BIRZEIT UNIVERSITY

Computer Science Department

COMP133 (Second Semester 2020/2021)

Final Project ***Due Date: Saturday(05/6/2021) 11:59 PM on Ritaj Only***

Notes:

1. Submit your project (main.c File) through Ritaj only by replying on a message called Project solution .

2. The *project* should be submitted on the due date, late submission will NOT be accepted for any reason.

3. The *project* is an individual effort. Copying the project will be treated as a cheating attempt, which may lead to fail the course.

(Late projects will not be accepted for any reason)

4. Yoy are not allowed to use structure

Assume that a file contains *students' ids, full names, and their scores (Assignments grade, quizzes grade, Midterm grade, Practical exam grade, and final exam grade) (each column is separated by $).* You are required to write a C program to do the following:

- Using the concept of parallel arrays create records for students with above attributes (id, full name, score).(you are not allowed to use structure)
- Ask the user to enter the input file name and read it (suppose that, there are different files you could read data from). Read the data from the file and store it in record for students, which has IDs, Names, and Scores. The IDs should be declared as integers, the Names as a two-dimensional array of characters and the Scores as doubles. Assume that the maximum length of full name of any student is 50 characters. Also, you may assume that there will be No more than a 1000 student records in the file.
- Calculate the final grade as the flowing:

Grade= (Assignment)*15%+(Quizzes) *15%+(Midterm exam) *25%+(Practical Exam) *10%+(Final) *35% Assuming that data in files are arranged in same order of the above equation with respect to grades

Hint: read form file, calculate the final score, and store it in the record before going to the next step.

- Display the following menu to the user and read the entered choice:

1) Sort data in **ascending** order according to students' IDs and then display it.
2) Sort data in **ascending** order according to students' names and then display it.
3) Sort data in **descending** order according to students' scores and then display it.

Note: After running any of the above menus items, ask the user if he/she would like to save the current result, if so, prompt user to enter file name.

4) Ask the user to enter a student ID and display his score
5) Ask the user to enter a student name and display his score
6) Exit the program

- The program should keep displaying the menu until the user selects to exit from the program.

Implement each of the first five menu options as a separate function.

The attached file “data.txt” is for test.