Lab 3 File I/O

Goals-

Properly open and close files

Read data from a text file

Write data to a text file

Test coverage for testing

Write a program that merges the numbers in two files and writes all the numbers into a third file. Your input comes from two different files and writes the output to a third file. Each input file contains a list of numbers of type int in order from the smallest to the largest. After the program is run, the output file will contain all the numbers from the two input files in one longer list in from smallest to largest. Your program should define a function that is called with the two input-file streams and the output-file stream as three arguments.

You should not read in all the numbers and sort them. Read from each input file and deicide which number should be written into the output file. Then read another number from that input file.

Remember to check all files to make sure they open correctly.

HINT: Please practice incremental development and work in small pieces. As an example- First open a file and print it to the screen. Then add the output file and just copy the first file into the second. Then add the second file and just copy each into the output file. Once all of that is working, add the sorting of the input from the 2 files into the third. Again, this is an example but work in small pieces to make debugging easier.

HINT: Do not test on just 2 input files. There are at least 4 (significantly) different configurations of input files that could change the behavior of your program. Do not make assumptions about the input files, such as the number alternate in order. For example, one file having all the odd numbers in order while the other only has even numbers. As part of your analysis you should consider the different combinations of numbers in the input files.

You should also include a copy of 2 input files you used for testing. That way if we have a question we can make certain that the program worked correctly for you. The TAs are welcome to create their own files to test your program.

Modular Grading

We are using modular grading. Each lab will be divided into specific modules. Each module will be graded pass/fail. It either works properly or it does not. 10% of every lab

or assignment grade is style/comments or other elements of self-documenting code and clarity. Remember the labs are worth 10 points total.

Programming style- 1 point

You successfully open and close the files and test for success- 3 points

You successful create and save a sorted file- 4 points

Provide a brief description of how you tested your program, specifically the different combinations of input files you used- 1 point