# TIC TAC TOE GAME – PROJECT REPORT


## Name – Manish Mohit Budhwar

## SAP ID - 590024677


---

## 1. Title of the Project

**Tic Tac Toe Game in C (Human vs Human & Human vs Computer)**

---

## 2. Aim of the Project

To design and implement a **console-based Tic Tac Toe game in C**, supporting both **Human vs Human** and **Human vs Computer** game modes, while demonstrating structured programming concepts.

---

## 3. Objectives

1. Implement a 3×3 Tic Tac Toe board using **2D arrays**.
2. Demonstrate the use of **functions** for modularity.
3. Apply **conditional logic** and **loops** for turn handling and game flow.
4. Use **random number generation** (`rand()`, `srand()`) for computer gameplay.
5. Implement **win detection** and **draw detection** algorithms.

---

## 4. Problem Statement

Develop a **C program** that allows two modes of Tic Tac Toe gameplay:

### Mode 1: Human vs Human

- Two players take turns entering positions (1–9).
- The program validates moves and updates the board.

### Mode 2: Human vs Computer

- User plays as ‘X’, computer plays as ‘O’.
- Computer selects valid moves randomly.

The program must correctly check for:

- Win (row/column/diagonal match)
- Draw (board is full)
- Invalid move input

---

# 5. Technologies / Concepts Used

- **C Language**
- Header Files:
  - `stdio.h` – Input/Output
  - `stdlib.h` – Random functions (`rand`)
  - `time.h` – Seed generator (`time(0)`)
- **2D Arrays**
- **Functions and Calls**
- **Loops (`while`, `for`)**
- **Conditionals (`if-else`)**
- **Random number generation**

---

# 6. Project Overview

The program uses a **global 3×3 board** and several functions:

- `resetBoard()` — Initializes board with positions 1–9
- `displayBoard()` — Prints the board layout
- `humanMove()` — Handles user move input and validation
- `computerMove()` — Randomly selects an empty position
- `checkWin()` — Checks rows, columns, diagonals
- `checkDraw()` — Determines if board is full
- `playHumanVsHuman()` — Game loop for two players
- `playHumanVsComputer()` — Game loop vs computer
- `main()` — Displays the menu and starts the game

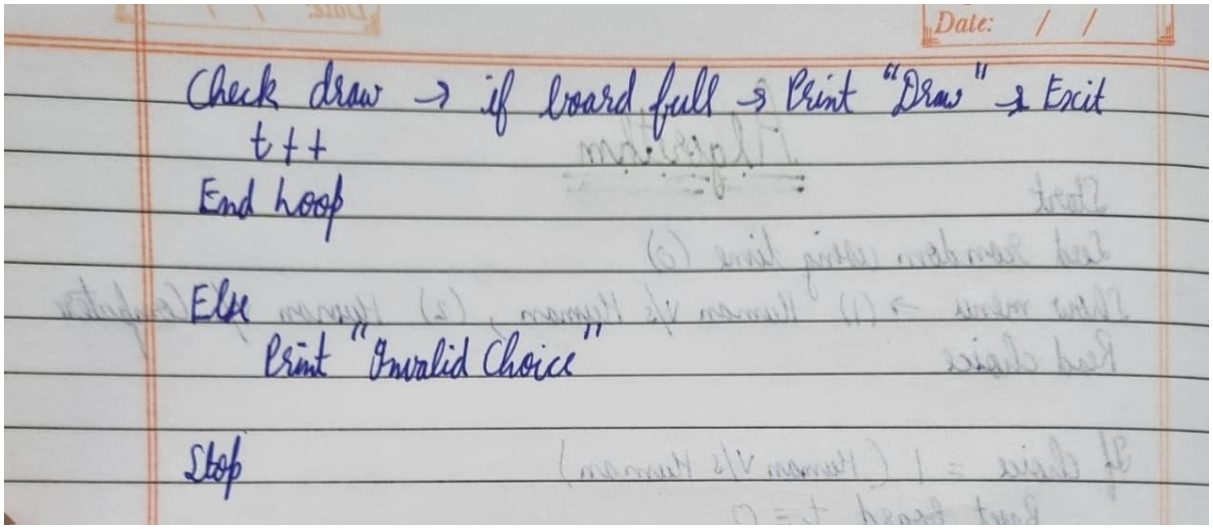
---

# 7. Algorithm / Pseudocode

# Algorithm

Start
Seed random using time (0)
Show menu → (1) Human v/s Human, (2) Human v/s Computer
Read choice

