Data Structure Lab Assignment (CS 2172)

Assignment 1 : Array Concepts
Time: 1 week

Problem

Under this assignment you are required to create a suitable data-structure and associated functions for maintaining a simple student-register.

The Student Register

- **The Data Structure**: The maximum number of entries (students) that the register should support is a predefined constant. Each entry (student) in the register should consist of the following fields.
    - **name**: a null-terminated string containing the last and first names separated by a comma, for example "Tagore, Rabindranath". This string cannot be empty.
    - **roll**: a 6 digit integer number unique for each student. A roll number cannot start with 0.
    - **telephone**: a null-terminated non-empty string containing a telephone number.
    - **address**: consists of null-terminated strings containing address. These strings can be empty.

- **Associated Functions**: Let ***SReg*** and ***Student*** be the user-defined data-types for a student-register and a student, respectively. The Student Register should support the following operations.
    - ***int add(SReg sr, Student s)*** – adds a new student ***s*** to the student register ***sr***. A new student is a student for whom the roll does not already exist in the student register ***sr***. If ***s*** already exists in **sr** (that is, the roll field of ***s*** matches with roll field of some entry of ***sr***), the function returns ***0***, otherwise the function returns ***1***.
    - ***Student get(SReg sr, int r)*** – returns the student from ***sr*** whose roll field matches with ***r***. If there is no such student in ***sr***, the roll field of the returned ***Student*** is 0.
    - ***int delete*(*SReg sr, int r)*** – deletes the student from ***sr*** whose roll field matches with ***r***. If there is no such student in ***sr***, the function returns 0; it returns 1 otherwise.
    - ***int modify*(*SReg sr, Student s)*** – updates the fields of an existing student of ***sr*** whose roll field matches with that of ***s,*** taking values from the corresponding fields of ***s***. The function returns 0 if no such student exists in ***sr***; otherwise it returns 1.
    - ***sortStudents(SReg sr)*** – sorts the students of the student register ***sr*** in alphabetically ascending order of names.
    - ***int getCount(SReg sr)*** – returns the number of students in the student register ***sr***..
    - ***export(SReg sr, String fname)*** – saves the student register ***sr*** to a file having name ***fname***.

- ***load(SReg sr, String fname)*** – loads students in the student register ***sr*** from the file having name ***fname***. Please note that a file generated by the ***export(SRef sr, String fname)*** function can be used by this function.

C - like syntax has been used in the above specification of the student register. Please note that they are just suggestive. You apply your own judgement in choosing the data types of the functions and there parameters.

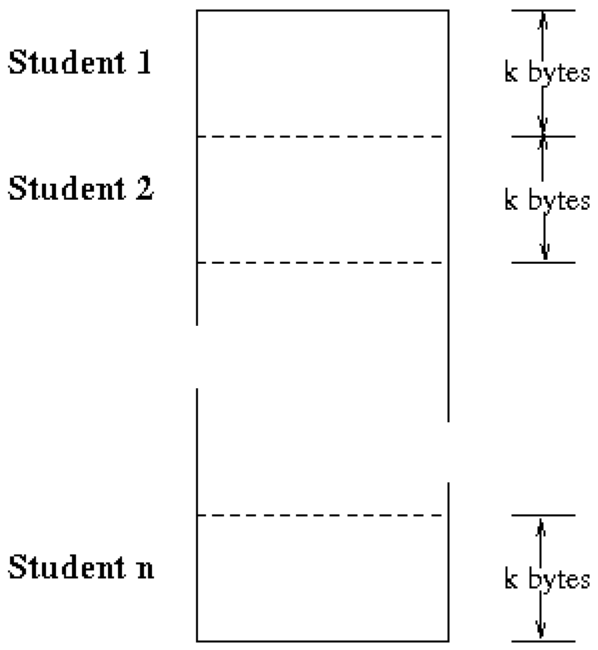
Implement the student register as a C program with all the above mentioned functionalities. Write a suitable main() function for demonstrating that your student register program supports all the features.

Extension: Should be attempted once the above portion is fully completed

Note: Keep the previous code and create a new code to develop this extension

Extension:

This is an extension of the previous assignment. Let all the (say ***n***) student-records of a student-register be of the same size (***k*** bytes, say) and when exported be stored in a file as shown in the following figure.



File containing a Student Register

Add a function "***int exportBin(SReg sr, String fname)***" – saves the student register ***sr*** to a file having name ***fname*** and storing student records in the above mentioned manner. This function returns 0 in case of any error while exporting. Otherwise it returns 1.

Add a function "**sortBin(String fname)**" that sorts the student records of the student register which is available (has been exported earlier using ***exportBin()*** function) in the file ***fname*** in the above mentioned manner. Please note that the records in the file should be sorted keeping them in the file itself - without reading all the records in the register and then sorting.

Add a function "***int loadBin(SReg sr, String fname)"*** – loads the student records from the file ***fname*** into the student register ***sr***. The file fname is expected to have student records in the above mentioned manner (that is, which has been created by some earlier call to ***exportBin()*** function). This function returns 0 in case of any error while loading. Otherwise it returns 1.

Write a supporting main() function to demonstrate that all your functions are working as expected.