DOCUMENTATION

TOPIC: MATHS QUIZ GAME

By:

ARISHA MUMTAZ(2312358)

FARHEEN IMAM(2312363)



**ACKNOWLEDGMENT**

We are really grateful because we managed to

complete our project within the time given by our teacher **[sir nabeel and sir abid]**. This assignment cannot be completed without the effort and co-operation from our group members, Group members **[**farheen imam and arisha mumtaz**].**

We also sincerely thank our teacher **[Abid Ali and Sir Nabeel]** for the guidance and encouragement in finishing this project and also for teaching us in this course.

Last but not the least, we would like to express our gratitude to our friends and respondents for the support and willingness directly or indirectly to spend some times with us to fill in the questionnaires.

**CONTENT**

[Introduction 4](#_30j0zll)

[Objectives of the project 5](#_1fob9te)

[Topic of Our Project 6](#_3znysh7)

[Hardware/ Software Requirements 7](#_2et92p0)

[WORK ANALYSIS 8](#_gjdgxs)

Introduction:-

Welcome to the Math Quiz Game – a console-based application designed to test and improve your mathematical skills interactively. Users can participate in a quiz, providing their name and answers to mathematical questions. The program records this information in a "phonebook.csv" file, creating a log of players and their scores.

This game offers a straightforward and user-friendly experience, allowing players to enjoy a quick math challenge and track their performance over time. The accompanying C code provides a foundation for user input, file handling, and data storage.

*Objectives of the project:-*

1. Learn Math Fun: Create a fun quiz to help people enjoy and improve their math skills.

2. Easy to Use: Make a simple game where players can easily enter their name and answers.

3. Track Progress: Keep a record of players and their scores in a "phonebook.csv" file.

4. Quick Challenge: Design a game that offers a quick and enjoyable math challenge.

5. Make It Yours: Allow users to modify the game according to their preferences.

6. For Everyone: Create an inclusive environment where players of all skill levels can join in.

*Topic of Our Project:-*

The Math Quiz Adventure is a fun and educational game created in C. It uses programming to handle quiz questions, user responses, and scoring. Players can customize their experience by choosing quiz difficulty and tracking their progress.

Just like in other games, the Math Quiz Adventure is designed for easy interaction. Players can enter their names, pick quiz options, and enjoy quick math challenges. The game is created to make learning math enjoyable and accessible to all levels of players.

Hardware/ Software Requirements:-

**Software Requirements:**

Operating System: Windows, macOS, or Linux

C Compiler: GCC (GNU Compiler Collection) recommended for compilation

Visual Studio Code or any preferred text editor for code modification.

Terminal or Command Prompt for running the compiled program.

Basic understanding of C programming concepts.

**Hardware Requirements:**

A regular computer or laptop that works with Visual Studio Code.

-Make sure there's enough space on your computer for the Math Quiz Game and its stuff.

The simplicity of the Math Quiz Game ensures that these software and hardware requirements are easily met, making it accessible for anyone interested in exploring and modifying the code.

WORK ANALYSIS

👩‍💻 ARISHA AND FARHEEN’S CODING STRATEGY:🌟

Explore: 🌍 Understanding the Code

Design: 🎨 Designing the Code

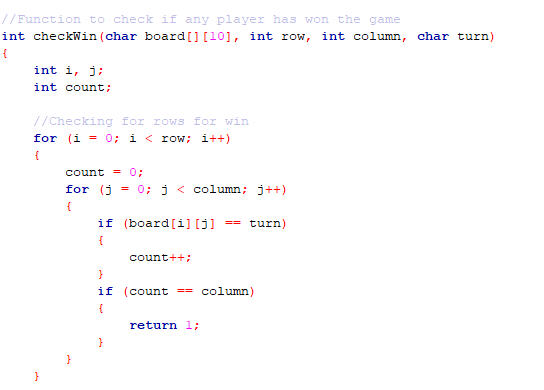
Code: 💻 Building the Code

Test: 🔍 Testing for Code

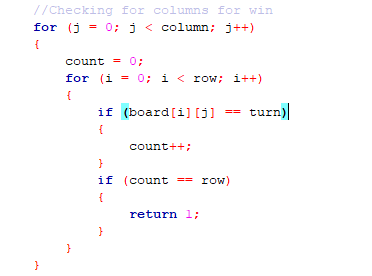
Document: 📜 Documenting the Code

Code Snippets:

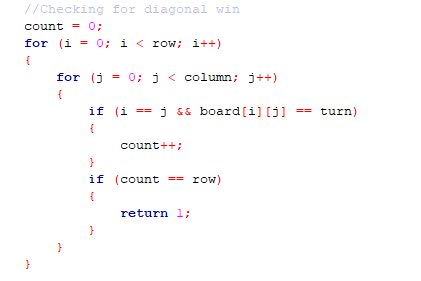
Winning conditions:



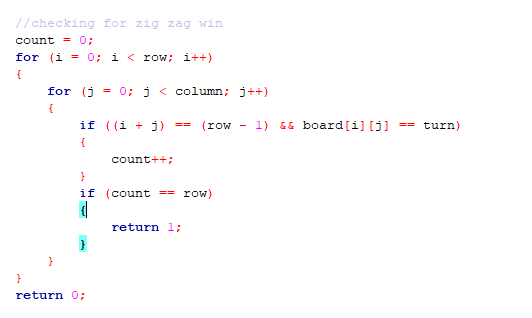
This code is used to check if a player has won a game. It takes in a two-dimensional array (representing a game board) as an argument, as well as the row and column size of the board, and the turn of the player. It loops through each row of the board and increments a count whenever it finds the player's turn. If the count reaches the column size, then the player has won the game and the function returns 1.



This code is checking a two dimensional array (board) for a win by a certain player (turn). The program looks at each column in the array (j), counts how many times the player's turn appears in that column (count), and if the count is equal to the number of rows (row), it returns 1, which signifies a win.

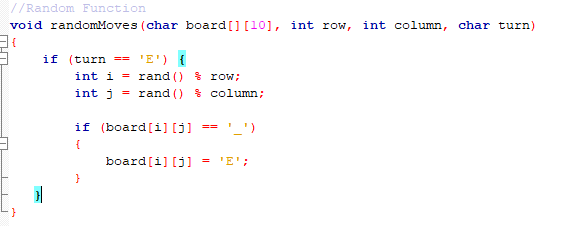


This code is looping through the board array and counting how many times the turn value appears on the diagonal of the board. If the turn appears on the diagonal the same number of times as the number of rows, then the function will return 1, indicating a diagonal win.



This code is checking for a zig zag win in a game. It is looping through a given 2D array (board) and checking for the same turn (turn) on each diagonal. If a diagonal has the same turn (turn) for each row (row) then the function will return 1, signifying a win. If none of the diagonals have the same turn (turn) for each row (row) then the function will return 0, signifying a loss.

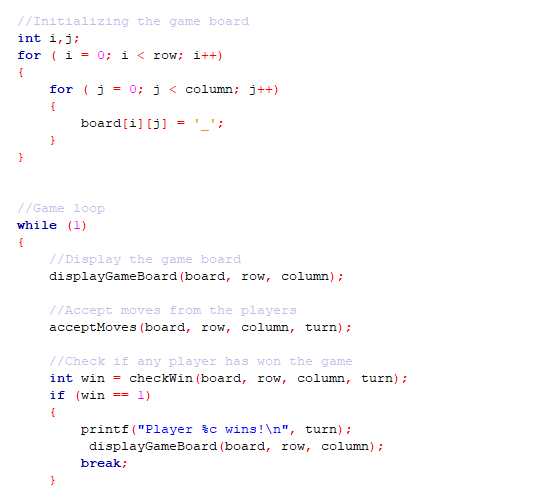
Random:

