Daniel Beck and Diane Truong
Team Divide By Zero

Requirements for Project Multi-Threaded Sorter:

Functional Requirements
* The program should accept a single argument sent to it depicting how many threads to use.
* The argument should be a number indicating how many threads should be created to sort.
* The program should run through all sorting methods (8+) in four size increments.
* The program should split into a user specified number of concurrent threads to perform tests.
* Each individual thread should run through all and sizes for each method once.
* The first thread to complete all sorts should average the results of all but the last sorting method.
* The last thread to complete all sorts should average the final sorting algorithm and size.
* Once all threads complete, a report file should be created to store all average sorting results.

Sorting algorithms to be used:
Merge, Insertion, Quick, Radix, Selection, Bubble, Heap and Counting sorts (and others should we desire to add more)

Non-functional Requirements
* If no argument is given when running the program, the program could prompt for one.
* The program could provide an error message if the specified thread count is not a multiple of 4.
* The program could prompt the user to ask if it should do all tests or just a specific one.
* The program could ask to overwrite the output file if it exists, append it, or discard new results.
* The last thread to complete a test could output to screen a message indicating the test and size just completed.