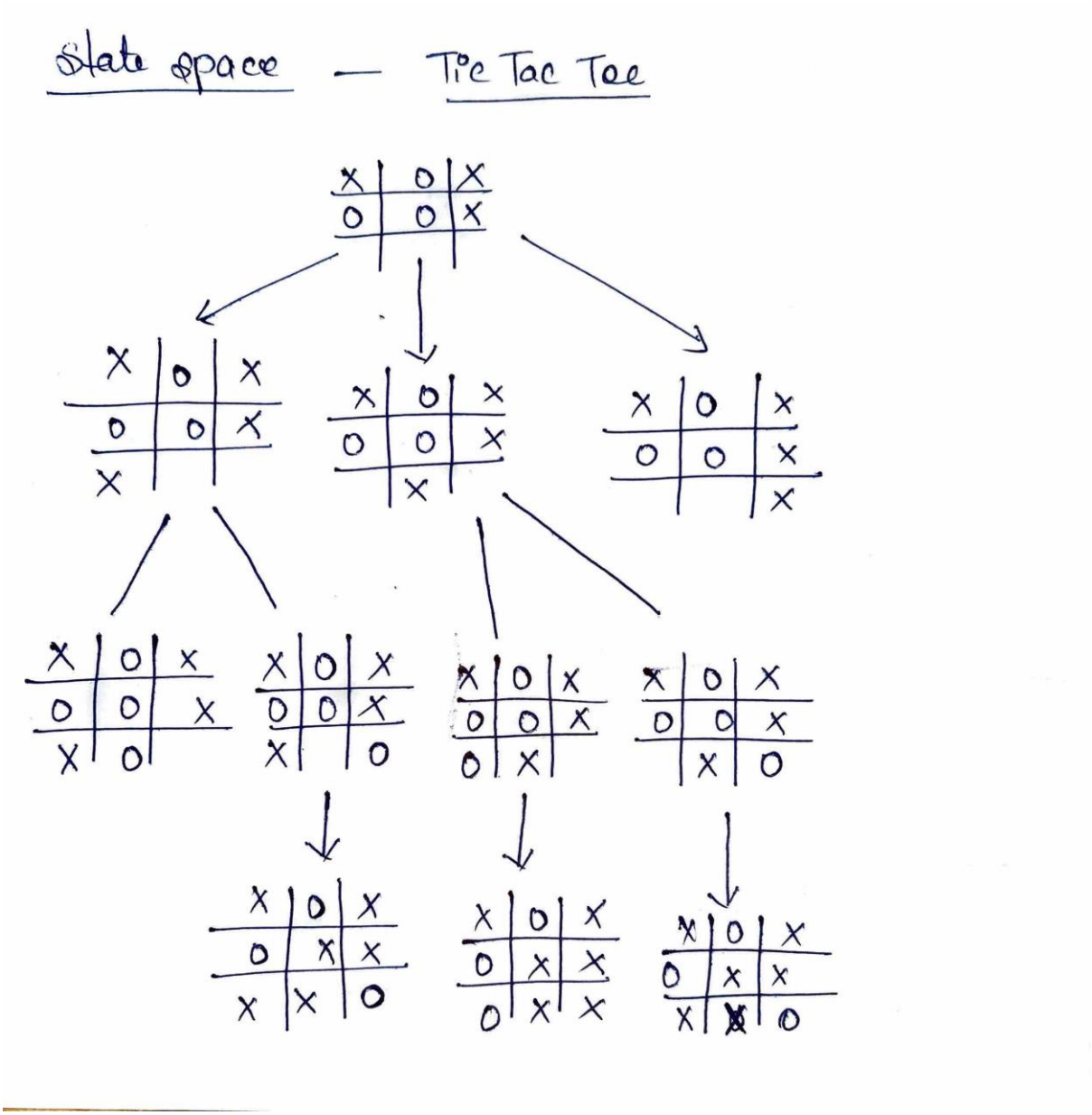
LAB PROGRAM 1

Implement Tic Tac Toe game.



Lab - 2 Tictac Toe 1/10/24

Algorithm:

1. Initialize game board – Create a 3×3 board filled with empty spaces.
2. Print the board positions and the corresponding index numbers (0-8) for user input.
3. Set human player as "x" and the computer as "0".
4. Game loop:

   Repeat until the game ends (win or draw)

   Print current board state.

   // check current player

   If the current player is "x" (human), prompt the user to input their move.

   For validating I/P, ensure that user has entered b/w 0 and 8 and also check that the selected cell is not already occupied, if invalid prompt again.

   if the current player is "0" (computer) generate a list of available moves and implement the strategy to choose the best move, win move & block move at random.
5. Place the current player's mark either 0 or x in the selected cell.
6. Call check_winner(board) to check for the winner.
7. Call is_board_full(board) to check for the draw and then exit the loop.
8. Then player can switch from x to 0.

