

Presented by: **Maryam Siddiqui**

**www.linkedin.com/in/maryam-siddiqui-ghani**

Task: **Tic-Tac-Toe**

**Project Report: Tic-Tac-Toe Game**

**Introduction**

The "Tic-Tac-Toe Game" project is a console-based game developed in C++ as an internship task for SoftCode. The objective of this project is to create a classic two-player Tic-Tac-Toe game that provides players with an interactive and strategic gaming experience. The game allows players to take turns and compete against each other to achieve a winning combination on the game board.

**Project Scope**

The scope of this project involves implementing a console-based version of the Tic-Tac-Toe game. The players take turns entering their moves on a 3x3 game board, aiming to create a line of their own symbol ('X' or 'O') horizontally, vertically, or diagonally. The game includes features such as displaying the game board, validating moves, checking for wins or draws, and alternating between players.

**Features**

* Interactive console interface for players to input their moves.
* Display of the Tic-Tac-Toe game board with visual separators.
* Validation of player moves to prevent invalid selections.
* Detection of win conditions for both 'X' and 'O' players.
* Identification of a draw when all cells are filled without a win.
* Automatic switching between players after each move.

**Implementation**

The "Tic-Tac-Toe Game" is implemented using C++ programming language and organized into distinct functions for modularity and ease of understanding.

**Functions:**

**1. displayBoard(char board[3][3]):**

Displays the current state of the Tic-Tac-Toe game board on the console, showing the positions of 'X' and 'O' symbols.

**2. checkWin(char board[3][3], char player):**

Checks if the specified player ('X' or 'O') has won by examining rows, columns, and diagonals on the game board.

**3. checkDraw(char board[3][3]):**

Determines if the game has ended in a draw by verifying if all cells on the game board are filled.

**4. isValidMove(char board[3][3], int row, int col):**

Validates whether a player's move to a specific cell is valid,considering the boundaries and existing content of the cell.

**5. switchPlayers(char &currentPlayer):**

Alternates between 'X' and 'O' players by updating the value of the `currentPlayer` variable.

**6. main():**

* Manages the main gameplay loop.
* Initializes the game board and player turn.
* Accepts player input for row and column selections.
* Checks for valid moves and updates the game board.
* Checks for win conditions and draws.
* Alternates player turns until the game is won or drawn.
* Displays a thank-you message when the game ends.

**Conclusion**

The "Tic-Tac-Toe Game" project showcases the creation of an interactive and engaging console-based game using fundamental programming concepts. By incorporating functions for board display, move validation, win/draw checks, and player switching, the project demonstrates proficiency in designing and developing a classic game that offers entertainment and challenge to players.

**Acknowledgments**

This project was completed by Maryam Siddiqui as part of an internship task for SoftCode. Special thanks to mentors and guides for their support during the development process.

**Contact Information**

For any inquiries or feedback related to this project, please contact:

Maryam Siddiqui

Note: This project report template serves as a starting point and can be customized further to match your specific project's details and requirements.

Feel free to expand and tailor this project report template according to your project's specifics.

**Source code:**

#include <iostream>

using namespace std;

// Function prototypes

void displayBoard(char board[3][3]);

bool checkWin(char board[3][3], char player);

bool checkDraw(char board[3][3]);

bool isValidMove(char board[3][3], int row, int col);

void switchPlayers(char &currentPlayer);

int main() {

char board[3][3] = {{' ', ' ', ' '}, {' ', ' ', ' '}, {' ', ' ', ' '}};

char currentPlayer = 'X';

bool gameWon = false;

bool gameDraw = false;

cout << "Tic-Tac-Toe Game" << endl;

do {

int row, col;

displayBoard(board);

// Player Input

cout << "Player " << currentPlayer << ", enter your move (row and column): ";

cin >> row >> col;

// Adjust row and col to match the array index (starting from 0)

row--;

col--;

if (isValidMove(board, row, col)) {

// Update Board

board[row][col] = currentPlayer;

// Check for Win

if (checkWin(board, currentPlayer)) {

displayBoard(board);

cout << "Player " << currentPlayer << " wins!" << endl;

gameWon = true;

}

// Check for Draw

if (checkDraw(board)) {

displayBoard(board);

cout << "It's a draw!" << endl;

gameDraw = true;

}

// Switch Players

switchPlayers(currentPlayer);

} else {

cout << "Invalid move. Please try again." << endl;

}

} while (!gameWon && !gameDraw);

cout << "Thank you for playing!" << endl;

return 0;

