Program 5
100 points

Program 5a: You do not need nor may you use arrays for this program. Write a program that merges the numbers in two files and writes the results to a third file. The program reads input from two different files and writes the output to a third file. Each input file contains a list of integers in ascending order, that is they start small and get bigger as they go. After the program is executed, the output file contains the numbers from the two input files in one longer list, also in ascending order. Your program should define a function with three arguments: one for each of the two input FILE pointers and one for the output FILE pointer. The input files should be named `numbers1.txt` and `numbers2.txt`, and the output file should be named `output.txt`.

Some test cases that you may want to consider include the following:

- What if one of the files contains no numbers?
- What if both of the files contain no numbers?
- What happens if one of the files has many more numbers than the other?
- What happens if there are several blank lines after the last number in both files?
- What happens if there are no blank lines, spaces, or characters of any sort after the last number in each file?

Name your source code file `program5a.cpp`.

Program 5b: Write a program to compute average grades for a course. The course records are in a single file and are organized according to the following format: Each line contains a student's first name, then one space, then the student's last name, then one space, then some number of quiz scores that, if they exist, are separated by one space. Each student will have zero to ten scores, and each score is an integer not greater than 100. Your program will read data from this file and write its output to a second file. The data in the output file will be the same as the data in the input file except that there will be one additional number at the end of each line: the average of the student's ten quiz scores.

The output file must be formatted such that first and last names appear together in a left justified column that is 20 characters wide. Each quiz score should be listed in a right justified column that is 4 characters wide, and the average should appear in its own right justified column that is 10 characters wide.

Note that if a student has fewer than 10 scores, the average is still the sum of the quiz scores divided by 10; these students are assumed to have missed one or more of the quizzes. The output file should contain a line (or lines) at the beginning of the file providing appropriate column headings. Use formatting statements to make the layout clean and easy to read.

After writing the required data to an output file, your program will close all files and then copy the contents of the "output" file to the "input" file by reopening the input file for writing and opening the output file for reading. Thus, the net effect of the program is to change the contents of the input file. **Do not** attempt to copy the output file to the input file until you have thoroughly tested and debugged the rest of your program to ensure that it operates correctly.

You should use at least two functions that include FILE pointers in their parameter lists. (These functions may have other parameters as well.) The input file should be called `quiz.txt` and the output file should be called `average.txt`.

Some test cases you may want to consider are the following:

- What if the input file is empty?
- What if a student does not have any quiz grades at all?
- What if multiple students in a row don't have any quiz grades?
- What if there are extra new lines at the end of the file?
- What if the last record in the file does not have a new line after it but rather ends with end of file?

Name your source code file `program5b.cpp`.

This assignment is typically viewed as being the most difficult of the semester, not because it is long, but because it is troublesome to get precisely correct without breaking something else that seemed to be working before. Please start early and see the instructor if you need help.

One very useful strategy is to write a function that, given a FILE pointer, will extract out exactly one person's full name and write it to the output file. Write another function that given an FILE pointer will extract out one person's quiz scores and list them in the output file along with the average score. Then use these two functions in a loop that will keep doing this until there is no more data in the input file.

Rubric: **Your code must compile, link, and execute in the Linux environment using gcc no compile equals no credit.**

In addition, you must include a comments section at the beginning of each of your files that provides information about the file and its intended purpose. For example,

```
/*
  Author:  David B. Adams
  Course:  COMP 141, Computer Programming I
  Date:    Today’s Date
  Description:  This file implements the
                functionality required by
                Program 5A.
*/
```

The following criteria will be used to grade your submission:

- Does the code compile?
- Does the code function according to the problem specification?
- Is there an appropriate comments section at the beginning of each file (similar to the one shown above)?
- Is the code readable and well-formatted? Is it well-documented and clear?
- Do you have pre and post conditions for each of your functions?

The available points are distributed according to the following weights:

Correctness:	85%	*Supports course outcome 5*
Comments:	5%	
Organization:	10%	

A program that does not compile or link will not be graded.