Cp Sc 1110 Program 4
Due: Tuesday Oct. 28th, 11:59 pm

The purpose of this assignment is to give you practice in using partially filled arrays and functions.

Problem Statement

Orange Labs is testing more LED bulbs. Suspect bulbs are those whose readings are outside of a range of a percent factor less than average to a percent factor greater than average. They have created a text file that first contains a decimal number percentage that is to be used to determine if a reading is outside the range. 7.5% would be entered as 7.5. This number is followed by a maximum of 200 integer bulb readings. Write a C program for them with the following specifications:

1. General:
   - 1.1. Save your code as prog4.c.
   - 1.2. For all functions except main place prototypes in the global area before main and the definition after main.
   - 1.3. Use the functions defined below and closely follow the sample output below with no blank lines.
   - 1.4. The program will read a single file using standard input redirection with no prompting.
2. *main*
   - 2.1. Procedure:
     - 2.1.1. Define a named constant for the maximum readings.
     - 2.1.2. Define an array to hold the bulb readings using the named constant and then other needed variables.
     - 2.1.3. Read and store the percent factor.
     - 2.1.4. Use *loadBulbs* to load the readings into the array.
     - 2.1.5. Use *printBulbs* to print the readings.
     - 2.1.6. Use *minReading*, *maxReading*, *averageReading*, and *suspectCount* to gather statistics from the readings array.
     - 2.1.7. Print the statistics heading and statistics as shown with percentages to one decimal place.
3. *loadBulbs*
   - 3.1. Parameters: (1) Integer readings array and (2) integer maximum readings
   - 3.2. Return: Number of readings placed into array
   - 3.3. Procedure:
     - 3.3.1. Read readings from the file and place them sequentially into the array.
     - 3.3.2. If there are more readings in the file than can be placed in the array, display an error message indicating the maximum readings and use the exit function from stdlib.h to exit the program with a return value of 1.
4. *printBulbs*
   - 4.1. Parameters: (1) Integer readings array and (2) integer number of readings placed in array
   - 4.2. Return: None
   - 4.3. Procedure:
     - 4.3.1. Print the readings heading as shown.
     - 4.3.2. Print the readings 10 per line.
5. *minReading, maxReading, and averageReading*
   - 5.1. Parameters: (1) Integer readings array and (2) integer number of readings placed in array
   - 5.2. Return: Integer statistic as indicated by name, average truncated to integer
6. *suspectCount*
   - 6.1. Parameters: (1) Integer readings array, (2) integer number of readings placed in array, (3) integer average reading, and (4) decimal percentage
   - 6.2. Return: Count of readings that are outside average +- percent of average as a decimal numbers

On a SOC machine, you can copy the sample test file to your current directory by entering

```
cp /home/psterli/public_html/prog/sample4.txt ./
```

You can view the file in a browser and save on your computer from here:

```
http://www.cs.clemson.edu/~psterli/prog/sample4.txt
```

To run your program with the sample file:

```
./a.out < sample4.txt
```

You should also make up your own data file and test for any additional conditions that should be tested such as when the number of readings is evenly divisible by 10.

Sample Input

```
5.9
753 763 754 811 844 792 754 758 781 802 841 755 831 798 796
827 849 813 801 756 790 750 782 761 846 800 846 798 769 830
776 772 809 848 799 769 789 803 844 820 821 834 792 767 781
804 844 846
```

Sample Output (Note: No blank lines)

```
========================================
48 Readings
========================================
753 763 754 811 844 792 754 758 781 802
841 755 831 798 796 827 849 813 801 756
790 750 782 761 846 800 846 798 769 830
776 772 809 848 799 769 789 803 844 820
821 834 792 767 781 804 844 846
========================================
Statistics
========================================
Minimum: 750
Maximum: 849
Range: 99
Average: 799
Outside 5.9%: 3 = 6.2%
========================================
```

Requirements

1. Your program must adhere to this assignment's problem statement requirements and the general programming assignment requirements. Violations will lead to deductions. General requirements: http://www.cs.clemson.edu/course/cpsc111/lect/requirements.pdf

2. Submit prog4.c electronically using Webhandin:
    - 2.1. If your program is stored on the SOC system and you are submitting from a non-SOC machine: You must first transfer your C program to your local machine. If your machine is a Windows machine you will use SSH Secure Shell File Transfer Client, MobaXterm Sftp download, or a comparable product. On a Mac OS or Linux machine, you will use sftp. See the appropriate documentation for instructions.
    - 2.2. Go to http://handin.cs.clemson.edu to open Webhandin.
    - 2.3. If requested sign in with your username and password.
    - 2.4. On the My Courses page, click on the link containing `CPSC 1110`.
    - 2.5. On the Assignment page select `prog4`.
    - 2.6. The place where you put your files is referred to as a "bucket". When browsing your bucket you'll be shown some information about the assignment and a list of files which you have submitted. From this page, you are able to submit a file or delete a previously submitted one. To submit a file, click the button toward the bottom of the page. To delete one, click the delete button next to the file. If you've submitted a file, there will be a time noted at the top of the page which marks when Webhandin received your submission.
    - 2.7. Submit `prog4.c`. After submitting, click on the filename link. Your program should be displayed.