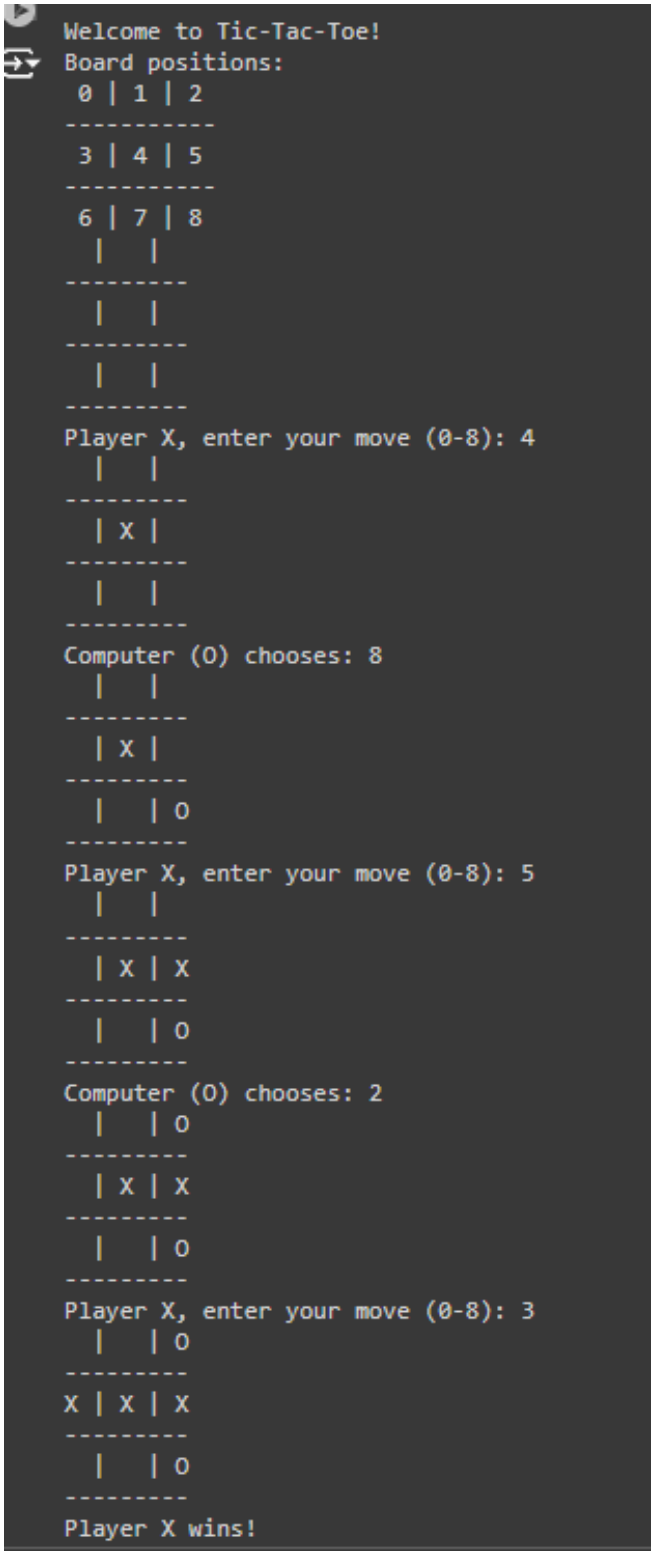
Code:

```
import random

def print_board(board):
    for row in board:
        print(" | ".join(row))
        print("-" * 9)

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

def get_available_moves(board):
    return [(i, j) for i in range(3) for j in range(3) if board[i][j] == " "]
```

```
def tic_tac_toe():
    board = [[" " for _ in range(3)] for _ in range(3)]
    current_player = "X"  # Human player
    computer_player = "O"

    print("Welcome to Tic-Tac-Toe!")
    print("Board positions:")
    print(" 0 | 1 | 2")
    print("-----------")
    print(" 3 | 4 | 5")
    print("-----------")
    print(" 6 | 7 | 8")

    while True:
        print_board(board)

        if current_player == "X":
            move = input(f"Player {current_player}, enter your move (0-8): ")
            try:
                move = int(move)
                row, col = divmod(move, 3)
                if board[row][col] != " ":
                    print("Invalid move. Cell already occupied. Try again.")
                    continue
            except (ValueError, IndexError):
                print("Invalid input. Please enter a number between 0 and 8.")
                continue
        else:
            # Computer's turn
            available_moves = get_available_moves(board)
```

```
            row, col = random.choice(available_moves)
            print(f"Computer ({computer_player}) chooses: {row * 3 + col}")

        board[row][col] = current_player
        winner = check_winner(board)

        if winner:
            print_board(board)
            print(f"Player {winner} wins!")
            break

        if is_board_full(board):
            print_board(board)
            print("It's a draw!")
            break

        # Switch players
        current_player = computer_player if current_player == "X" else "X"

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

Output:


Welcome to Tic-Tac-Toe!
Board positions:
0 | 1 | 2
3 | 4 | 5
6 | 7 | 8
Player X, enter your move (0-8): 4
Computer (O) chooses: 8
Player X, enter your move (0-8): 5
Computer (O) chooses: 2
Player X, enter your move (0-8): 3
X | X | X
Player X wins!