C Programming Final Project

Introduction:

It is required to implement a simple **Student Management System** for a university that supports adding, removing, retrieving, and updating student data.

The information stored in the system for a single student is as follows:

- Student Name: A single string representing the student's first and last name (Ex: Hussam Wael)
- Student ID: a 7-digit code that is unique for each student (Ex:1209654)
- Student Gender: A single character representing the gender of the student ('M' or 'F')
- Academic Year: An integral number between 1 and FACULTY_MAX_ACADEMIC_YEAR (Should be implemented as a configurable constant)
- GPA: a positive floating-point value between 0 and 4.0

Main Menu Description:

Upon running the program, a menu should be displayed like the following:

```
/*************************************************************************/
                    Student Management System
/*************************************************************************/

Choose the desired operation:

    1- Add a new student
    2- Remove a student
    3- Retrieve student's data
    4- Update student's data
```

The user shall choose the operation by entering its number and should get an error message if they enter an out-of-range value. In this case, the system should ask the user to re-enter a valid number.

Operation 1 Description: Adding a student:

Upon choosing to add a student record, the system should ask for all the fields of information mentioned above. The system should also perform checking of the values of the entered fields if they are valid, and if they're not an error message should be displayed, and the system should ask the user to re-enter the values **starting from the error field**.

Expected error checks:

- Student Name: must consist of the first and last names only. The system should check that this field consists of exactly two words.
- Student ID: Must be a 7-digit integral code with no leading zeros, and any other values should not be accepted. Also note that the Student ID must be UNIQUE, so entering an ID of another existing student is considered invalid.
- Student Gender: It's either 'M' or 'F'.
- Academic Year: An integral value between 1 and FACULTY_MAX_ACADEMIC_YEAR is only accepted.
- GPA: Must be a positive number between 0 and 4.0.

Upon entering all fields correctly, the system should add the new student record to its internal storage, and it should display a message indicating that the new student was added successfully.

Operation 2 Description: Removing a student:

Upon choosing to remove a student record, the system should ask for the ID of the student they seek to remove their data. It should check that the entered ID is correct before removing the student record with the given ID.

The possible outcomes of this operation are as follows:

- Invalid ID: If the value of the entered Student ID is invalid.
- ID Not Found: If the entered Student ID does not exist among the student records stored inside the system.
- Operation Successful: When the system successfully removes the student's data.

Operation 3 Description: Retrieving Student's Data:

Upon choosing this option, the system should ask the user in which way they want to search:

1) Search by name
2) Search by ID

Note that the ID is unique, so the system will display at most one student record. On the other hand, there's no uniqueness constraint on the student's name. This means that if the user chooses to search by name, then if multiple students have the same name, the system should display all their data records.

Error checks should be performed on the search attribute before using it to search for the student's data. The possible outcomes of this operation are as follows:

- Invalid ID/Name: When the search field value entered is invalid.
- Student Not Found: If there exists no student with the given ID/Name.
- Operation Successful: When the system can retrieve the information of the student(s) and display it.

Operation 4 Description: Updating student's data:

Upon choosing this option, the system should ask for the ID of the student the user seeks to update their data. Error Checking is performed on the ID and if it's correct, the system should ask the user which field they want to update out of the following:

1) Academic Year
2) GPA

Upon choosing the desired field, the system should request the user to enter the new value of this field, perform error checking on this value, and update the student record if the value is valid.

General Guidelines:

- This project is a command line project, there's no need for any GUI.
- You can use <stdio.h> for receiving inputs and displaying outputs on the terminal. You can use one of the following IDEs: (VS Code, Code Blocks, Eclipse,etc …).
- Divide the system into multiple modules, where each module has its own (.h) and (.c) files.
- Make sure your code is clean and well commented. Preferably use the naming conventions we discussed during the last session.
- Each function you implement should be commented on with a description of what the function does, what arguments it needs, and what the return value of it is.

Bonuses (Optional):

- Add a fifth option in the main menu that generates a statistical report using the students' data. For example, the report can show the user the top three students with the highest GPAs for each of the academic years.
- Upload your project on a GitHub repository and add the repo link to your submission.
- Find a way to store the student's data persistently, so that when the program restarts, the data of the students entered before would still be present in the system.

Submission Details:

- A google form will be uploaded on our channel on Microsoft Teams for project submissions.
- Project Deliverables are:
    - Zip file containing all the necessary files to run your project.
    - GitHub repository link. (Bonus)
- **Project Deadline: Thursday 17/8 at 11:59 PM**