CSE 102 Spring 2024 – Computer Programming Assignment 4

Due on March 27, 2024 at 23:59

You will write a program letting a user ask questions about a set of people whose information are kept in two seperate text files.

The first text file provides a set of information for each person (could be repeated). The following are the possible formatting of these info:

- A person is defined by their <id> followed by some other information.
- Along with an <id>, a student will have course information as shown below:
  <id> ; <midterm1> ; <midterm2> ; <final> ; <course_id>
  or:
  < id> ; <midterm> ; <final> ; <course_id>
- Similarly, an instructor will have:
  <id> ; <# of classes> ; <department>

The second text file relates the <id> of a person with <name>, <surname> and <department> etc. At each line:

- A person will have an <id> followed by <name>, <surname>, <role> and <department> information seperated by ';'.
  <id> ; <name> ; <surname> ; <role> ; <department>

Note that <id>, <midterm1>, <midterm2>, <midterm>, <final>, <course_id>, and <department> are all integers. The rest words (a series of characters without whitespaces).

Part 1. Utility functions [60 pts]

You should implement the following functions:

- char first_initial (FILE * id_fp, int id): A function that goes through the id file and finds the initial letter of the first name of the person with the given id.
- char last_initial(FILE * id_fp, int id): Above function returning the initial letter of the person's last name.
- int get_id_fi(FILE * id_fp, char first_initial): Returns the id of a person given the first initial.
- int get_id_li(FILE * id_fp, char first_initial): Returns the id of a person given the last initial.
- int average_grade(FILE * info_fp, int id): Calculate and return the GPA of a person defined by the given id.

TIP: You can create and destroy extra files for additional dynamic storage. If you believe you need to include additional functions, you can add your own functions. However, ensure that you maintain the existing format without altering it.

Part 2. Query [40 pts]

Write a program (including your functions above) that does the following:

- Display a menu of several choices:
  - 'p': Print all the users pass or fail.
    - Your program goes through the entire list of students and their exams.
    - For each course, calculate the letter grade using the table below.
  - 'n': Print only the user with a specific first initial.
    - Ask user to enter a letter for the initial (first name).
    - Do the same thing as in 'p' only on the matching students.
  - 'g': Calculate the GPA of a given student.
    - Ask user id.
    - Find all the grades for this user and calculate average grade (using the letter grade conversion given in the table below).
  - 'c': Print whether each user passed or failed the same class.
    - Ask user to enter course_id.
    - Find all the grades for this course and calculate average grade (using the letter grade conversion given in the table below).
  - 't': Print the number of classes for the instructor.
    - Ask user id.
    - Find the # of classes for the instructor
  - 'd': Print the department of all persons according to the role.
    - Ask user to enter a letter for the initial (role).
    - Find all persons id and department matching with the role.
  - 'l': Print the course_id of a given student.
    - Ask user id.
    - If this person does not a student, give a warning.
    - Find course_id with using id.
  - 'e': Quit the program.
- Allows user to continue selecting a choice and continuing with another one or exit.
- If there are two midterm marks, you can use this formula for the average, etc.

$$\boldsymbol{M = \frac{M_1 + M_2}{2}}$$

- To calculate GPA, take the relevant Midterm GPA and Final GPA grades in the table, sum them, and divide by two. If the result does not correspond to any letter or GPA grade in the table, round the result.

$$\boldsymbol{GPA = \frac{M + F}{2}}$$

Midterm < 40 and Midterm<MidtermAverage	Final < 40 and Final<FinalAverage	F	1
Midterm > 40 and Midterm<MidtermAverage	Final > 40 and Final<FinalAverage	D	2.5
Midterm > 40 and Midterm>MidtermAverage	Final > 40 and Final>FinalAverage	C	3
90>Midterm> MidtermAverage	90>Final>FinalAverage	B	3.5
Midterm=>90	Final=>90	A	4

Notes:

- You should use swtich-case when taking an option from user.
- You should use “first.txt” and “second.txt” when reading files.
- In this homework, using **arrays and strings are not allowed**. If you don’t take serious this warning you will get **-50** for this assignment.

IMPORTANT NOTES:

- Submit your homework as a zip file named as your student id (StudentID.zip) and this file should include:
  - YourStudentID.c file
  - A pdf file named "YourStudentID.pdf" including a YouTube link and screenshots of your program outputs. In the video, you are expected to provide a demo of your assignment. For each requested functionality, you must explicitly explain your solution approach and also execute and display the outputs. The video should not exceed 4 minutes. Please ensure that your camera is turned on during the recording.
- The output format must be as given, do not change it.
- Compile your work with given command “gcc --ansi your_program.c -o your_program”.
- Your work will be evaluated using gcc version 11.4.0.
- For any questions and problems, you can always contact me **via email** (nbengisucolak@gtu.edu.tr), or you can find me in Room 122 during scheduled office hours on March 13-27, 2024, between 10:30 and 12:30.