cs402 - Fall 2023

Systems Programming Lab #3 - C Pointers

---

1. Goal

- Gain experience using pointers and dynamic memory allocation (malloc) in C.
- Practice using gdb and valgrind to debug C memory bugs.

2. Lab Description

For this lab, you will implement a program that reads data from a file and computes some basic statistical analysis measures for the data.

The program should work without any changes for data sets of any size, i.e. it would work for 10 data values or for 10 million without re-compilation.

Name your program `basicstats`

2.1 Program Start-up

Your program will take one command line argument, which is the name of the input file containing the data to be analyzed.

Two sample files are provided, small.txt and large.txt.

```
$  ./basicstats small.txt
  Results:
  --------
  Num values:            12
        mean:        85.776
      median:        67.470
      stddev:        90.380
  Unused array capacity: 8

$  ./basicstats large.txt
  Results:
  --------
  Num values:            30
        mean:      2035.600
      median:      1956.000
      stddev:      1496.153
  Unused array capacity: 10
```

2.2 Statistic Functions

mean: the average of values in the set. For example, if the set is S={5, 6, 4, 2, 7}, then the average is 4.8

median: the middle value in the set of values. For the same set S the median value is 5 (2 and 4 are smaller and 6 and 7 are larger). If the set has an even number of values, then you average the two values that are left and right of center.

standard deviation: you are going to calculate the *population standard deviation* by using the following formula:

```
stddev = sqrt((sum((xi - mean)^2))/N);
```

where the sum() goess from 1 to N, xi is the i-th element, N is the number of elements in the data set, and sqrt() is the square root function.

NOTE: To use sqrt() you'll need to link your code with the math library.

2.3 File Format

The input file format consists of several lines of ASCII text. Each row in the file is one datum, as a regular decimal number, e.g. 123.45, -6.78901, etc.

3. Requirements

Compute the mean (average), mode, geometric mean, harmonic mean, median (the middle value), and the standard deviation of the set of values and print them out.

NOTE: You may not use math libraries in your code.

Print out the results, plus information about the number of values in the data set and the amount of unused capacity in the array storing the values.

The array of values must be dynamically allocated on the heap by calling malloc(). You should start out allocating an array of 20 float values. As you read in values into the current array, if you run out of capacity:

- Call malloc to allocate space for a new array that is twice the size of the current full one.
- Copy values from the old full array to the new array (and make the new array the current one).
- Free the space allocated by the old array by calling free.

NOTE: there are other ways to do the memory allocation/re-allocation than described above, however for this assignment you have to use the method described above.

When all of the data values have been read in from the file, the function should return the filled, dynamically allocated, array to the caller (the function's return type is float *). The size and capacity of the array should be "passed" back to the caller via the pointer parameters.

For full credit, your submission should meet the following requirements:

- Your code should be commented, modular, robust, and use meaningful variable and function names. This includes having a top-level comment describing your program and listing your name and the date. In addition, every function should include a brief description of its behavior.
- You should not use any global variables for this assignment.
- It should be evident that you applied top-down design when constructing your submission (e.g., there are multiple functions, each with a specific, documented role).
- You should not assume that we will test your code with the sample input files that have been provided.

4. Tips

- Try getting your code to work without the re-allocation and copying part first (for fewer than 20 values). Once that works, then go back and get it to work for larger numbers of input values that require allocating new heap space, copying the old values to the new larger space, and freeing up the old space.
- Use doubles to store and compute the mean and the square root.

5. Submit your work

Here is what you need to deliver on the Blackboard:

- A link to your git repository such that we can clone it, build an executable and test it.

NOTE: please make sure your repository contains a README file that explains how to build an executable and execute it.

Last update: Aug 21, 2023	Virgil Bistriceanu	cs402	Computer Science

---