UPES

UNIVERSITY OF TOMORROW

“Programming in C Project Report”

Name – Pratyaksh Singh and Prashant Kumar
Sapid – 590023001 and 590027491
Batch - B45
Faculty Name – Mrs. Dolly Das

Abstract:-

This project implements a simple two■player Tic Tac Toe game in the C programming language. The game uses a 3×3 board represented by a 2D array and allows two human players to take turns. The program checks for valid moves, win conditions, and draws. It demonstrates core programming concepts such as arrays, functions, loops, and condition checking.

Problem Definition:-

The goal of the project is to design and implement a console■based Tic Tac Toe game for two players. The system must allow: (1) alternated turns, (2) input validation, (3) display of current board state, (4) detection of wins or draws, and (5) clean program termination with proper results.

System Design:-

The program design is divided into modules:

- init_board(): initializes the game board
- display_board(): prints the current board state
- player_turn(): allows each player to make a validated move- check_win(): checks all win conditions (rows, columns, diagonals)

Algorithm (Simplified):

1. Initialize board
2. Loop for max 9 turns
3. Display board
4. Take input from current player
5. Validate and update board
6. Check win conditions
7. Announce winner or draw

Implementation Details:-

Below is the complete C implementation used in the project:

```
#include <stdio.h>
#include <stdlib.h>
char board[3][3];

void init_board();
void display_board();
char check_win();
void player_turn(int player_num, char marker);
int main() {

    char winner = ' ';
int turn;

    printf("Welcome to Tic Tac Toe Game (Pass n play -Two players only).\n");
printf("Player 1 is X, Player 2 is O.\n\n");     init_board();

    for (turn = 1; turn <= 9; turn++)
{         display_board();

        if (turn % 2 != 0)
{            player_turn(1,
'X');        } else
{            player_turn(2,
'O');        }

        winner = check_win();
if (winner != ' ')
{            break;
}     }     display_board();

    if (winner == 'X') printf("Congratulations! Player 1 (X) wins!\n");
else if (winner == 'O') printf("Congratulations! Player 2 (O) wins!\n");
else printf("It's a draw!\n");

    return 0; }

void init_board() {     int i,
j;     for (i = 0; i < 3; i++)
for (j = 0; j < 3; j++)
board[i][j] = ' '; }

void   display_board()   {              printf("\n\n\tTic   Tac   Toe\n\n");
printf("Player  1  (X)  -  Player  2  (O)\n\n");       printf("       |       |
\n");       printf("   %c   |   %c   |   %c   \n", board[0][0], board[0][1],
board[0][2]);      printf("__|_|__\n");      printf("      |      |      \n");
printf("   %c   |   %c   |   %c   \n", board[1][0], board[1][1], board[1][2]);
printf("__|_|__\n");      printf("      |      |      \n");
    printf("  %c  |  %c  |  %c  \n", board[2][0], board[2][1], board[2][2]);
printf("     |     |     \n\n"); }

void player_turn(int player_num, char marker)
{     int choice, row, column;

    while (1) {         printf("Player %d (%c), enter a number (1-9): ",
player_num, marker);         scanf("%d", &choice);

        row = (choice - 1) / 3;
column = (choice - 1) % 3;

        if (choice >= 1 && choice <= 9 && board[row][column] == ' ')
{ board[row][column] = marker; break; }         else printf("Invalid
move. Try again.\n");     } }

char check_win() {     int i;     for (i = 0; i < 3; i++) {         if (board[i][0] ==
board[i][1] && board[i][1] == board[i][2] && board[i][0] != ' ')             return
board[i][0];         if (board[0][i] == board[1][i] && board[1][i] == board[2][i] &&
board[0][i] != ' ')             return board[0][i];
```

```
    }

    if (board[0][0] == board[1][1] && board[1][1] == board[2][2] && board[0][0] != ' ')
return board[0][0];

    if (board[0][2] == board[1][1] && board[1][1] == board[2][0] && board[0][2] != ' ')
return board[0][2];

    return ' '; }
```

Testing & Results:-

This program was tested with multiple valid and invalid moves. All win conditions were correctly detected, and draw scenarios were verified.

Conclusion & Future Work:-

The Tic Tac Toe game meets all requirements and functions correctly. Possible future improvements include adding AI, GUI, and network multiplayer.