}

// Function to display the Tic-Tac-Toe board

void displayBoard(char board[3][3]) {

cout << endl;

for (int i = 0; i < 3; i++) {

for (int j = 0; j < 3; j++) {

cout << " " << board[i][j] << " ";

if (j < 2) cout << "|";

}

cout << endl;

if (i < 2) cout << "---+---+---" << endl;

}

cout << endl;

}

// Function to check if a player has won

bool checkWin(char board[3][3], char player) {

// Check rows, columns, and diagonals

for (int i = 0; i < 3; i++) {

if (board[i][0] == player && board[i][1] == player && board[i][2] == player) return true; // Check rows

if (board[0][i] == player && board[1][i] == player && board[2][i] == player) return true; // Check columns

}

if (board[0][0] == player && board[1][1] == player && board[2][2] == player) return true; // Check main diagonal

if (board[0][2] == player && board[1][1] == player && board[2][0] == player) return true; // Check secondary diagonal

return false;

}

// Function to check if the game is a draw

bool checkDraw(char board[3][3]) {

// Check if there are no empty cells

for (int i = 0; i < 3; i++) {

for (int j = 0; j < 3; j++) {

if (board[i][j] == ' ') return false;

}

}

return true;

}

// Function to check if a move is valid

bool isValidMove(char board[3][3], int row, int col) {

// Check if the cell is empty and within the valid range

return (row >= 0 && row < 3 && col >= 0 && col < 3 && board[row][col] == ' ');

}

// Function to switch players

void switchPlayers(char &currentPlayer) {

if (currentPlayer == 'X') {

currentPlayer = 'O';

} else {

currentPlayer = 'X';

