Attempt Every Part of the Following

University Registration System

It is required to write a C program to perform all the functions of the credit hour registration system of an educational institution (e.g. Alexandria University – Faculty of Engineering). The program should include a data structure for each one of the possible students available in the institution. The data structure should include at least the following:

1. Student name, which is to be defined as a structure containing the following:
   a. First name
   b. Middle name
   c. Last name
2. Student registration number, which may be automatically generated for new students or entered (e.g., 2020-term 2-student number 13 => 2020013) by the user according to the user requirements.
3. Student nationality.
4. Student ID, which is a structure containing the following:
   a. type of ID (e.g., national ID, passport, …)
   b. ID number
   c. Validity of ID which is a structure of the type date as will be explained later on in the student age section
5. Student home address which is a structure containing the following:
   a. Flat number
   b. House number
   c. Street name
   d. Area/District name
   e. City name
   f. Governorate name
   g. Country
   h. Postal code
6. Student mailing address which is a structure containing the same as the previous address field. Please note that the operator should only enter this address in case it is different from the home address. Otherwise the program has to set the data to be the same automatically.
7. Student age (calculated automatically) in addition to his/her date of birth, which is a structure containing the following:
   a. Day of birth (restricted between 1 and 31)
   b. Month of birth (restricted between 1 and 12)
   c. Year of birth (restricted between 1980 and present date)
8. Student gender.
9. First enrolment term and year.
10. Student department of enrolment.
11. Student tutor name and code.
12. Current term, which is calculated according to the credit hours finished.
13. Payments, which is a structure containing the following:
    a. Type of funding (e.g., self-funded or grant-funded)
    b. Number of payments done
    c. Total amount paid so far
    d. An array of structure containing the details of each payment. These include:
       i. Amount paid
       ii. Transaction number

   iii. Bank of transaction
   iv. Date/Time of transaction, which is a structure for the date containing the day, month and year of the transaction as explained above in the date of birth.

14. Student contact, which is a structure containing the following:
    a. Contact e-mail of the student
    b. Student home phone number
    c. Student mobile number
15. Credit hours, which is a structure containing the following:
    a. Maximum required for the degree
    b. Finished
    c. Current
    d. Remaining (The remaining hours = Maximum -Finished – Current)

    The maximum credit hours required should be entered by the operator. The finished and current hours should be computed from the sum of finished and current courses' credit hours entered in the field relating to the course finished and current. The remaining hours should be computed from the equation presented above.
16. Student GPA, which is a structure containing the following:
    a. Overall GPA attained so far by the student (should be initialised to zeros for a new student).
    b. Number of semesters finished so far.
    c. An array of 10 elements constituting the individual semester GPAs (should be initialised to zeros for a new student).
17. Courses, which is a structure containing the following:
    a. Finished
    b. Current (number of courses bound by the student status calculated above)

    Each one of the above (Finished and Current) group of courses is a structure containing the following:

    a) Number of courses
    b) Total number of credit hours for these courses
    c) Array (of size 100 for the finished courses and 8 for the current courses) of structure for the course data, which should include the following:
       i. Course code
       ii. Course name
       iii. Course credit hours
       iv. Number of times taking the course
       v. Semester and year for taking the course in the case of finished courses
       vi. Course tutors which is a structure containing the following:
          - Course lecturer (first, middle and last) name
          - Course GTA (first, middle and last) name
          - Course lab GTA (first, middle and last) name
       vii. Marks acquired during the course which is a structure containing the following:
          - First mid-term marks out of 20 (may containing fractions)
          - Second mid-term marks out of 20 (may contain fractions)
          - Total year work marks out of 40 (may contain fractions)
          - final exam marks out of 60 (may contain fractions)
          - sum of all the above out of 100 (may contain fractions)
       viii. Grade of the student in this course according to the standard grading scheme
       ix. Student percentage absence during the course

<u>For the above data structure, it is required to perform the following:</u>

1. Define a new type in C to incorporate the above data structure as well as all its requirements.
2. Implement the above data structure in C.
3. A dynamic array containing the data of all the students should be defined as a local variable inside the "main" function. This array should be sized so as to exactly contain the data of the students defined in the program according to the new type implemented above. Dynamic allocation is used to ensure no waste of space occurs in the memory.
4. Write functions to perform the required tasks from the system. Such functions may include (but are not limited to) the following:
   a) A function to read the data of one student from the user.
   b) A function to print the data of one student on the screen.
   c) A function to read the data of one student from a text file.
   d) A function to print the data of one student in a text file.
   e) A set of functions, which would return (get) the value of each of the items belonging to the data of a certain student (e.g., a function to return the age of a student, a function to return the GPA of a student, ...)
   f) A set of functions which would write a new value (set) to the value stored in each one of the items belonging to the data of a certain student (e.g., a function to set the value of the age of the student, a function to set the value of the GPA of a student, ...)
   g) A function to calculate the age of the student.
   h) A function to calculate the current number of credit hours finished and being currently studied
   i) A function to calculate the overall GPA of the student in case of any of his/her course data is edited.
   j) A function to enter a new payment for one of the students with all its details outlined above.
   k) A function to edit/change the contact details of one of the students (e-mail, phone number and mobile number).
   l) A function to sort and retrieve students of a certain department and batch based on their GPA.
   m) A function to retrieve students coming from a certain family to apply discounts on paid fees.

<u>Please take into consideration the following:</u>

- Any function taking an input from the user in order to set a value in a student record should test the validity of the data presented to it before committing the changes otherwise, erroneous data changes may occur beyond repair in the database.
- The program should present the user with a user-friendly environment, which allows the user to select one of the allowed options involved. The main screen of the program should list the options (tasks to be performed) on the screen in a numbered format. The user is then prompted to enter his/her selection. The program will then execute the required task and print the result on the screen or at least a message stating to the user that the task has been performed correctly. The options in the main screen should include the functions stated above.

Best Wishes.