If choice = 1 (Human v/s Human)
  Reset board, t = 0
  Loop
    Display board
    If t even → player 'X' plays
    Else → player 'O' plays
    Ask position (1-9), convert to row/col
    If cell filled → ask again
    Place symbol
    Check win → If row/col/diagonal same → show board & declare winner → Exit
    Check draw → If all cells filled → Print "Draw" → Exit
    t++
  End Loop
Else if choice = 2 (Human v/s Computer)
  Reset board, t = 0
  Loop
    Display board
    If t even → human enters position for 'X'
      Convert, validate, place 'X'
    Else → computer generates random 1-9 → If empty place 'O'
    Check win → if 3 same symbols in row/col/diagonal
      if t even → Human wins else Computer wins → Exit




---

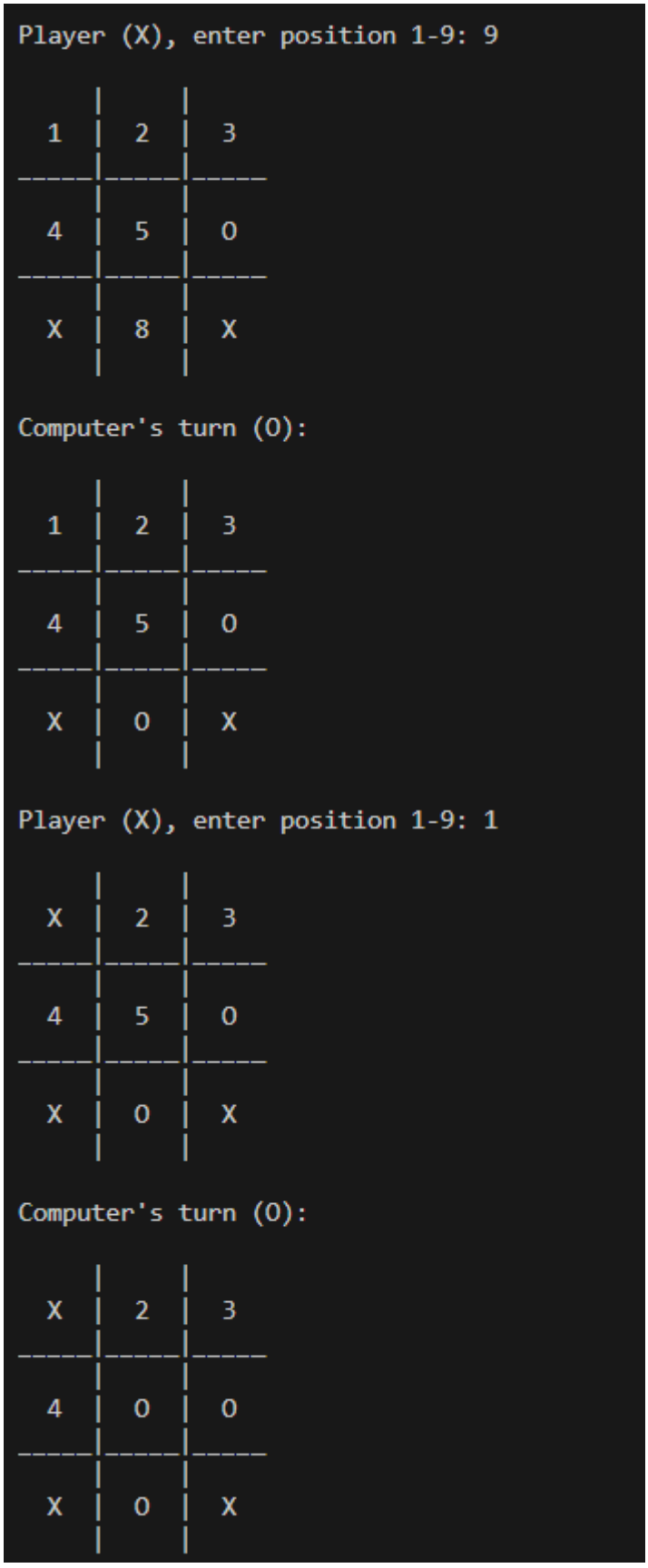
# 8. Code Explanation (Brief)

