PROGRAM TITLE 12

TIC TOC TOE GAME

AIM:

To write a python program for Tic Toc Toe game

PROCEDURE:

- **Initialize the Board:** Initialize the Tic Tac Toe board with empty cells.
- **Print the Board:** Write a function to print the current state of the board.
- **Check for Winner:** Write a function to check if there is a winner by examining rows, columns, and diagonals.
- **Check for Tie:** Write a function to check if the board is full, indicating a tie.
- **Game Loop**: Write a loop to handle player turns and moves. In each iteration, prompt the current player for their move, check for a winner or tie, and switch to the next player if the game continues. If there is a winner or tie, end the game and display the result.

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:

```
-----
Player 0, enter row number (0, 1, or 2): 2
Player 0, enter column number (0, 1, or 2): 2
  | X | 0
-----
  | 0 | X
-----
  | X | 0
-----
Player X, enter row number (0, 1, or 2): 0
Player X, enter column number (0, 1, or 2): 0
X | X | 0
-----
  | 0 | X
-----
  | X | 0
-----
Player 0, enter row number (0, 1, or 2): 2
Player 0, enter column number (0, 1, or 2): 0
X | X | 0
-----
  | 0 | X
-----
0 | X | 0
-----
Player 0 wins!
```

RESULT:

Hence the program been successfully executed and verified.