In [ ]:
```
import random

# Function to draw the Tic Tac Toe board
def drawBoard(board):
    print('   |   |')
    print(' ' + board[7] + ' | ' + board[8] + ' | ' + board[9])
    print('   |   |')
    print('-----------')
    print('   |   |')
    print(' ' + board[4] + ' | ' + board[5] + ' | ' + board[6])
    print('   |   |')
    print('-----------')
    print('   |   |')
    print(' ' + board[1] + ' | ' + board[2] + ' | ' + board[3])
    print('   |   |')

# Function to let the player choose 'X' or 'O'
def inputPlayerLetter():
    letter = ''
    while not (letter == 'X' or letter == 'O'):
        print('Do you want to be X or O?')
        letter = input().upper()

    if letter == 'X':
        return ['X', 'O']
    else:
        return ['O', 'X']

# Function to determine who goes first
def whoGoesFirst():
    user_choice = input("Choose who goes first (computer/player): ").lower()
    if user_choice == 'computer' or user_choice == 'player':
        return user_choice
    else:
        return whoGoesFirst()

# Function to ask if the player wants to play again
def playAgain():
    print('Do you want to play again? (yes or no)')
    play_again = input().lower()
    if play_again.startswith('y'):
        resetBoard()
        return True
    else:
        return False

# Function to make a move on the board
def makeMove(board, letter, move):
    board[move] = letter

# Function to check if a player has won
def isWinner(bo, le):
    return ((bo[7] == le and bo[8] == le and bo[9] == le) or  # across the top
            (bo[4] == le and bo[5] == le and bo[6] == le) or  # across the middle
```

```
            (bo[1] == le and bo[2] == le and bo[3] == le) or  # across the bottom
            (bo[1] == le and bo[5] == le and bo[9] == le) or  # diagonal
            (bo[3] == le and bo[5] == le and bo[7] == le) or  # diagonal
            (bo[1] == le and bo[4] == le and bo[7] == le) or  # down the left side
            (bo[2] == le and bo[5] == le and bo[8] == le) or  # down the middle
            (bo[3] == le and bo[6] == le and bo[9] == le))    # down the right side

# Function to create a copy of the board
def getBoardCopy(board):
    return board[:]

# Function to check if a space on the board is free
def isSpaceFree(board, move):
    return board[move] == ' '

# Function for the player to make a move
def getPlayerMove(board):
    move = ' '
    while move not in '1 2 3 4 5 6 7 8 9'.split() or not isSpaceFree(board, int(mov
        print('What is your next move? (1-9)')
        move = input()
    return int(move)

# Function for the computer to choose a random move from a list of possible moves
def chooseRandomMoveFromList(board, movesList):
    possibleMoves = []
    for i in movesList:
        if isSpaceFree(board, i):
            possibleMoves.append(i)

    if possibleMoves:
        return random.choice(possibleMoves)
    else:
        return None

# Function for the computer to make a move
def getComputerMove(board, computerLetter):
    if computerLetter == 'X':
        playerLetter = 'O'
    else:
        playerLetter = 'X'

    # Check if the computer can win in the next move
    for i in range(1, 10):
        copy = getBoardCopy(board)
        if isSpaceFree(copy, i):
            makeMove(copy, computerLetter, i)
            if isWinner(copy, computerLetter):
                return i

    # Check if the player could win on their next move, and block them
    for i in range(1, 10):
        copy = getBoardCopy(board)
        if isSpaceFree(copy, i):
            makeMove(copy, playerLetter, i)
            if isWinner(copy, playerLetter):
```

```
                return i

    # Try to take one of the corners if they are free
    move = chooseRandomMoveFromList(board, [1, 3, 7, 9])
    if move is not None:
        return move

    # Try to take the center if it is free
    if isSpaceFree(board, 5):
        return 5

    # Move on one of the sides
    return chooseRandomMoveFromList(board, [2, 4, 6, 8])

# Function to check if the board is full
def isBoardFull(board):
    return all(board[i] != ' ' for i in range(1, 10))



# Now placing X on the defined locations(before game starts).
def initializeBoard():
    # Creating 4 places for the board to be filled before the game starts.
    startBoxes = list(eval(input("Enter positions to be filled (comma-seperated-val
    board = [' '] * 10
    for box in startBoxes:
        board[box] = 'X'
    return board

# Function to reset the board and start a new game
def resetBoard():
    global theBoard
    theBoard = initializeBoard()

# Main game loop
print("Welcome to Tic Tac Toe")

while True:
    resetBoard()
    playerLetter, computerLetter = inputPlayerLetter()
    turn = whoGoesFirst()
    print('The ' + turn + ' will go first.')
    gameIsPlaying = True

    while gameIsPlaying:
        if turn == 'player':
            drawBoard(theBoard)
            move = getPlayerMove(theBoard)
            makeMove(theBoard, playerLetter, move)

            if isWinner(theBoard, playerLetter):
                drawBoard(theBoard)
                print("Hooray! You have won the game!")
                gameIsPlaying = False
            else:
                if isBoardFull(theBoard):
```

```
                    drawBoard(theBoard)
                    print('The game is a tie')
                    break
                else:
                    turn = 'computer'

        else:
            move = getComputerMove(theBoard, computerLetter)
            makeMove(theBoard, computerLetter, move)

            if isWinner(theBoard, computerLetter):
                drawBoard(theBoard)
                print('The computer has beaten you: you lose.')
                gameIsPlaying = False
            else:
                if isBoardFull(theBoard):
                    drawBoard(theBoard)
                    print('The game is a tie')
                else:
                    turn = 'player'

    if not playAgain():
        break
```

```
Welcome to Tic Tac Toe
```

```
Do you want to be X or O?
Do you want to be X or O?
Do you want to be X or O?
The computer will go first.
   |   |
 X |   | X
   |   |
-----------
   |   |
   |   |
   |   |
-----------
   |   |
 X | O | X
   |   |
What is your next move? (1-9)
   |   |
 X |   | X
   |   |
-----------
   |   |
 X |   |
   |   |
-----------
   |   |
 X | O | X
   |   |
Hooray! You have won the game!
Do you want to play again? (yes or no)
Do you want to be X or O?
The computer will go first.
   |   |
   |   |
   |   |
-----------
   |   |
 X |   | X
   |   |
-----------
   |   |
 X | X | X
   |   |
The computer has beaten you: you lose.
Do you want to play again? (yes or no)
```