}

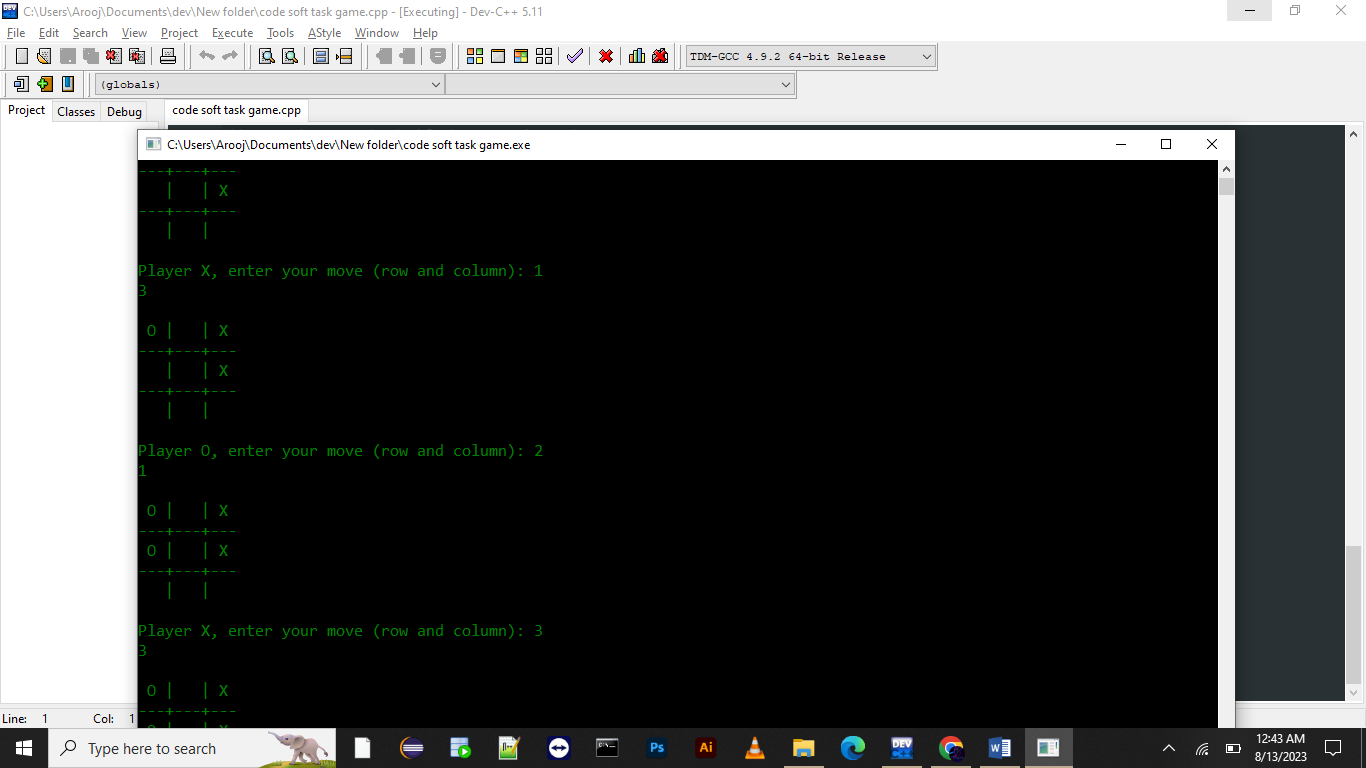
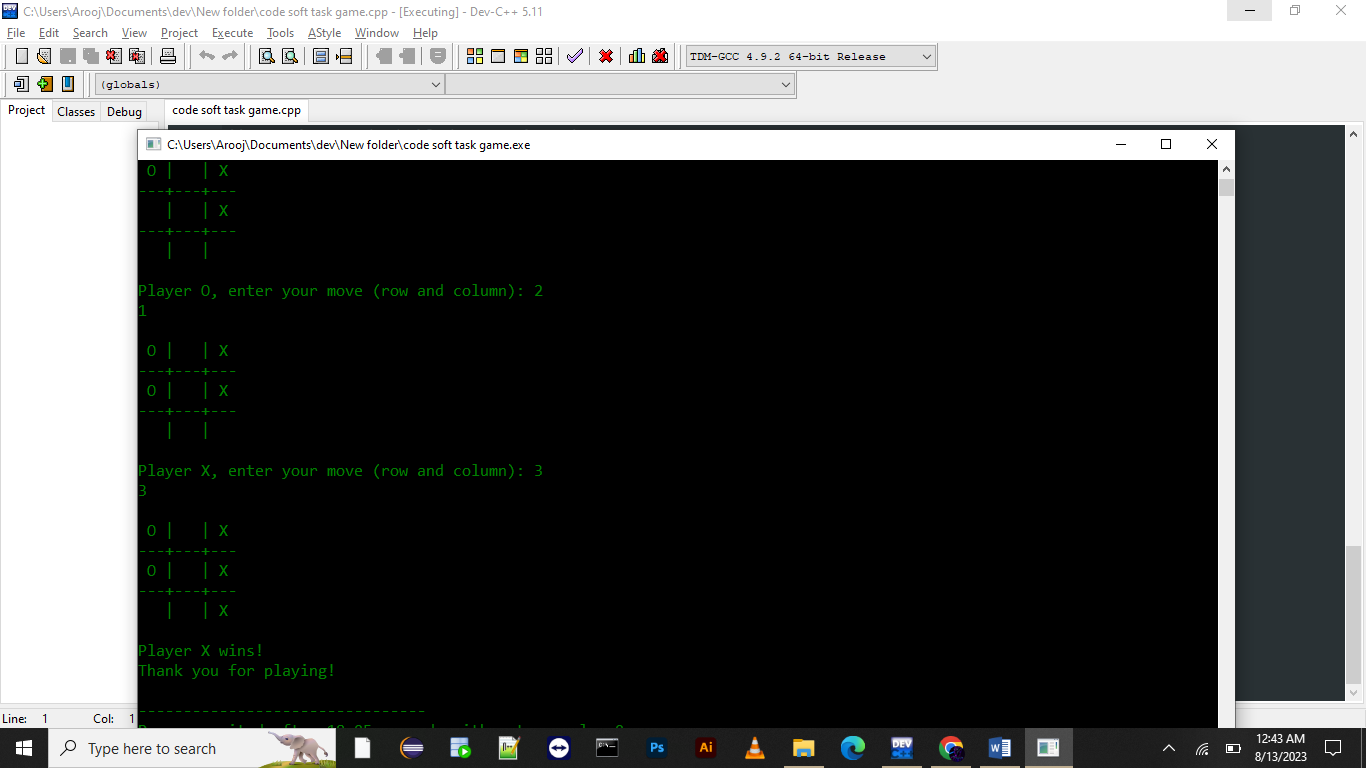
