Section 16

PDS Lab Lab-9 03.11.2023

Instructions:

- This lab is based on the topics: Structures
- You should save each program with the file name as specified against each problem as <Lab#>_<Assignment#>_<Roll#>.c. For example, **09_01_23CS10006.c** to save Program to 1st assignment in Lab 9 with Roll Number 23CS10006
- You should upload each program to the Moodle system. Also, copy+paste your programs to the text window on the test page.
- There will be no evaluation and hence grade, if you don't submit your .c files to the Moodle server. Use **emacs** editor and **gcc command** in terminal to run the following programs.
- Document your programs meaningfully using appropriately named variable and sufficient amount of comments. Documentation and proper code indentation carry marks.
- **The top two lines of your programs must contain the following information:**

  //Roll No.: <Type in your roll no.>
  //Name: <Type in your name>

Document your programs meaningfully using appropriately named variables and sufficient amount of comments.

1. Write a C program to build the following software:

Student Academic Record Management Software: Write the following functions to implement a student academic record management software.

Define the following variables with global scope:

- A structure named **student** consisting of the first name of student (20 characters), last name of student (20 characters), address (40 characters), mobile number (unsigned integer), grades in five subject (array of five characters), and gpa (float).
- An array named **studArray** containing 50 **student** structures.
- An integer named **nStuds** indicating the number of students in the class.

main: Set to **nStuds** zero (it indicates the next available **student** structure in **studArray)**. In an infinite loop, display the following menu, and then prompt the user to enter a number between 1 to 8.

1. Enrol student
2. Enter grade
3. Display students
4. Search student
5. Edit student grades
6. Modify Student Details
7. Display merit list
8. Exit the program

Depending on the user input, carry out the following:

1. Call the function **enrolStuds**
2. Call the function **enterGrades**
3. Call the function **displayStuds**
4. Call the function **searchStuds**
5. Call the function **editStudGrades**
6. Call the function **modifyStutDetails**
7. Call the function **dispMeritList**
8. Exit the program

enrolStuds: This function should first prompt the user to enter the number of students (**nStuds)** and then prompt the user to enter First Name, Last Name, address, and mobile number for each student and store the details of each student in the next available structure in the **studArray**. The students should be given roll numbers serially starting from 1. After the data for all the **nStuds** number of students have been stored in **studArray**, display the stored data with proper formatting.

enterGrades: This function should display the Roll number, First Name, Last Name of each student upto **nStuds** one at a time and prompt the user to enter letter grades in five subjects (a letter grade can be any of: E, A, B, C, D, P, or F) for each student. Any other character entered should be rejected, appropriate message displayed, and the user be prompted to enter again. Based on the entered letter grades, the GPA should to be computed and stored at appropriate place in the **studArray**. The grade to points conversion rule is E=10, A=9, B=8, C=7, D=6, P=5, and F=0. Assume the five subjects have equal credits. Display the data for all students with proper formatting.

displayStuds: It should first sort the **studArray** based on the first name of the students and then display the student details so that the first names appear alphabetically sorted. For the students whose first names are exactly the same, then sort these students based on the second name. Assume that no two student will have the both the same first and last name. **Note:** Use of appropriate string library function permitted.

searchStud: This function should prompt for a string to be entered and then display the details of all the students whose either the first name or the last name matches with the entered string. The matching of the strings should be case insensitive.

modifyStudDetails: This function should prompt for entering the roll number. It should then display the name, address and mobile number of the student and ask for the new address and mobile number. Finally, it should display the updated details of the student.

editStudGrades: This function should first prompt for the roll number. Based on the entered roll number, it should display all the details of the matching student including the current letter grades and prompt to enter the updated grades. Then the GPA should be calculated based on the updated grades and stored. The details of the student should be displayed in proper format after the update.

dispMeritList: This function should first sort the students in **studArray** in decreasing order of GPA. When there is a tie, the students should be sorted in alphabetic order based on the last name. Display the student details in merit order.