Master “C Language” in 30 Days Challenge

(By Tech Involvers)

Project 1: Student Management System

Instructions:

- Read the problem carefully before trying to solve it.
- Do the tasks on your own. Don’t copy it.
- The output of your program must be the same as given in the sample run.

Overview:

Develop a simple Student Management System (SMS) that performs various operations such as adding a student, displaying student details, searching for a student, and updating or deleting student records.

Requirements:

1. **Student Structure:**
   - Use a `struct` to represent a student. Include fields like `ID`, `name`, `age`, `gender`, and `marks`.
2. **File Handling:**
   - Store student data in a file to persist the data even after the program ends.
   - Implement functions to read from and write to the file.
3. **Functions:**
   - Add Student: Add a new student record.
   - Display Students: Display all student records.

- Search Student: Search for a student by ID.
- Update Student: Update details of an existing student.
- Delete Student: Delete a student record.

4. **Enumerations and Macros:**
   - Use `enum` to define student gender.
   - Use macros for constants like the maximum number of students.
5. **Pointers and Arrays:**
   - Use pointers for dynamic memory allocation if needed.
   - Use arrays for storing student records in memory.
6. **Error Handling:**
   - Implement error handling for file operations and invalid inputs.

Features Covered:

1. **Structs and typedef:**
   - `Student` structure and `Gender` enum.
2. **File Handling:**
   - `loadStudents` and `saveStudents` functions.
3. **Functions:**
   - Adding, displaying, searching, updating, and deleting students.
4. **Error Handling:**
   - Checking for file operations and array bounds.
5. **Pointers and Arrays:**
   - Using arrays to store student records and pointers for dynamic memory (if extended).
6. **Input/Output and Operators:**
   - Using `scanf` and `printf` for input and output, and various operators within functions.