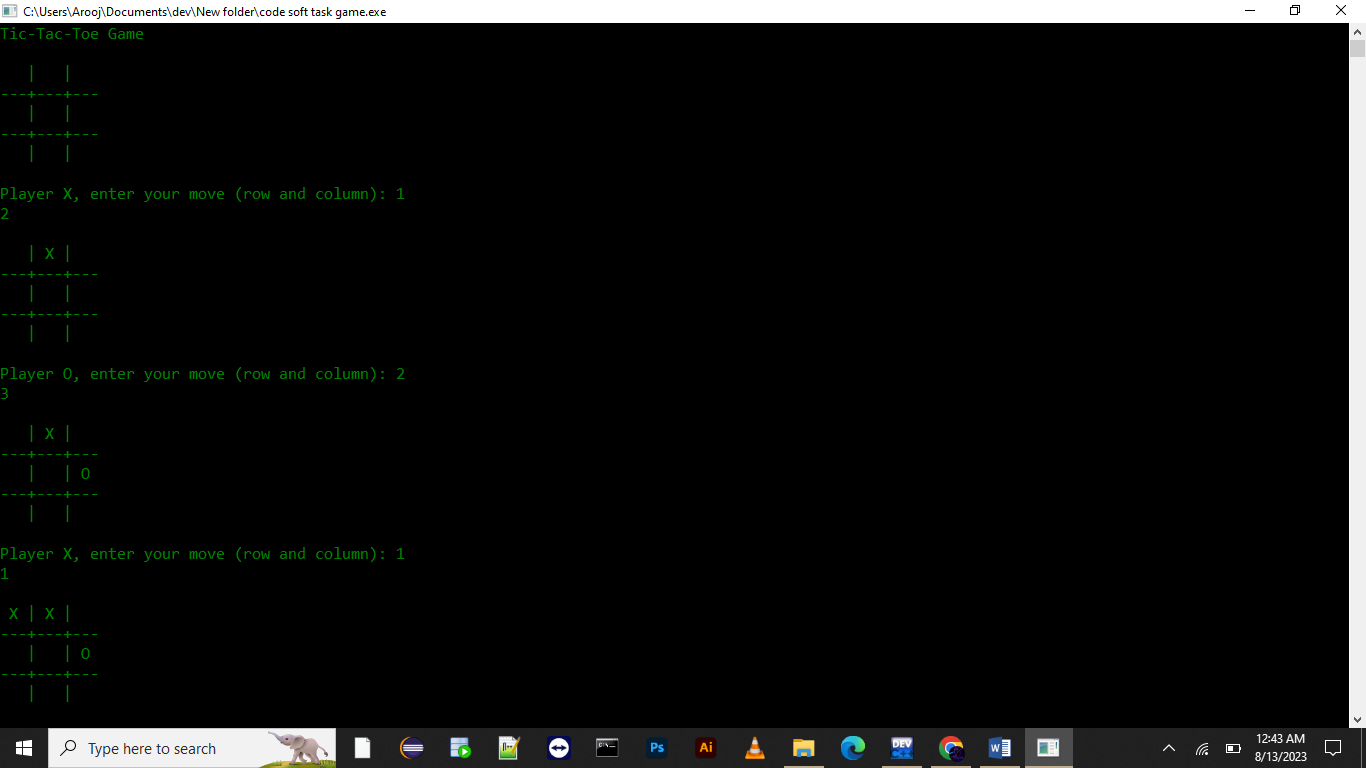
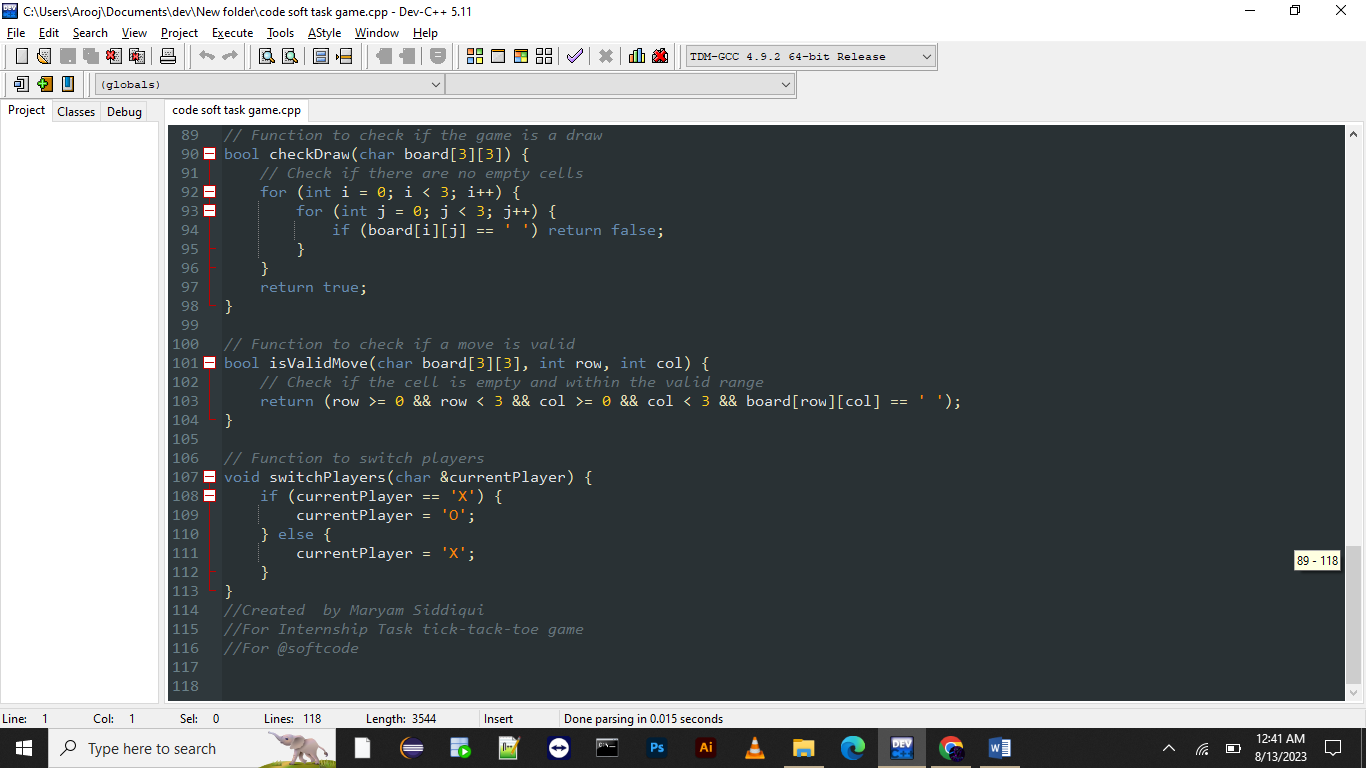
