || O ||
~~~~~~~~~
```

Congratulations! Player 1 wins!