CS 100 Project One – Fall 2018

Project Overview: In this project, you are asked to write a program to compute the semester weighted average for a student in CS100. Assume that CS100 has 10 labs, 10 quizzes (not 11 because we have not talked about how to drop the lowest score yet), 10 textbook exercises, 6 projects and 4 exams. The details of each evaluation item are spelled out in the following table.

Category	Percentage	Score Type	Score Range
10 labs	1% for each lab	integer	0-100
10 quizzes	1% for each quiz	real number	0-10
10 textbook exercises	1% for each exercise	real number	0-100
6 projects	2% for project 1 4% each for projects 2-3 5% each for projects 4-6	integer	0-100
4 exams	10% each for exams 1-3 15% for the final	real number	0-100

Your program will read a total of 40 values (some are integers and the others are real numbers) from the user, and use the grading scale specified in the above table to compute and print out the semester weighted average (a real number). The following shows a sample execution of the program, with the program prompts in blue and the user input in red. You must ask for these 40 values in this exact order.

```
Enter 10 lab scores (integers): 80 100 100 100 100 100 100 100 100 90
Enter 10 quiz scores (real numbers): 7.5 9 9 9 9 9 9 9 9 8.5
Enter 10 textbook exercise scores (real numbers): 90.5 100 100 100 100 100
100 100 100 89.5
Enter 6 project scores (integers): 70 80 80 80 80 100
Enter 4 exam scores (real numbers): 80 72.5 90 100
```

Your program should generate the following output line for the input above.

```
Your semester weighted average is 88.350000
```

Input redirection: the *scanf* function will get input from standard input, usually the keyboard. For ease of testing, you can use *vim* to create a data file, say **case1.dat**, and insert the following 40 values into the file.

```
80 100 100 100 100 100 100 100 100 90
7.5 9 9 9 9 9 9 9 9 8.5
90.5 100 100 100 100 100 100 100 100 89.5
70 80 80 80 80 100
80 72.5 90 100
```

Once you have the file **case1.dat** containing the 40 values, you can test the program using

```
./a.out < case1.dat
```

In this way, the program will read the values from the file **case1.dat** instead of the keyboard, which saves you from tedious data entry.

You need to create more test cases to test your program. To verify if your program works correctly with a test case, you can post the test case and its result to Piazza and ask if others agree with your result. **However, posting any part of C code from the project on Piazza is prohibited.**

What You Need To Do

- Create a directory named **`project1`** on your machine. In that directory, create a file named **`grade.c`**
- In **`grade.c`**, write the code needed to solve the problem stated above. Make sure that your program:
  - Has a header block of comments that includes your name and a brief overview of the program.
  - Reads 40 values in the specified order. You may assume that all input for this program will be valid (of correct type and within the specified range).
  - Computes the semester weighted average using the specified grading scale and print the result.
- When you are ready to submit your project, compress your **`project1`** directory into a single (compressed) zip file, **`project1.zip`**. See the **Basics** document on Blackboard if you don't remember how to do it.
- Once you have a compressed zip file named **`project1.zip`**, submit that file to Blackboard.

Project 1 is due at 5:00pm on Friday, August 7. Late projects are not accepted.

A project shall be completed individually, with no sharing of code or solutions.
All submissions will go through MOSS (Measure Of Software Similarity) for similarity check.
The University of Alabama's Code of Academic Conduct will be rigorously enforced.