Assignment

Objective:

- To have a better understanding of <u>list</u>
- Practice use <u>random</u> module
- Practice using functions, loops and if statement

Requirements: Create a board game and the board looks like this

```
------------------------
|  1  |  2  |  3  |  4  |
------------------------
|  5  |  6  |  7  |  8  |
------------------------
|  9  |  10 |  11 |  12 |
------------------------
|  13 |  14 |  15 |  16 |
------------------------
```

How the game should work has been introduced in class video, and here a few important points are listed:

1. The game has two players: computer & user of the program, computer will randomly select the position for its move. You can use your own choice of symbols to represent computer or user, for example, 'X" for user and "O" for computer.

2. Use list to store the **board** items and **available** positions; **both lists** contain the same items in the beginning of the game; ["1 ", "2 ", "3 ", "4 ", "5 ", "6 ", "7 ", "8 ", "9 ", "10", "11", "12", "13", "14", 15", "16"].

3. The program will <u>randomly</u> select who will make first move (computer or the user).

4. User can choose any available positions, computer will randomly select from the available positions; after a position is chosen, your program should remove the position item from **available** list, and change the position item in **board** list to the symbol representing computer/user. For example, in the first move, computer/user (randomly selected) choose "12", then

available = ["1 ", "2 ", "3 ", "4 ", "5 ", "6 ", "7 ", "8 ", "9 ", "10", "11", "13", "14", 15", "16"]

board = ["1 ", "2 ", "3 ", "4 ", "5 ", "6 ", "7 ", "8 ", "9 ", "10", "11", "X", "13", "14", 15", "16"].

5. There are totally <u>14 winning combinations</u>: (1, 2, 3, 4) (5, 6, 7, 8) (9, 10, 11, 12) (13, 14, 15, 16) (1, 5, 9, 13) (2, 6, 10, 14) (3, 7, 11, 15) (4, 8, 12, 16) (1, 6, 11, 16) (4, 7, 10, 13) (2, 7, 12) (3, 6, 9) (5, 10, 15) (8, 11, 14)

6. Use functions to simplify your job, for example, create **functions** to <u>print the board</u>, to <u>check winner</u>, to <u>validate user's choice</u>, etc.

7. Computer and user will move in turn.

8. After all positions are taken and still there is no winner, then it's a tie.