ASSIGNMENT 6

Tic Tac Toe Game

```
#include <stdio.h>

// Function prototypes
void displayBoard(char board[3][3]);
int checkWin(char board[3][3]);
void makeMove(char board[3][3], int player);

int main() {
   char board[3][3] = {
      {'1', '2', '3'},
      {'4', '5', '6'},
      {'7', '8', '9'}
   };

   int currentPlayer = 1; // Player 1 starts
   int gameStatus = 0;    // 0 = ongoing, 1 = win, -1 = draw

   printf("Welcome to Tic Tac Toe!\n");
   printf("Player 1 (X)  -  Player 2 (O)\n");

   while (gameStatus == 0) {
      displayBoard(board);
      makeMove(board, currentPlayer);
      gameStatus = checkWin(board);
      currentPlayer = (currentPlayer == 1) ? 2 : 1; // Switch
players
   }
```

```
    displayBoard(board);

    if (gameStatus == 1) {
        printf("Congratulations! Player %d wins!\n",
(currentPlayer == 1) ? 2 : 1);
    } else {
        printf("It's a draw!\n");
    }

    return 0;
}

// Function to display the board
void displayBoard(char board[3][3]) {
    printf("\n");
    for (int i = 0; i < 3; i++) {
        for (int j = 0; j < 3; j++) {
            printf(" %c ", board[i][j]);
            if (j < 2) printf("|");
        }
        printf("\n");
        if (i < 2) printf("---|---|---\n");
    }
    printf("\n");
}

// Function to check if there is a win or a draw
int checkWin(char board[3][3]) {
    // Check rows and columns
    for (int i = 0; i < 3; i++) {
```

```
      if (board[i][0] == board[i][1] && board[i][1] ==
board[i][2]) return 1;
      if (board[0][i] == board[1][i] && board[1][i] ==
board[2][i]) return 1;
   }

   // Check diagonals
   if (board[0][0] == board[1][1] && board[1][1] ==
board[2][2]) return 1;
   if (board[0][2] == board[1][1] && board[1][1] ==
board[2][0]) return 1;

   // Check for a draw
   int draw = 1;
   for (int i = 0; i < 3; i++) {
      for (int j = 0; j < 3; j++) {
         if (board[i][j] != 'X' && board[i][j] != 'O') {
            draw = 0; // Empty space found
         }
      }
   }
   if (draw) return -1;

   return 0; // Game is still ongoing
}

// Function to handle player moves
void makeMove(char board[3][3], int player) {
   char mark = (player == 1) ? 'X' : 'O';
   int choice;
   int row, col;
```

```
    while (1) {
        printf("Player %d, enter your move (1-9): ", player);
        scanf("%d", &choice);

        if (choice < 1 || choice > 9) {
            printf("Invalid input. Please choose a number
between 1 and 9.\n");
            continue;
        }

        // Map the choice to row and column
        row = (choice - 1) / 3;
        col = (choice - 1) % 3;

        // Check if the chosen cell is free
        if (board[row][col] != 'X' && board[row][col] != 'O') {
            board[row][col] = mark;
            break;
        } else {
            printf("Cell already occupied. Try again.\n");
        }
    }
}
```