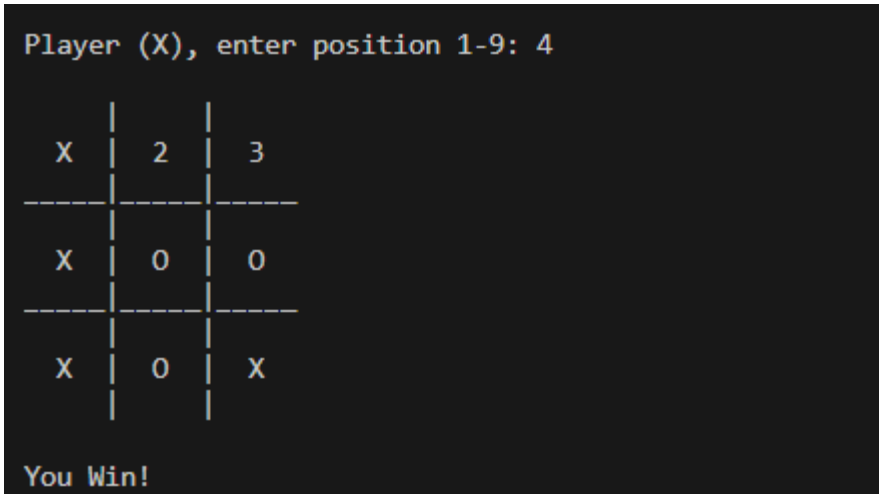
- The board is a **global 3×3 array**, accessible to all functions.
- The game runs in loops and alternates turns using `t % 2`.
- Winning patterns are checked using three rows, three columns, and two diagonals.
- Draw occurs when no cell contains numbers 1–9.
- Computer uses `rand()` to choose a valid empty spot.
- Recursive re-calling of `humanMove()` ensures valid input.

---

# 9.Output

```
manish@Manish:~/Workspace/project/output$ ./"pro"
TIC TAC TOE GAME
1. Human vs Human
2. Human vs Computer
Choose mode: 2

     |     |
  1  |  2  |  3
_____|_____|_____
     |     |
  4  |  5  |  6
_____|_____|_____
     |     |
  7  |  8  |  9
     |     |

Player (X), enter position 1-9: 7

     |     |
  1  |  2  |  3
_____|_____|_____
     |     |
  4  |  5  |  6
_____|_____|_____
     |     |
  X  |  8  |  9
     |     |

Computer's turn (O):

     |     |
  1  |  2  |  3
_____|_____|_____
     |     |
  4  |  5  |  O
_____|_____|_____
     |     |
  X  |  8  |  9
     |     |
```


Player (X), enter position 1-9: 9
1 | 2 | 3
4 | 5 | O
X | 8 | X
Computer's turn (O):
1 | 2 | 3
4 | 5 | O
X | O | X
Player (X), enter position 1-9: 1
X | 2 | 3
4 | 5 | O
X | O | X
Computer's turn (O):
X | 2 | 3
4 | O | O
X | O | X




---

# 10. Possible Enhancements

- Add smarter computer AI (minimax algorithm).
- Add colored output for X and O.
- Add option to replay the game.
- Replace recursion in `humanMove()` with loops.
- Save score history.

---

# 11. Conclusion

This project successfully implements a **functional and interactive Tic Tac Toe game** using C language.
It demonstrates core programming concepts such as:

- Modular programming
- Decision-making
- Loops
- 2D array manipulation
- Randomization
- Game logic handling

It provides a strong foundation for further game development and more advanced AI integration.