Final project

Student record management

Program description

Using the C programming language, your group needs to develop a student management system which will help our academic assistance to manage the information of students belonging to B1 classes of this academic year.

The program works as follow:

No	Tasks	Techniques to be implemented	Task type
1	**Enter student number:** The program asks users to enter number of students (maximum 1000)	Declare an integer variable to hold this value Use scanf() function to read value from keyboard	**Common tasks** (same for every group)
2	**Enter students' information:** for each student, user type the studentID, full name, birthdate, overall grades of three subjects (Algebra, Calculus, Basic Programming).	Use array to store data Exploit the struct to organize data structure better	
3	**Print student list as a table to screen:** The program prints out the student list with the above information in the table format. The header includes following columns: studentID, full name (format: lastName then firstName), birthdate, Algebra, Calculus, Basic Programming, GPA. Each row corresponds to data of each student.	Just iterate throughout the array and use printf function with proper format.	
4	**Print student list as a table to a text file:** The program prints out the student list with the above information in the table format. The header includes following columns: be studentID, full name, birthdate,	Just iterate throughout the array and use printf function with proper format.	

	Algebra, Calculus, Basic Programming, GPA. Each row corresponds to data of each student.		
5	**Process grades:** find the students having highest GPA, lowest GPA, highest BP grade.	Write the 03 functions for each task	
6	**Print out student lastName:** to the screen.	Process the string to find the last word in the name of each student.	**Selective tasks** Selecting **2 from 5** for each group as below: **STask_number_1** = 6+GroupID %5 **STask_number_2** = Task_number_1 +1 (if STask_number_1=10, STask_number_2 = 6)
7	**Find the oldest student:** print the oldest student with his/her ID, fullname, and birthdate	Write the function in 02 separates files (*.h and *.c)	
8	**Find the youngest student:** print the youngest student with his/her ID, fullname, and birthdate	Write the function in 02 separates files (*.h and *.c)	
9	**Search by studentID:** user enters an ID, the program will show the found record with all information; otherwise, the program will announce there is no matched ID.	iterate throughout the array and compare studentIDs with entered value.	
10	**Sort the student list by GPA in descending order:** print out the sorted list	Use a sorting algorithm e.g. bubble sort to sort the array	

Timeline

Lecture session 8: Kick-off project → list of groups

- Form up groups

- Lecturer instructs students to design and organize the project

Lecture session 9 (last session): Progress update → group work evaluation

- Lecturer comments group works

- Lecturer asks one by one group to evaluate their group work

- Students finalize and submit (source code + group work report + individual report)

Things to submit

The leader of the group submits:

1. Source code in a folder (if there are more than one file), please **zip** it.

Each student (including leader) submits:

1. Individual report

Instructions for the individual report:

- Compiling instructions: the commands used to compile the project and screen shots proof that you compiled successfully.
- Screen shots for testing your program: enter the data according to your group, i.e. if there are 5 students, you should type number 5 at 1st functionality, then enter data for each member.
- Each student submits: screen shots of your running task x^{th} among 7 tasks of your group, that is computed as below:

  x = 3 + your_student_ID%5 (your_student_ID = 3 last digits in your student ID)
- For example:

  Group 18 will do **7 tasks** = [1,2,3,4,5, STask_number_1, STask_number_2]

  STask_number_1 = 6 + 18%5 = 9

  STask_number_2 = 9 + 1 = 10

  A Student has the ID 22bi13**309**, member of Group 18, will submit the screen shot of the task $x^{th} = 7^{th}$ of 7 tasks of Group 18, meaning the task 10.

  x = 3 + 309%5 = 7