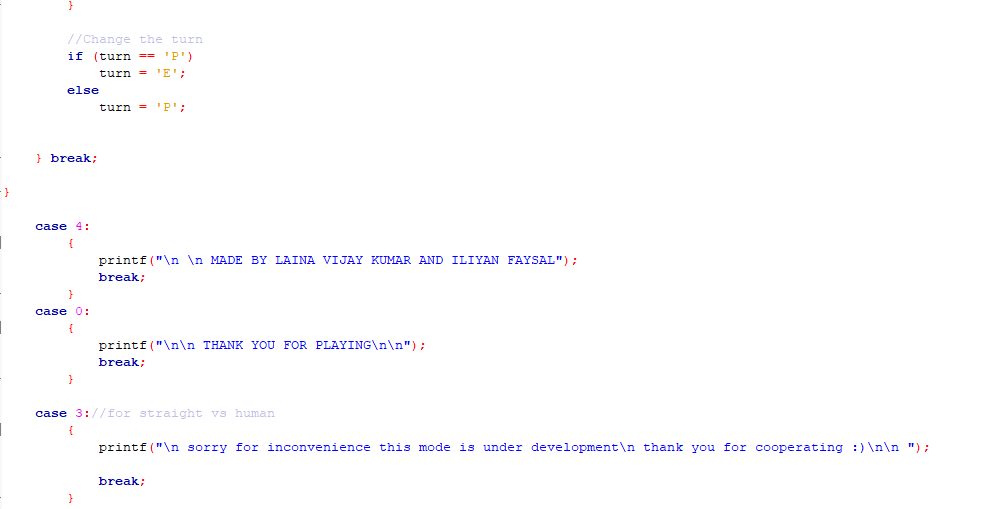


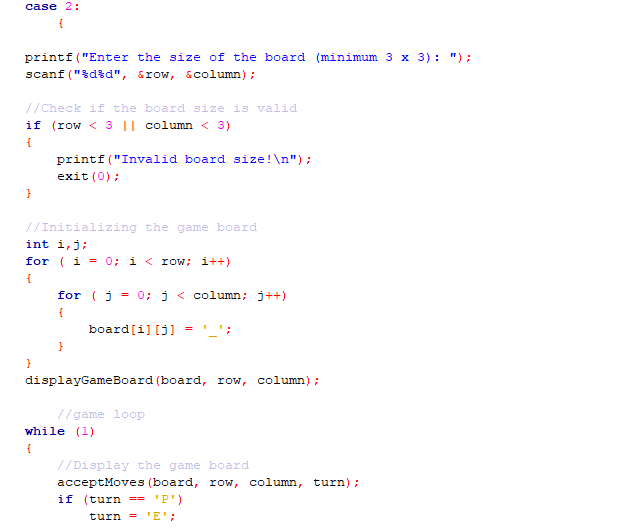
//This code creates a random move for the computer (represented by 'E') in a two-dimensional array of characters (board). It takes in the board, the row and column of the board, and the turn as parameters. It then generates random numbers for the row and column of the board, and checks if the position is empty ('\_'). If it is, it assigns 'E' to that position to indicate the computer's move.

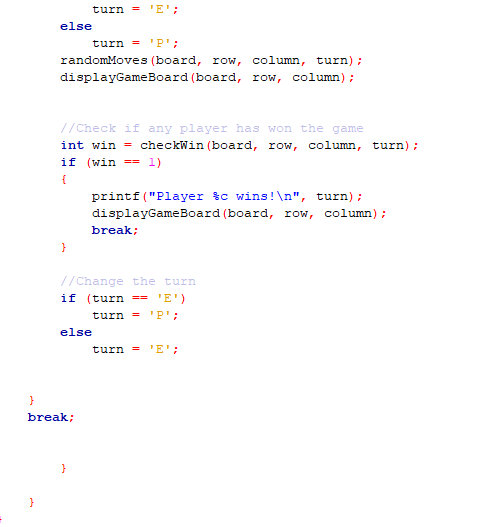
Main:





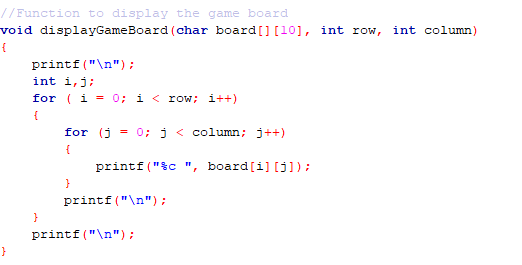






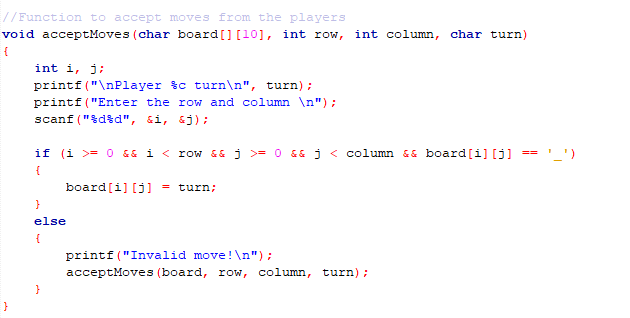
This code is an implementation of the game "Eater-Passer". The code allows the user to choose between four options: Human player vs. Human player, Random player vs. Human player, Straight line player vs. Human player, and Information. The code first asks the user to enter the size of the board (minimum 3 x 3) and creates a board of the chosen size. The game loop then starts and the game board is displayed. The players are then asked to make their moves, and the game checks if any player has won. Depending on the option chosen by the user, either Human player vs. Human player, Random player vs. Human player, or Straight line player vs. Human player is played. If the user selects Information, the creator's names are displayed. If the user selects Exit, the game ends.

Game board:



This code displays a game board stored in a 2D array. The function takes in a 2D array of characters, along with the number of rows and columns, as parameters. The code then iterates through the array, printing each character. Finally, it prints a new line after each row of characters is printed.

Moves:



This code is a function that allows players to make their moves on a board. It takes in the board, the row and column size, and the turn of the current player. It prints a prompt for the player to enter their row and column. It checks if the row and column are valid, and if the spot is empty, and if so it updates the board with the current player's mark. If the move is invalid, it prompts the player to enter another move.

THANK YOU