CS 100 Project Four – Fall 2018

Project Overview: In this project, you will implement a set of functions that can be used to retrieve information from a CS100 grade book stored in a CSV (comma-separated values) file. The CSV file consists of 47 columns as shown below. The first two columns are always the first name and the last name. The order of the remaining 45 columns are unknown, and they can be in any order.

- First Name and Last Name.
- L1 through L10 for 10 labs.
- B1 through B10 for 10 textbook exercises.
- Q1 through Q11 for 11 quizzes.
- P1 through P6 for 6 projects.
- E1 through E8 for 8 exams (tracing and coding are considered as two different exams).

The first row (or line) of the CSV file is the header, i.e. column names. Each following row represents a student. The first name and the last name in each row will not be blank or contain a space. Each score under the 45 headings will be either blank (with no score) or a real number. A blank score is considered as 0 in some calculations and is ignored in the others.

You are asked to implement the following five functions in the `functions.c` file.

- `double getMin(char csvfile[], char column[]);` Given a CSV file, return the minimum score of the specified column. The blank cells are excluded from the calculation.
- `double getMax(char csvfile[], char column[]);` Given a CSV file, return the maximum score of the specified column. The blank cells are excluded from the calculation.
- `double getAvg(char csvfile[], char column[]);` Given a CSV file, return the average score of the specified column. The blank cells are excluded from the calculation.
- `int getCount(char csvfile[], char column[], double threshold);` Given a CSV file, return the number of students with their specified score >= threshold. The blank cells are excluded from the calculation.
- `double getGrade(char csvfile[], char first[], char last[]);` Given a CSV file, return the weighted average of the specified student. A blank cell is viewed as 0 in the calculation. The weight percentage for each column is specified below. The lowest quiz score will be dropped from the calculation.

Column	Weight Percentage	Score Range
L1 through L10	1% for each lab	0-100
Q1 through Q11	1% for each quiz.	0-10
B1 through B10	1% for each exercise	0-100
P1 through P6	2% for P1 4% each for P2, P3 5% each for P4, P5, P6	0-100
E1 through E6	5% for each exam	0-50
E7, E8	7.5% for each exam	0-75

When implementing the above functions, you can assume `csvfile` is a valid CSV file and it can always be opened for reading, and `column` is a valid column name. You can also assume there is at least one non-blank score in each column. In addition, you are allowed to add additional helper functions in the `functions.c` file.

We recommend you use the `fgets` function to read a line from the CSV file, and use the `strsep` function to extract each field from the line. You can assume that each line does not exceed 300 characters in length and each field does not exceed 30 characters in length. It is not recommended to use the `strtok` function to extract each field from a CSV file because a cell could be blank in such a file. You shall use the `strcasecmp` function to perform case-insensitive comparison between two names. To compile this project, use the following command. You can download **main.c** from Blackboard and you shall not modify anything in **main.c**.

```
gcc -Wall -std=gnu99 main.c functions.c
```

Testing: a CSV file named case1.csv can be downloaded from Blackboard for testing. You can test the program using the following command, and a sample executions of the program is shown at the end of this document.

```
./a.out case1.csv
```

To verify whether you have implemented a function correctly, you can post the corresponding test commands and their results to Piazza and ask whether others agree with your results. **However, posting any part of C code from the project on Piazza is prohibited.** You can also use Microsoft Excel to load the CSV file and enter the formula that is equivalent to your test command to see whether they produced the same result.

What You Need To Do

- Create a directory named **project4** on your machine. Download **main.c** and **case1.csv** to that directory, and create a file named **functions.c** under that directory.
- In **functions.c**, implement the five functions as specified above, make sure there is a header block of comments that includes your name and a brief overview of your task.
- When you are ready to submit your project, compress your **project4** directory into a single (compressed) zip file, **project4.zip**. Make sure **project4.zip** contains the `project4` directory as well as `functions.c` under it. (`main.c` is not required.)
- Once you have a compressed zip file named **project4.zip**, submit that file to Blackboard.

Project 4 is due at 5:00pm on Friday, October 19. Late projects are not accepted.

**This document including its associated files is for your own personal use only.
You may not post this document or a portion of this document to a site
such as chegg.com without prior written authorization.**

**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.**

A sample execution of the program

```
./a.out case1.csv

Enter a command: min Q1
min(Q1)=6.5

Enter a command: min L3
min(L3)=17

Enter a command: max P1
max(P1)=100

Enter a command: max Q9
max(Q9)=10

Enter a command: avg E4
avg(E4)=46.413

Enter a command: avg Q7
avg(Q7)=9.35556

Enter a command: count B5 100
count(B5>=100)=36

Enter a command: count B5 90
count(B5>=90)=45

Enter a command: count Q10 10
count(Q10>=10)=17

Enter a command: count Q10 8
count(Q10>=8)=20

Enter a command: grade Lloyd Gibbon
grade(Lloyd Gibbon)=98.31

Enter a command: grade Stormy Beaufort
grade(Stormy Beaufort)=78.688

Enter a command: grade stormy beaufort
grade(stormy beaufort)=78.688

Enter a command: grade John Smith
No student named John Smith

Enter a command: quit
```