

Lab 10

© INSTRUCTOR: DR. MD. MAHFUZUR RAHMAN

Lab Quiz: 30 Points, Problem Solving: 70 Points

Objectives:

Today we will be covering the following topics:

1. Practice **structures** in a C program.

Instructions:

- Attendance is mandatory.
 - Labs must be completed individually.
 - If you have any questions, please do not hesitate to ask TA.
 - Follow submission instructions in the deliverable section.
 - There will be a lab quiz of 30 points arranged by Lab TA.
 - Visit the broader grading criteria after the deliverable section. (last page)
 - Lab assignments are due at midnight on the day of your lab (i.e., by 11:59 PM)
1. Develop a C program to manage student grades in a class. It should allow users to create a class roster, input grades for each student, and calculate and display various statistics. Define a structure named **student** with the following members:

```
char name[50]: /* To store the student's name */
int rollNo: /* To store a student's unique roll number (within this class), e.g.: 1, 2, 3, ... */
float marks[5]: /* To store an array of grades for five different subjects */
```

Use the vi editor to create your program and save it as **lab10.c**.

Program Functionality:

- (a) (5 points) Class Size Determination:
 - i) Prompt the user to enter the total number of students in the class (≤ 50 students)
- (b) (10 points) Student Data Input:
 - i) Loop through the number of students.
 - ii) For each student, prompt the user to enter their name and roll number.
 - iii) Then, prompt the user to enter grades for each of the five subjects.
 - iv) Make sure the grades (marks) are checked in your code (e.g., $0 \leq \text{grade} \leq 100$).
- (c) (15 points) Grade Calculations:
 - i) For each student, calculate the total grade (marks) by summing up their individual subject grades.
 - ii) Calculate the average grade for the student by dividing the total grade by the number of subjects (e.g. 5).
- (d) (20 points) Output and Statistics:
 - i) Display a formatted table showing **name**, **rollNo**, **marks**, **total_marks**, and **average_marks** for students.
 - ii) Calculate and display the class average (average of all students' average grade).
 - iii) Identify the student with the highest average grade and the lowest average grade.

2. Now, do the following additional tasks:

- (a) (05 points) Make sure you explained your code to the TA or give enough documentation in your submission.
- (b) (03 points) Start recording your session using the `script` utility.
- (c) (03 points) Show the contents of `lab10.c` using the `cat` command.
- (d) (03 points) Compile `lab10.c` with required flags for the object file name [`use -o`] and C version [`-std=c99`].
- (e) (03 points) Run your program using appropriate command.
- (f) (03 points) Finish your recording (use the `exit` command).

Deliverables

For today's lab, clean the text file (`.txt`) you recorded during your terminal session, if there are unwanted control characters. In other words, make it as you observed during your terminal session. Please name your text file as **last-name_firstname_lab10.txt**. You will need to submit the text file (terminal session record) and your C file (`lab10.c`) to the **Lab 10** dropbox in iCollege.

Broader Grading Criteria

- If no C (`.c`) file is submitted (regardless if `.txt` file submitted or not), a student will receive only 40% for attendance. Submission will not be graded.
- If C file is given but no `.txt` file (terminal session) is given, a submission will receive maximum 70% (will vary between 40% to 70% based on the correctness of the C program).
- If a `.txt` file is given along with the `.c` file, but the `.txt` file is not clean and not comprehensible to the TA, a submission will receive maximum 80% (will vary between 40% to 80% based on the correctness of the C program).
- If both clean `.txt` file and the `.c` file are given, your submission will be normally evaluated based on the tasks and the corresponding point distributions.
- **Screenshots will not satisfy the requirements for code and/or the `.txt` files submission.**
- There should be compatibility between lab quiz performance and problem-solving (programming) performance. Otherwise, you may be called for an interview with Lab TA.