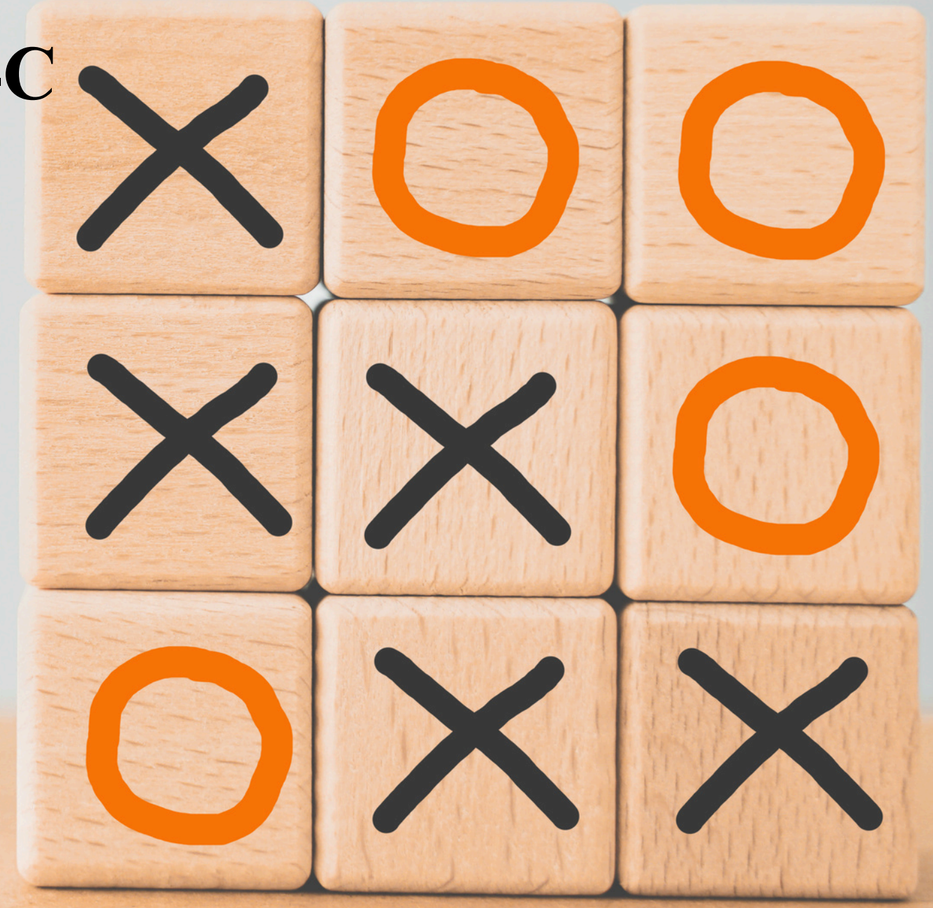
PROJECT TITLE : TIC-TAC-TOE GAME

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CSE-C



TEAM MEMBERS

P.SANTHIYA SHREE

N.SUBIKSHA

AIM :

The aim of this project is to develop a simple yet engaging TIC-TAC-TOE game that allows tow players compete against each other .where players take turns marking X or O on 3×3 board.

SOURCE CODE

```c
// carried
#include <stdio.h>

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

void displayBoard() {
    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("|");
        }
        if (i < 2) printf("\n---+---+---\n");
    }
    printf("\n");
}

int checkWin() {
    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;
    }
    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;
    return 0;
}

void makeMove() {
    int choice;
    char mark = (currentPlayer == 1) ? 'X' : 'O';

    printf("Player %d (%c), enter your move (1-9):
        ", currentPlayer, mark);
    scanf("%d", &choice);

    int row = (choice - 1) / 3;
    int col = (choice - 1) % 3;

    if (choice < 1 || choice > 9 || board[row][col]
        == 'X' || board[row][col] == 'O') {
        printf("Invalid move. Try again.\n");
        makeMove();
    } else {
        board[row][col] = mark;
    }
}

int main() {
    int moves = 0;

    printf("Welcome to Tic-Tac-Toe!\n");
    displayBoard();

    while (moves < 9) {
        makeMove();
        displayBoard();
        if (checkWin()) {
            printf("Player %d wins!
                Congratulations!\n", currentPlayer
                );
            return 0;
        }
        currentPlayer = (currentPlayer == 1) ? 2 :
            1;
        moves++;
    }

    printf("It's a draw!\n");
    return 0;
}
```

PROGRAM EXPLANATION

1. **Board & Players:**

   board[3][3] holds the game state.

   currentPlayer alternates between 1 (X) and 2 (O).

2. **Functions:**

   displayBoard(): Prints the board.

   checkWin(): Checks rows, columns, and diagonals for a win.

