CPE 100 Introduction to Computer Programming
International Sections August 2020
Laboratory Exercise 7

Objective

This lab is intended to give you practice creating a program that uses functions with array-based and pointer-based output arguments.

Instructions

Write a program called **studentStats.c.** First the program should ask how many quizzes there were in the course (maximum 10). Then the program should loop, getting and storing the student name and his or her quiz grades. For each student, it should calculate and print the average (mean) and standard deviation of that student's quizzes, before continuing to the next student. The program should exit when the user enters “DONE” for the student name.

The standard deviation measures how much a set of data values vary among themselves. A larger standard deviation—often abbreviated "SD"—means that there's more variability in the data values. You can find the formula for SD below.

Here is a sample run of this program.

```
How many quizzes in the course (max is 10)? 4
Enter name of student 1: John
        Quiz grade 1 for John:  10
        Quiz grade 2 for John:   8
        Quiz grade 3 for John:   9
        Quiz grade 4 for John:   10

       John's average grade is 9.25
        The standard deviation is 0.9574

Enter name of student 2:  Helen
        Quiz grade 1 for Helen:   0
        Quiz grade 2 for Helen:   8
        Quiz grade 3 for Helen:   2
        Quiz grade 4 for Helen:   10

       Helen's average grade is 5.00
        The standard deviation is 4.7609

Enter name for student 3: DONE
Goodbye!
```

Put the code to get the quiz grades in a function called **getQuizGrades**. This function should have three arguments: ***a string*** (input argument holding the student's name), ***an array of integers*** (output argument for storing the quizzes) and ***a count*** (input argument holding number of quizzes, which will be less than or equal to 10 and the same for every student). This function should be a *void* function (no return value).

Put the code to calculate the mean and standard deviation for one student's grades in a function called **calculateStats**. This function should have three arguments: the ***array of grades*** (an input argument), the ***count*** (an input argument), and a ***pointer to a double*** (an output argument), where you will return the standard deviation. You should ***return the average*** as the function value.

To calculate the standard deviation, use the following formula:

$$SD = \sqrt{\frac{\sum (x - \bar{x})^2}{N - 1}}$$

The sigma character (**Σ**) means "summation". The **x** with a bar on the top is the average value over N measurements (quizzes in this case). In words, this formula means: *For each quiz, subtract the average of all the N quizzes from this quiz score, and square the result. Add that to a total. When you have summed up the squared differences for all quiz scores, divide the total of the squared differences by the number of quizzes minus 1 (N-1). Take the square root of the result.*

Sample Function Declarations

```
/* Get 'count' quiz grades for the student with the name 'studentName'
 * and store the results in the 'grades' array.
 */
void getQuizGrades(char studentName[], int grades[], int count);

/* Calculate the mean (average) and standard deviation for a set of 'count' grades
 * stored in the 'grades' array.
 * Return the calculated standard deviation in the address pointed to by 'pSD'.
 * Return the average as the function value.
 */
double calculateGrades(int grades[], int count, double* pSD);
```

Hints and Recommendations

1. Start by writing the loop to get the student names. Create stubs for the **getQuizGrades()** and **calculateStats()** functions. Compile and test your program with the stub functions. Then implement and test **getQuizGrades()**. Finally, write the code for **calculateStats()**.
2. Since you only need to hold one name at a time, not all the names, you do **not** need a two-dimensional array of characters. You can re-use the same one dimensional array (string) for the name again and again.
3. Similarly, since you will be processing each student's grades, then moving to the next student, you only need a *single dimensional array of grades* (with a dimension of 10 according to the instructions).
4. In **calculateStats()**, you will need to loop through the quiz grades twice. In the first loop, you will calculate the average. In the second, you will calculate the SD, which needs the average. (There are ways to do this with only one loop, but I don't want to confuse you.)
5. Don't worry in this program about checking the values the user types to make sure they are correct. Focus on the problem of writing functions with output arguments. (Unless of course you finish early and you are bored...!)
6. Note that quiz grades are integers. However, both the mean and standard deviation should be floating point values (doubles). To make sure your results are correct, you should store the sum of values (for the mean calculation) and the sum of squared deviations (for the SD) in *double* variables, before you do any division or take the square root.
7. If you are using Linux, you may need to compile and link like this:

   ```
   gcc -o studentStats studentStats.c -lm
   ```

   The “-lm” links in the math library which contains the `sqrt()` function. You also need to include `math.h`.