CMP\_SC 3050 Homework 1 Makeup

Due: 11:59:59 pm, 10/13/2015

This is an opportunity for you to make-up your grade in first homework. We shall take the best of the two scores from Homework 1 or the Makeup as your first score.

The first part of the homework specifies the exact problems that your submission should solve and the second part describes the constraints that you **must** follow (no exceptions). Absolutely no late work will be accepted.

**Specification:**

For this programming assignment, one compressed file containing the directory of all of the assignment files should be submitted to Blackboard prior to the due date. Please do **not** submit an executable. It is recommended that your assignment must is a buildable NetBeans project. If you want to not use NetBeans, please use a Makefile. A sample Makefile for Homework 1 is included in the content area for Homework 1. If you are having trouble making a Makefile, please contact one of the TAs and the TA shall help you make it. You are welcome to re-use the code for Homework 1 posted for parsing the input.

The assignment should:

* Take two filenames as input in the command line. The first file is going to be the input in the program and the second file is going to be the file in which you will write the output.
* The first file represents a list of numbers. The numbers can be positive or negative. A positive number is represented as a string of digits and a negative number is represented as a minus sign (-) followed immediately by a string of digits. Numbers shall be separated by whitespaces, newlines and tab-spaces. A sample input file can be found on Blackboard.
* If the input file has n numbers then the output file shall contain exactly n lines. Assuming that the numbers are read left to right and top to bottom from the input file, the i-th line of the output file should be the median of the first i numbers in the input file. That is, the 1st line will contain the median of the 1st number, 2nd line the median of the first 2 numbers, 3 line the median of 3rd number and so on… For the assignment, the median of a list of i numbers is defined as follows:
  + If i is odd, then the median is the number such that there are exactly (i-1)/2 less than the median in the list.
  + If i is even, then the median is the (i/2+1)th smallest numbers in the list.
* It is important that each line in the output file should itself consist of exactly one number. It is absolutely essential that you do not put any whitespaces or any other character in that line (except, of course, the end of line.)

**Constraints:**

The following is a list of constraints for this assignment; failure to adhere to any of these constrains will result in a loss of points or even a zero.

* The assignment can be completed in either C or C++.
* Built-in data structures and external libraries other than standard input/output functions may not be used for this assignment. If the program requires a stack, linked list, or any other structure, it is up to you to provide it. If you have any doubt on whether or not any technique you wish to use is acceptable, do not hesitate to ask.
* A moderate amount of error checking and resource management is required. Even if you do some error recovery, you must report errors in the input. Your application should ensure that each line from the input file is properly formatted, that the file is successfully opened, the file is successfully closed upon reading of the file and that all allocated space is de-allocated at the exit. You may re-use any data structure that is given in the posted solution of Homework 1
* For reporting input format error, please use the enum type provided in statistics\_error.h. Please do not alter the files statistics\_error.h .
* A moderate amount of formatting and documentation is required. Comments should be descriptive and used to illustrate the purpose and inner workings of an algorithm or function; they should not be used to annotate each line or self-evident logic.
* The assignment must be submitted via Blackboard. Please compress the folder before submitting using zip.
* The input size is unknown, and the input range is INT\_MIN to INT\_MAX. This means that your program should not make any assumptions about the range or size of the input.

**Timing Constraints**

A good algorithm for this homework should run in O(n log n). While we shall not be grading the exact time complexity of your solution, we shall be timing your solution and grading efficiency. As a guideline, on an input file consisting of million integers, your program should run under a minute of system time to receive full credit for efficiency.

**Grading**:

There are 10 points possible for this assignment. The grade breakdown is as follows:

* 1 points for error checking and resource management.
* 1 points for general programming style and adherence to the constraints.
* 5 points for correctly outputting the list of medians in the output file.
* 3 points for efficiency.

If the program fails to compile or crashes due to a runtime exception, a grade of zero will be assigned.