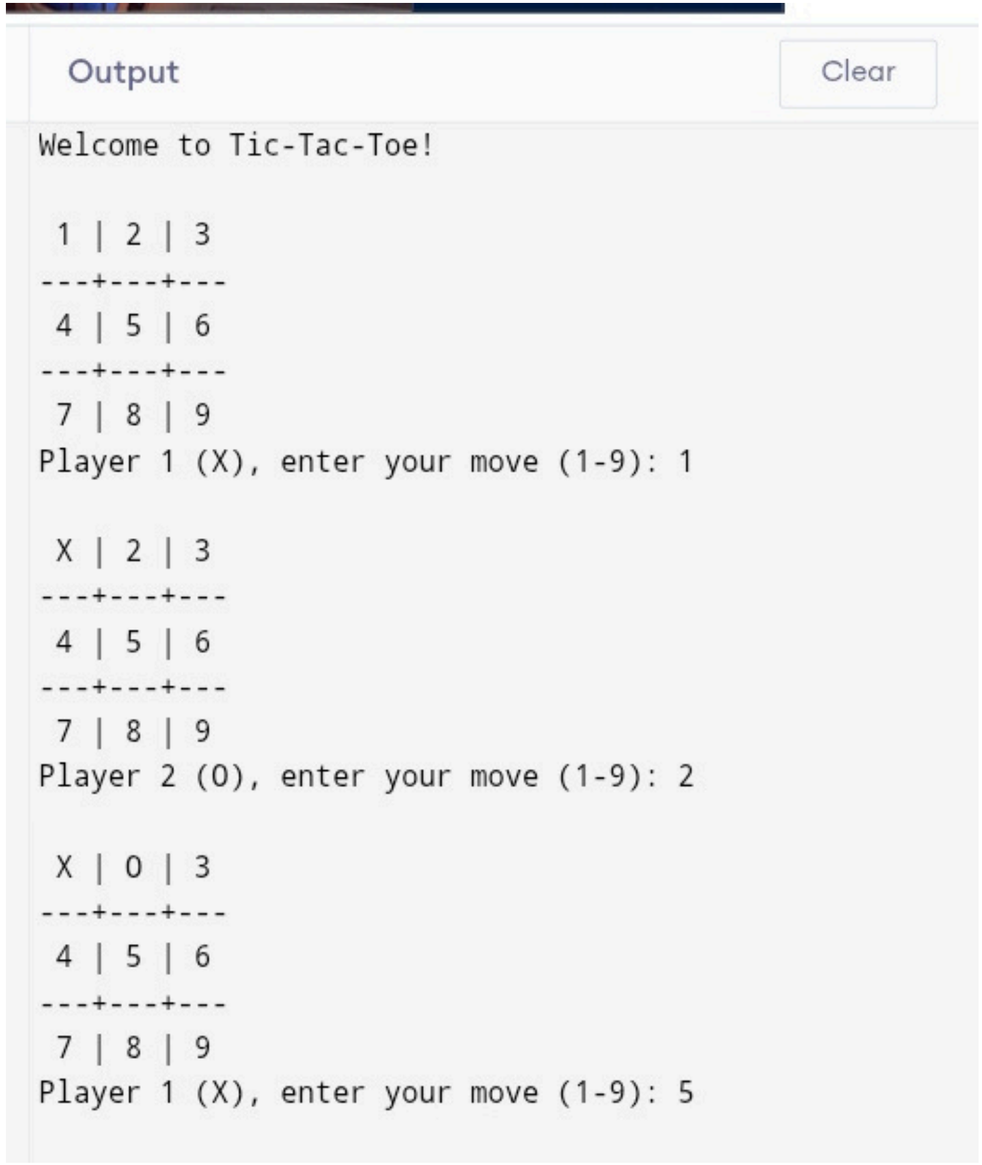
   makeMove(): Handles player input, validates, and updates the board.

3. **Game Loop:**

   Alternates players, displays the board, and checks for a win after each move.

   Ends with a win or a draw after 9 moves.

SOURCE OUTPUT



Output

Clear

```
Welcome to Tic-Tac-Toe!

 1 | 2 | 3
---+---+---
 4 | 5 | 6
---+---+---
 7 | 8 | 9
Player 1 (X), enter your move (1-9): 1

 X | 2 | 3
---+---+---
 4 | 5 | 6
---+---+---
 7 | 8 | 9
Player 2 (O), enter your move (1-9): 2

 X | O | 3
---+---+---
 4 | 5 | 6
---+---+---
 7 | 8 | 9
Player 1 (X), enter your move (1-9): 5

 X | O | 3
---+---+---
 4 | X | 6
---+---+---
 7 | 8 | 9
Player 2 (O), enter your move (1-9): 3

 X | O | O
---+---+---
 4 | X | 6
---+---+---
 7 | 8 | 9
Player 1 (X), enter your move (1-9): 9

 X | O | O
---+---+---
 4 | X | 6
---+---+---
 7 | 8 | X
Player 1 wins! Congratulations!


=== Code Execution Successful ===
```

OUTPUT EXPLANATION :

This is the output of a Tic-Tac-Toe game where:

Player 1 (X) and Player 2 (O) take turns selecting positions (1–9) to place their marks.

The board updates after every move.

Player 1 wins by forming a diagonal line with X in positions 1, 5, and 9.

The game concludes with a congratulatory message for the winner.

WHAT WE HAVE LEARNED

This mini-project teaches:

2D arrays for the game board,

Loops and conditions for logic flow,

Modular coding with reusable functions,

Input validation and player interaction,

Win checks and edge case handling,
Building strong basics in programming and logic.

THANK YOU