12. Write the python program for Tic Tac Toe game

**AIM :** Tic Tac Toe game

**ALGORITHM :**

1. The ‘print\_board’ function prints the current state of the Tic Tac Toe board.
2. The’check\_winner’ function checks if the current player has won the game.
3. The ‘is\_board\_full’ function checks if the board is completely filled, resulting in a draw.
4. The ‘play\_tic\_tac\_toe’ function contains the main game loop and player input handling.
5. The game starts with an empty 3x3 board and alternates between players "X" and "O".
6. Players input their moves by specifying row and column numbers.
7. After each move, the program checks for a winner or a draw and ends the game if necessary.

**PROGRAM :**

def print\_board(board):

for row in board:

print(" | ".join(row))

print("-" \* 9)

def check\_winner(board, player):

for i in range(3):

if all(cell == player for cell in board[i]):

return True

if all(board[j][i] == player for j in range(3)):

return True

if all(board[i][i] == player for i in range(3)) or all(board[i][2 - i] == player for i in range(3)):

return True

return False

def is\_board\_full(board):

return all(cell != " " for row in board for cell in row)

def play\_tic\_tac\_toe():

board = [[" " for \_ in range(3)] for \_ in range(3)]

current\_player = "X"

while True:

print\_board(board)

print(f"Player {current\_player}'s turn")

while True:

row = int(input("Enter row (0, 1, 2): "))

col = int(input("Enter column (0, 1, 2): "))

if 0 <= row < 3 and 0 <= col < 3 and board[row][col] == " ":

break

else:

print("Invalid move. Try again.")

board[row][col] = current\_player

if check\_winner(board, current\_player):

print\_board(board)

print(f"Player {current\_player} wins!")

break

elif is\_board\_full(board):

print\_board(board)

print("It's a draw!")

break

current\_player = "O" if current\_player == "X" else "X"

play\_tic\_tac\_toe()

**OUT PUT :**

