**P4 Testing Document**

This was the by far the least troublesome programming assignment this semester. I had only one head-scratching issue, being the obligatory confusion during the transference of the code from CodeBlocks to Eustis. This time, it was returning the value of i in the bookCombo function after having iterated through the entirety of the array. In CodeBlocks, it worked fine. After some tweaks I discovered the fix was explicitly declaring i as int i = 0, whereas before I had simply done int i;. Then it worked fine. Eustis is very picky.

Since this program has us run the program multiple times with separate .in files for each test case, we were only asked to submit one pair of test case files with the assignment.

Scholar\_00 is the sample input #3 from the assignment page. The sample inputs were crucial to fine-tuning the program.

Scholar\_01 tests is 6 books, with a page limit of 89. The pages in each book are 1, 1, 1, 90, 1, and 1. This was set up to test having multiple of the same page counts and also having a page count greater than the page limit.

Scholar\_02 and scholar\_03 were created using a simple program, where globals modified the book count and page limit. Then a for loop creates a randomized page count for each book.

Scholar\_02, which was submitted with the project, was created to make a large book count that utilized the recursive function (and consequently superseding the initial insertion sort), while also being easily verified with a quick glance.

Scholar\_03 is the max case. 10^5 books and 10^14 page limit. Page counts for each book were randomized. I was blown away by how fast it compiled. Expected a memory leak on the first try.