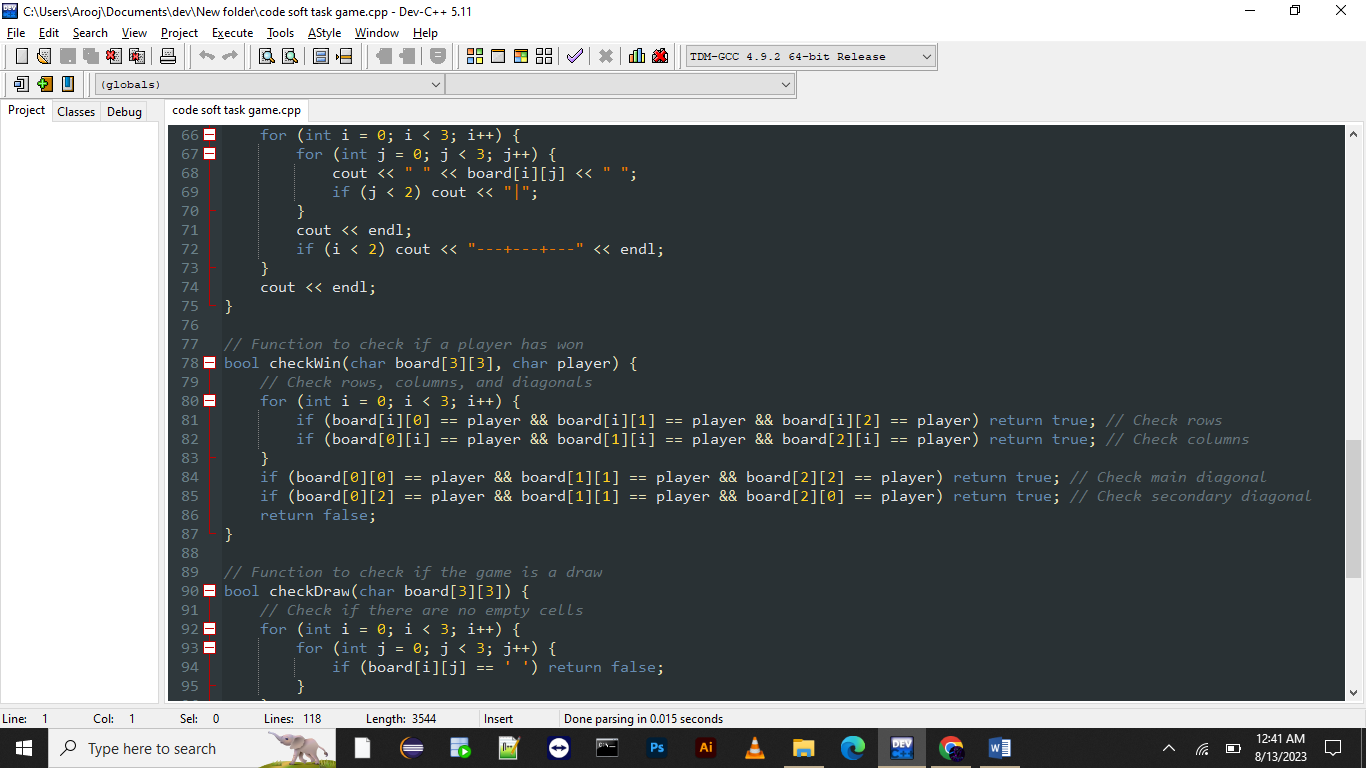
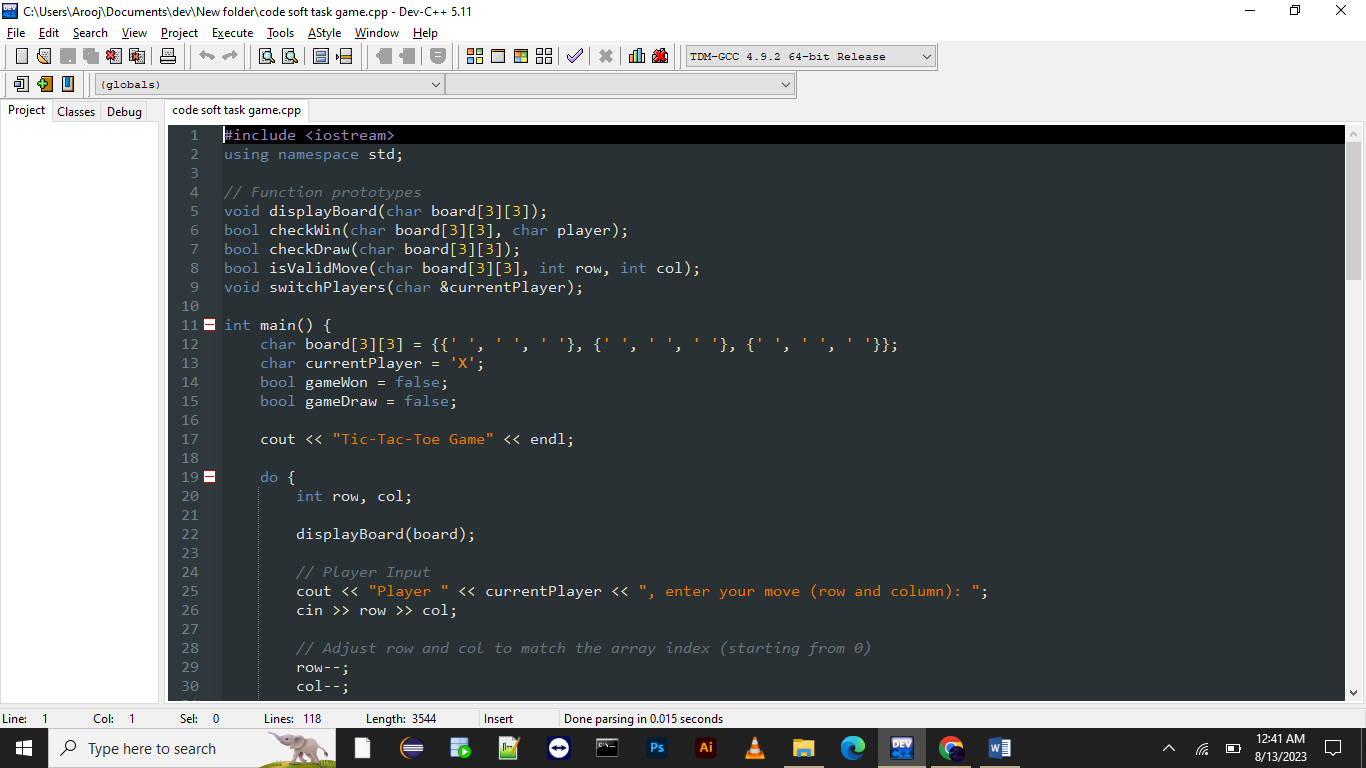
}

**//Created by Maryam Siddiqui**

**//For Internship Task tick-tack-toe game**

**//For @softcode**

**Output:**

****