LAB 1

Implementation Of Tic-Tac-Toe

Code

```
def print_board(board):
    print("\nCurrent Board:")
            for row in board:
        print(row)
    print()



def check_winner(board, player):
            # Check rows
            for row in board:
        if all(cell == player for cell in row):
            return True

            # Check columns
            for col in range(3):
        if all(board[row][col] == player for row in range(3)):
            return True

            # Check diagonals
```

```
        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 tic_tac_toe():
    board = [[' ' for _ in range(3)] for _ in range(3)]
    players = ['X', 'O']
    move_count = 0

    while True:
        current_player = players[move_count % 2]
        print(f"Player {current_player}, enter row (0-2): ", end="")
        row = int(input())
        print(f"Player {current_player}, enter col (0-2): ", end="")
        col = int(input())

        # If cell is empty
        if board[row][col] == ' ':
            board[row][col] = current_player
            move_count += 1
            print_board(board)
```

```
            if check_winner(board, current_player):
                print(f"Player {current_player} wins!")
                    print(f"Total moves (cost): {move_count}")
                break
            if move_count == 9:  # Board full
                print("It's a draw!")
                break
        else:
            print("Cell already taken! Try again.")



# Run the game
tic_tac_toe()
```

Output:

```
= RESTART: C:/Users/student/AppData/Local/Programs/Python/Python313/302/lab1.py
Santhosh N (USN : 1BM23CS302)
['-', '-', '-']
['-', '-', '-']
['-', '-', '-']
Enter position to place X:
0
0
['X', '-', '-']
['-', '-', '-']
['-', '-', '-']
Enter position to place O:
0
1
['X', 'O', '-']
['-', '-', '-']
['-', '-', '-']
Enter position to place X:
2
2
['X', 'O', '-']
['-', '-', '-']
['-', '-', 'X']
Enter position to place O:
2
0
['X', 'O', '-']
['-', '-', '-']
['O', '-', 'X']
Enter position to place X:
1
1
['X', 'O', '-']
['-', 'X', '-']
['O', '-', 'X']
X wins!
Game Over
Total moves made (cost): 5
```