CS102: Introduction to Computer Science

Summer 2014

Program #2

Consider a course similar to CS 102 (the programming section only), but containing only 15 students. The workload for the course consists of four programs and one final exam. The grading breakdown is as follow:

- Program #1: 10%
- Program #2: 15%
- Program #3: 15%
- Program #4: 20%
- Final Exam: 40%

You are going to write a program that reads in all grades of all students and then outputs a grid displaying the grades, including the weighted average for each student.

Since we have not learned about how to read information from a text file yet, your program should just read from standard input. The input will consist of 15 lines (one per student), and each line will contain the five grades for one student (for programs 1 through 4 and then for the final exam). Grades will be separated by whitespace. You can assume that all input values will be integers from 0 to 100 (i.e., valid grades for assignments); there will be no text or other invalid characters in the input. Do not have the program prompt the user to enter grades; just read them in. When I test the programs, I will redirect standard input to come from a file.

The output should be a nicely formatted chart that looks exactly like the example shown on the next page if the input data is the same. The values in all columns should be right justified. The values in the final column should contain exactly two digits after the decimal point. All columns should be exactly eight characters wide. The column headers should match those in the example. Every line should end with a single Linux-style newline character.

Your homework will be graded out of 100 points with the following breakdown:

- **Correctness**: You should follow all instructions exactly as stated above. This is an individual assignment (i.e., you should not collaborate with anyone else). **75 points**.
- **Elegance and Efficiency**: You should use the concepts we have learned in class to write your program in a simple, elegant manner. I will give some advice, and it is OK if you don't take it, but if I consider your solution less elegant, you will lose points. **15 points**.
- **Format**: Your program should use proper indentation and other spacing which makes the code readable and easy to understand. **5 points**.
- **Comments**: You should include one comment at the top of your program indicating your full name and (briefly) what the program does; one comment above each function explainng what the function does (optionally you can describe the parameters and return value, if any); and optionally one or more short comments within the code explaining how it works. **5 points**.

Submitting assignments: Email me your code (to **CarlSable.Cooper@gmail.com**) as an attachment. Do NOT attach the executable (only send the source file). Please state in your e-mail which environment you used to develop the code (e.g., Cygwin, Quincy, Ubuntu, etc.). This program is due the night of Tuesday, July 29, before midnight.

CS102: Introduction to Computer Science

Summer 2014

Program #2

I am going to redirect standard input to come from a file when I run your programs. Hypothetically, let's say the content of such a file is as follows:

```
90 80 85 87 74
75 81 82 73 80
60 50 70 60 80
90 90 90 90 90
95 90 98 99 74
88 82 94 91 88
98 99 100 98 95
72 99 75 88 81
100 100 100 90 100
64 71 88 75 80
90 85 0 77 60
90 100 95 80 75
72 60 55 70 65
85 98 92 96 89
70 75 68 79 82
```

Then the output of your program should look exactly like this:

```
  STUD # PROG #1 PROG #2 PROG #3 PROG #4     EXAM AVERAGE
       1      90      80      85      87      74   80.75
       2      75      81      82      73      80   78.55
       3      60      50      70      60      80   68.00
       4      90      90      90      90      90   90.00
       5      95      90      98      99      74   87.10
       6      88      82      94      91      88   88.60
       7      98      99     100      98      95   97.25
       8      72      99      75      88      81   83.30
       9     100     100     100      90     100   98.00
      10      64      71      88      75      80   77.25
      11      90      85       0      77      60   61.15
      12      90     100      95      80      75   84.25
      13      72      60      55      70      65   64.45
      14      85      98      92      96      89   91.80
      15      70      75      68      79      82   77.05
```

I will also redirect standard output to a file, and I will compare that output to my own output using the Linux / Cygwin "diff" command. If there are any differences, you will lose points.

I suggest storing the input in a two-dimensional array of integers. You may make that a global variable if you want, but I did not. I suggest breaking down the program into two functions in addition to main; one to read the values from standard input, and another to display the chart. You may want to store the number of students and number of grades per students as constants. I used a one-dimensional array of floats to store the percentages that each assignment is worth.