Spring 2017 CMSC 140 Programming Project 6: Rainfall Statistics

Concepts tested in this project

- Reading data from a file into arrays
- Processing parallel arrays
- Writing array data to files
- Passing arrays to functions
- Processing arrays using functions

Project Description

Write a program that will calculate and display on a file the total rainfall for the year, the average monthly rainfall, the months with the highest and lowest amounts, and the months with rainfalls of 4 inches or more.

Project Specifications

Input for this project has two sources. The rainfalls for each month beginning with January and ending with December are read, in order, from an input file. The user must enter: the names of the input and output files.

Output also has two sources. The program title and the programmer name should appear on both the console and the file. If the input file does not open, an error message should appear on the console. The message "Processing complete" should appear on the console just before the main program ends (make it the last statement before "return;"). The file output shows the months and associated rainfalls as well as the calculated and summary results.

Processing Requirements

Use two parallel arrays in the main function for this project. The names of the months should use an initialization list and the rainfalls will be stored in an array of doubles. The arrays will be passed as parameters. (NO GLOBAL VARIABLES--a global constant of 12 is fine.) Assume that the file contains exactly 12 lines of input and is constructed correctly.   Output to the file will be handled by the main function. Assume that a rainfall of exactly 4 inches will be stored as 4.0 (not 3.9999...).

Create functions for the following:

1. Read the rainfalls: Prompt for and get the input file name and try to open the file. If the file opens, read the rainfalls from the file, store them into the array, and return 0. If the file does not open, print an error message on the console and return -1. If the file is not found, do NOT process the rainfalls or attempt to write to the output file.

2. Calculate and return the total of rainfall array elements.   The rainfall array should be a parameter.

3. Find and return the subscript of the highest rainfall amount. The rainfall array should be a parameter.

4. Find and return the subscript of the lowest rainfall amount. The rainfall array should be a parameter.

Sample File Input:

```
3.80
5.20
4.70
6.30
3.90
0.91
0.91
2.16
3.40
2
2.00
3.70
```

Sample Screen Output:

```
C:\windows\system32\cmd.exe

          YEARLY RAINFALL CALCULATIONS

Please enter the name of the output file e:\\out6.txt
Please enter the name of the input file e:\\in6s17.txt

Programmer: insert your name here
Processing complete
Press any key to continue . . .
```

Sample File Output:

```
          YEARLY RAINFALL CALCULATIONS

     Month                 Rainfall(in inches)

January                         3.80
February                        5.20
March                           4.70
April                           6.30
May                             3.90
June                            0.91
July                            0.91
August                          2.16
September                       3.40
October                         2.00
November                        2.00
December                        3.70


The total rainfall for the year is 38.98 inches.
The average monthly rainfall is 3.25 inches.
The highest rainfall of 6.30 occurred in April
The lowest rainfall of 0.91 occurred in June
The lowest rainfall of 0.91 occurred in July

Rainfall of 4 inches or more occurred in February.

Rainfall of 4 inches or more occurred in March.

Rainfall of 4 inches or more occurred in April.

Programmer: insert your name here
```

Project 6 Submission requirements:
Deliverables

1. A zip file named *LastNameFirstName_Project6.zip* containing:
   a. A Word document that includes:
      - Screen shots showing sample test data. (at least 2 of each--screen and file--different from those given.
      - A flowchart showing your main function logic. Use a striped rectangle containing the name of the function for each function call.
   b. Your source code file called rainfallStatistics.cpp
   c. Output file
   d. Test plan (table) with at least 2 different data files

2. A zip file named LastNameFirstName_Project6_MOSS.zip containing ONLY rainfallStatistics.cpp

**Include the following header at the beginning of your program:

```
/*
 * Class: CMSC140
 * Instructor: _____
 * Description: (Give a brief description for Project 6)
 * Due date: _____
 * Platform/Compiler: (Windows 7, Microsoft Visual Studio 2013 for example)
 * I pledge that I have completed the programming assignment independently.
   I have not copied the code from a student or any source.
   I have not given my code to any student.
   Print your Name here: _____

        Pseudocode for algorithm design (show the logic in all functions)
*/
```

Grading Criteria for Project 6

The following are components on which the projects will be graded. If program does not compile, project will get grade "0". Contact your instructor prior to the project submission due date, if you have compilation issues.

Attributes	Value (points)
Functionality (If project does not compile, project will get grade "0")	Total 100
Displays the console input and report file output appropriately formatted	30
Calculates and displays the total and required items correctly	30
Program executes correctly (produces expected output)	15
Meets all requirements	15
Overall Look-and-Feel	10
Total	Total 100 points

Project General Requirements (points will be deducted)

Attributes	Value(points)
Programming Style and proper naming convention: (see coding standards)	(-20 pts maximum)
Curt or unclear variable names	-5
Long variable names should use camel case or underscores to separate words	-5
Comments and internal notes	
Sparse and inadequate comments. (Missing with blocks of code, before functions, or with variable definitions)	-5
File header is not included (project description, name, etc.)	-5
Essentially no comments	-10
Indentation and white spaces should be a visual aid to understanding code structure	
Indenting is mostly okay, but sometimes inconsistent.	-5
No indenting, or very inconsistent indenting that is a barrier to understanding the code	-10
Lack of white space separating variables and operators. Lack of white spaces separating functions and major code blocks	-5
Test Plan	(-20 pts maximum)
Missing Entirely	-20
Cursory or inadequate testing (at least 2 different data files in table form)	-10
Missing Required Items	(-20 pts maximum)
Pseudocode (with code), Flowchart missing	-20
Screen shots cursory or incomplete (at least 2 different from examples)	-5
Screen shots completely missing	-10
An output file not included as a separate file	-5
Missing total, high and low rainfall months, months with 4 inches or more with amounts or not shown correctly	-5
Incorrect parameters passed to functions or parameters missing	-5
Awkward Code Internal Structure	(-10 pts maximum)
Error message is missing when input file is missing	-5
Tries to process data when input file is missing	-5
Poor structured programming: more than one "return" statement at end of each function ("break" and "continue" used)	-5 each
Other poor coding practices not mentioned	-5