EXPERIMENT NO : 27

AIM : Build a bot to play Tic Tac Toe game in python

PROGRAM

```
board = [' ']
* 10
computer =
'X' human =
'O'


def display_board(board):
        print(f'{board[1]} | {board[2]} |{board[3]}')
        print(f'{board[4]} | {board[5]} |{board[6]}')
        print(f'{board[7]} | {board[8]} |
        {board[9]}') print('-' * 10)

def check_win():
        if board[1] == board[2] == board[3] and board[1] !
                = ' ': return True
        elif board[4] == board[5] == board[6] and
                board[4] != ' ': return True
        elif board[7] == board[8] == board[9] and
                board[7] != ' ': return True
        elif board[1] == board[4] == board[7] and
                board[1] != ' ': return True
        elif board[2] == board[5] == board[8] and
                board[2] != ' ': return True
        elif board[3] == board[6] == board[9] and
                board[3] != ' ': return True
        elif board[1] == board[5] == board[9] and
                board[1] != ' ': return True
        elif board[7] == board[5] == board[3] and
                board[7] != ' ': return True
        else:
                return False

def is_win(letter):
        If board[1] == board[2] == board[3] and board[1]
                == letter: return True
        elif board[4] == board[5] == board[6] and board[4] ==
                letter: return True
        elif board[7] == board[8] == board[9] and board[7] ==
                letter: return True
        elif board[1] == board[4] == board[7] and board[1] ==
                letter: return True
```

```
        elif board[2] == board[5] == board[8] and board[2]
                ==letter: return True
        elif board[3] == board[6] == board[9] and board[3] ==
                letter: return True
        elif board[1] == board[5] == board[9] and board[1] ==
                letter: return True
        elif board[7] == board[5] == board[3] and board[7] ==
                letter: return True
        else:
                return False

def check_draw():
        if board.count(' ') < 2:
                return True
        else:
                return False

def is_available(pos):
        return True if board[pos] == ' ' else False

def insert(letter, pos):
        if is_available(pos):
                board[pos] =
                letter if
                check_win():
                        if letter == 'X':
                                display_board(board) # Display the final board before
                                exiting print("Computer Wins")
                                exit()
                        else:
                                display_board(board) # Display the final board before
                                exiting print("Human wins")
                                exit()
                if check_draw():
                        display_board(board) # Display the final board before exiting
                        print("Draw")
                        exit()
        else
        :       pos = int(input("Not Free! Please re-enter a position"))
                insert(letter, pos)


def human_move(letter):
        pos = int(input("Enter the position to
        insert:")) insert(letter, pos)

def
        computer_move(le
        tter): best_score
        = -100
```

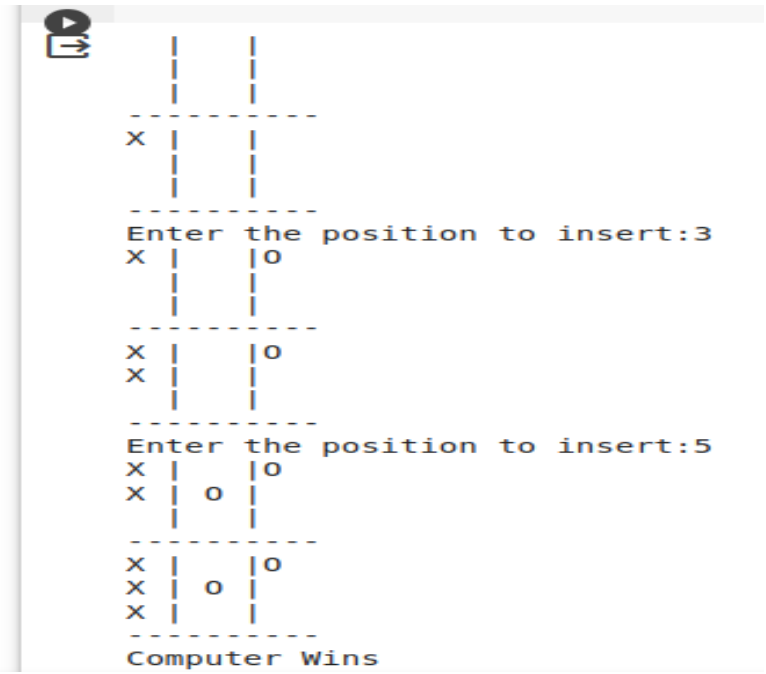
```
        best_pos = 0
        for index in range(1,
                len(board)): if
                is_available(index):
                        board[index] = letter
                        score = minimax(board,
                        False) board[index] = " "
                        If score > best_score:
                                best_score =
                                score best_pos
                                = index
        insert(letter,
        best_pos) return

def minimax(board,
        is_maximizing): if
        is_win(computer):
                return 10
        elif
        is_win(human):
                return -
        10 elif
        check_draw():
                return 0
        if is_maximizing: # computer turn x
                best_score = -100
                for index in range(1,
                        len(board)): if
                        is_available(index):
                                board[index] = computer
                                score = minimax(board,
                                False) board[index] = " "
                                best_score = max(best_score,score)
                return best_score
        else: # human turn o
                best_score = 100
                for index in range(1,
                        len(board)): if
                        is_available(index):
                                board[index] = human
                                score = minimax(board,
                                True) board[index] = " "
                                best_score = min(best_score,score)
                return best_score


# Initial board setup
display_board(board)

while not check_win() and not check_draw():
```

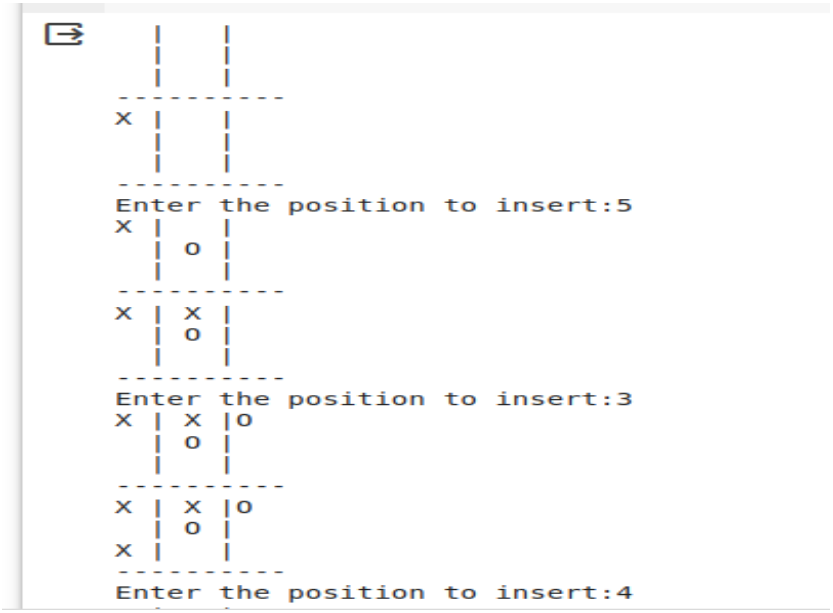
```
computer_move(computer)
if check_win() or
        check_draw(): break
display_board(board) # Display the board after computer's move
```

```
human_move(human)
display_board(board) # Display the board after human's move
```

RESULT : Successfully builded bot to play Tic Tac Toe game

OUTPUT :





```
X | X |O
O | O |
X |   |
----------
X | X |O
O | O |X
X |   |
----------
Enter the position to insert:8
X | X |O
O | O |X
X | O |
----------
X | X |O
O | O |X
X | O |X
----------
Draw
```