

## **PROGRAM TITLE 12**

### **TIC TOC TOE GAME**

#### **AIM:**

To write a python program for Tic Toc Toe game

#### **PROCEDURE:**

##### **1. Initialize the Board:**

- Create a 3x3 grid to represent the game board.
- Initialize all cells to be empty.

##### **2. Display the Board:**

- Display the current state of the board to the players.

##### **3. Player Moves:**

- Allow Player 1 ('X') to make a move by selecting an empty cell on the board.
- Check if the selected move is valid (the cell is empty).
- If valid, mark the cell with 'X', and move to the next step.
- If invalid, ask the player to make another move.

##### **4. Check for a Win:**

- After each move, check if the current player has won by having three symbols in a row (horizontally, vertically, or diagonally).
- If a win is detected, end the game and declare the winner.

##### **5. Check for a Draw:**

- After each move, check if the board is full (no empty cells).
- If the board is full and no player has won, declare the game as a draw.

##### **6. Switch Players:**

- Switch to the next player (Player 2, 'O') and repeat steps 3-5.

## 7. Repeat Until Game Over:

- Continue alternating between players until a player wins or the game ends in a draw.

## 8. End of Game:

- Display the final state of the board.
- Declare the winner or announce a draw.

## CODING:

```
def print_board(board):
    for row in board:
        print(" | ".join(row))
    print("-" * 5)

def check_winner(board):
    for row in board:
        if row[0] == row[1] == row[2] != ' ':
            return row[0]

    for col in range(3):
        if board[0][col] == board[1][col] == board[2][col] != ' ':
            return board[0][col]

    if board[0][0] == board[1][1] == board[2][2] != ' ':
        return board[0][0]
    if board[0][2] == board[1][1] == board[2][0] != ' ':
        return board[0][2]

    return None
```

```
def is_board_full(board):
```

```
    for row in board:
```

```
        for cell in row:            if
```

```
            cell == ' ':
```

```
                return False    return
```

```
    True
```

```
def play_game():
```

```
    board = [[' ']*3 for _ in range(3)]
```

```
    current_player = 'X'
```

```
    while True:
```

```
        print_board(board)        row = int(input(f'Player {current_player}, enter row
number (0, 1, or 2): '))        col = int(input(f'Player {current_player}, enter column
number (0, 1, or 2): '))
```

```
        if board[row][col] != ' ':
```

```
            print("That cell is already occupied. Try again.")
```

```
        continue
```

```
        board[row][col] = current_player
```

```
        winner = check_winner(board)
```

```
        if winner:
```

```
            print_board(board)
```

```
        print(f'Player {winner} wins!')
```

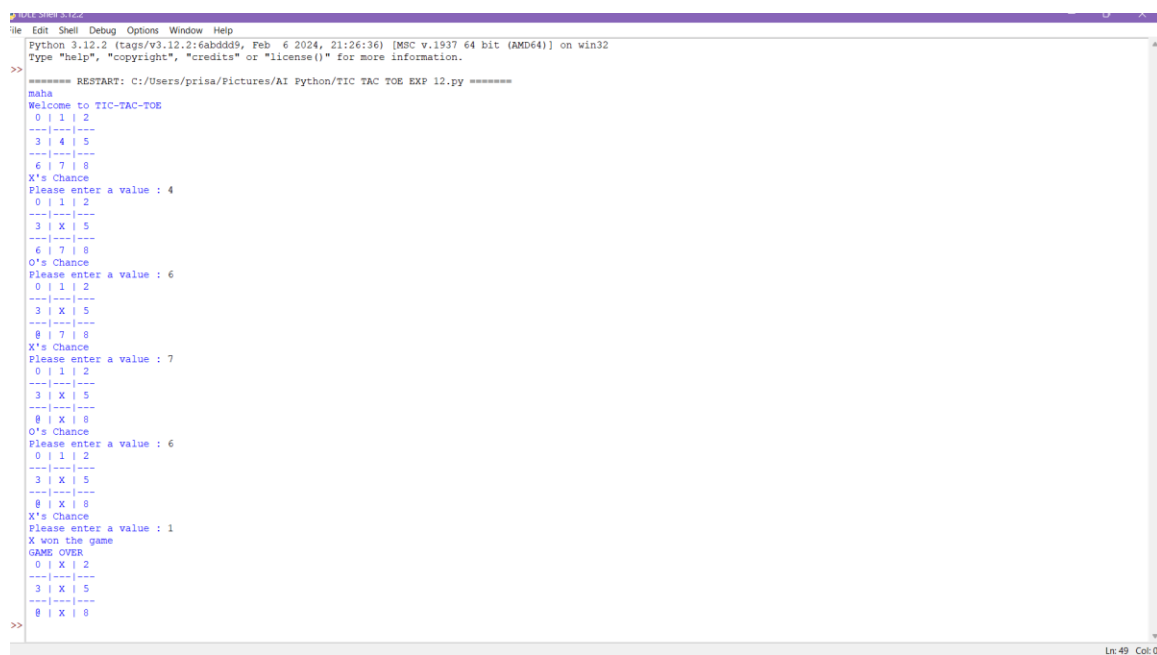
```
        break        elif is_board_full(board):
```

```
print_board(board)      print("It's a
tie!")      break
```

```
current_player = 'O' if current_player == 'X' else 'X'
```

```
if __name__ == "__main__":
    play_game()
```

## OUTPUT:



```
Python 3.12.2 (tags/v3.12.2:6abdd99, Feb 6 2024, 21:26:36) [MSC v.1937 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: C:/Users/prisa/Pictures/AI Python/TIC TAC TOE EXP 12.py =====
baha
Welcome to TIC-TAC-TOE
 0 | 1 | 2
---|---|---
 3 | 4 | 5
---|---|---
 6 | 7 | 8
X's Chance
Please enter a value : 4
 0 | 1 | 2
---|---|---
 3 | X | 5
---|---|---
 6 | 7 | 8
O's Chance
Please enter a value : 6
 0 | 1 | 2
---|---|---
 3 | X | 5
---|---|---
 6 | 7 | 8
X's Chance
Please enter a value : 7
 0 | 1 | 2
---|---|---
 3 | X | 5
---|---|---
 6 | X | 8
O's Chance
Please enter a value : 6
 0 | 1 | 2
---|---|---
 3 | X | 5
---|---|---
 6 | X | 8
X's Chance
Please enter a value : 1
X won the game
GAME OVER
 0 | X | 2
---|---|---
 3 | X | 5
---|---|---
 6 | X | 8
>>>
```

## RESULT:

Hence the program been successfully executed and verified.