Using Files

Due at the start of your next lab period

Objectives

Read and write ASCII data files
Use a command-line argument
Write a C program that uses multiple files and functions
Write and use header files
Use the math library

Requirements

Write a C program that computes simple statistics from a file of ASCII numbers.

Your program must:

1) Check for the correct number of command-line arguments
2) Get the name of the input file from the command line and try to open that file
3) Check that the file was opened successfully
4) Read all of the values in that file while updating the required variables
5) NOT use an array for the data
6) Check that there was at least two values read from the file
7) Display the result to the console
8) Write the same result to an output file where the name of that output file is Result_xxx, where "xxx" is the name of the input file. You must check that this file was opened successfully.
9) Close files when you are done with them
10) Consist of three files: two source files (statsMain.c, statistics.c) and a header file (statistics.h).
    a) The following two functions must be in statistics.c

```
/* Compute mean from sum and count */
double mean(const double sum, const int count);
/* Compute sample standard deviation from sum, sum of squares, and
count */
double ssdev(const double sum, const double sumsq, const int count);
```

    b) The following function must be in statsMain.c and be located after the main()

```
void printStats (FILE *dest, const int count, const double mean,
const double ssdev);
```

    This function must be used for printing all results
    c) All files are opened and closed in main().
11) Use `#ifndef` to prevent the possibility of multiple inclusions of the header file

Hint

The formula for computing the sample standard deviation is

$$\sigma_S = \sqrt{\frac{N\left(\sum_{i=1}^{N} x_i^2\right) - \left(\sum_{i=1}^{N} x_i\right)^2}{N(N-1)}}$$

Unlike what you probably learned in statistics, this formula does not require the use of arrays.

Evaluation

Demonstrate your working, documented program to the instructor	7/10
Do all of the required error checking, file closures, etc.	3/10

Sample Runs

```
dale$ ./statsMain numbers.txt
6 Values, Mean = 4.33383, Sample Standard Deviation = 7.15378

dale$ ./statsMain numbers111.txt
111 Values, Mean = 52.4038, Sample Standard Deviation = 29.0565
```

And the file Result_numbers.txt contains

```
6 Values, Mean = 4.33383, Sample Standard Deviation = 7.15378
```

Examples of the error checking

```
dale$ ./statsMain someNonExistentFile.txt
./statsMain: Unable to open input file ”someNonExistentFile.txt”

dale$ ./statsMain
Usage: ./statsMain dataFileName
```