CT-159

DATA STRUCTURE ALGORITHMS AND APPLICATION

PROJECT TITLE : MATH CLASH

GROUP MEMBERS:

- **Musfirah Shakil (CT-24051)**
- **Kinza (CT-24056)**
- **Falak Naz (CT-24059)**

PROJECT OVERVIEW

Math Clash is an interactive educational game designed to make practicing mathematics fun and challenging. It combines gameplay with learning by presenting players with randomly generated math expressions of varying difficulty, which they must solve under

time pressure. The game supports user authentication, a scoring system, a dashboard to track progress, and a leaderboard for friendly competition.

The game features a **graphical interface powered by SFML (Simple and Fast Multimedia Library)**, which is a modern, easy-to-use multimedia library designed for building interactive applications and games in C++. SFML provides a wide range of features, including handling **graphics, window management, user input, audio, and timing**, all in a simple and efficient way. In Math Clash, SFML is used extensively to create **interactive buttons, text input boxes, and countdown timers**, making the gameplay intuitive and visually appealing. The library also allows for smooth **animation and rendering**, so elements like question displays, progress bars, and level transitions appear seamless. Additionally, SFML simplifies **event handling**, letting the game respond to mouse clicks, keyboard input, and window events with minimal code. Its combination of **performance, flexibility, and simplicity** makes it ideal for educational games like Math Clash, where a responsive and attractive interface enhances the learning experience.

The game also includes a **user authentication system**, allowing players to securely log in, sign up, and manage their passwords. It generates **math questions dynamically**, with difficulty increasing as players progress through levels. Each question is **time-limited**, adding an element of challenge and excitement. Players earn points for correct answers, while wrong answers or skipped questions result in **point deductions**, making the scoring system engaging and fair. The game also lets players **retry questions they previously got wrong**, helping them learn from mistakes. Finally, players can view their progress on a **dashboard** and see how they rank against others on a **leaderboard**, encouraging improvement and friendly competition.

DSA CONCEPTS

1. Vector (Dynamic Array)

In Math Clash, a **vector** is used to store all registered users in the game. This allows the program to dynamically add new users during sign-up and easily search through existing users during login. The vector provides **fast access by index**, making it efficient to update user scores, track game statistics, and save/load all users from the file system. Whenever

a user signs up, their data is pushed to the vector, and during login, the program loops through the vector to verify credentials. Using a vector simplifies **management of multiple users** while ensuring that operations like insertion and traversal remain efficient.

2. List (Doubly Linked List)

A **list** is used to keep track of questions that users answer incorrectly. This is crucial for the "Retry Failed Questions" feature. Using a list ensures that failed questions **maintain their original order** and can be efficiently inserted at the end when a question is answered incorrectly. It also allows quick removal from the front when a user retries a question, without the overhead of shifting elements like an array would require. This makes the retry feature efficient and ensures that the user can attempt questions in the same sequence they were originally answered incorrectly.

3. Stack (Used for Expression Evaluation)

Stacks are used in Math Clash to **evaluate arithmetic expressions** while respecting operator precedence. Two stacks are maintained: one for numbers and one for operators. As the program reads the expression, it pushes numbers to the value stack and operators to the operator stack. When an operator with lower precedence is encountered, or at the end of the expression, operators are popped and applied to values in a **Last-In-First-Out (LIFO)** manner. This ensures that multiplication and division are computed before addition and subtraction, producing the correct answer for each generated math question.

4. Sorting Algorithm (STL sort / Introsort)

In the leaderboard feature, the program uses **sorting** to arrange users by their total scores in descending order. This allows players to see their rank and compare with others easily. Using sort() from the STL ensures fast and reliable sorting, even if there are many users. The program converts the user data into a vector of pairs (score, username) and sorts it so that the highest scores appear at the top, creating an accurate and visually clear leaderboard.

5. Expression Parsing Algorithm (with Precedence Rules)

To generate math questions and compute their correct answers, the game uses an **expression parsing algorithm** that respects operator precedence. This ensures that the evaluation is mathematically correct rather than strictly left-to-right. The algorithm leverages stacks to store numbers and operators, evaluates expressions step-by-step, and produces predictable results. This is essential because players are presented with randomly generated expressions, and the system must know the exact correct answer for scoring and validation.

CONCLUSION

Math clash is a tun and useful game that helps players practice math in practice math in an enjoyable way. It works smoothly beeause of SFML and uses important data struchures like vectors ,list, stack and sorting to manage uses and questions .Overalll the project shows how programming and learning can come together to create a helpful educational game.