Lab C File I/O and Recursion

Goals-
Properly open and close files
Read data from a text file
Write data to a text file
Create and use a recursive function

Recursion? Oh no! Recursion has a bad reputation, but those are people who may not understand it. Recursion is a new abstract concept and requires some study. Take advantage of the help sessions or contact one of us for help.

Write a program that reads in the contents of a file, uses a recursive function to reverse the characters read in, and writes the results to a separate output file.

Prompt the user for the name of the input file and the output file. Use exceptions and report any file errors to the user.

Read in the contents of the input file into a dynamic array. It will start at size 50. If it is full and you are not at the end of the file then you will create a new array of twice the size and copy the characters read so far. Then continue reading the values. If the array fills again, double its size, copy the values, and continue reading.

Once the file has been read in, create a **recursive** function that will reverse the contents of the array. You do not need to count the number of values read in. The base or stopping case is an empty array. You simply need to put each value at the appropriate end of the output array.

NOTE: See the example at this end of this file.

For this lab you can use a single array. Pass it as a reference parameter in the function then write it to the output file. If you have difficulty understanding that and want to use 2 arrays that would be fine also.

When the function has reversed the input array, write the contents order to the output file.

Too many students make this harder than it is by writing code then thinking about it. Some just write code then keep “debugging” until they get it to do something that looks close. Start with paper and pencil and develop the algorithm. Then write your program.

You should prompt the user to enter the names of the two input files and the output file.

HINT: Please practice incremental development and work in small pieces. For 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 opening the second file and just copy each into the output file. Once all of that is working, add the selecting of the input from the 2 files to write to the third. This is just an example but work in small pieces to make debugging easier.

Grading

Programming style- 1 point

You successfully open and close the files and report any errors- 2 points

You successfully create read the data- 2 points

You successfully manage the dynamic array- 2 points

You successfully use your recursive function to reverse the input- 2 points

You successfully write the data to the output file- 1 point

Include your makefile and all files in a zip archive or it will not be graded.

Recursively reverse a string?

To reverse a string you take off the first character, reverse the remaining substring, and put the character back into the string.

```
Reverse string “ab”, remove ‘a’, reverse “b”
   Reverse string “b”, remove ‘b’, reverse “”
          Reverse string “”, return “”  //base case!
   Return “” + b  //concatenation
Return “b” + a  //creates “ba”
```

Voila! You’re finished.

NOTE: You do NOT need to know the length of the string, nor keep any count while doing the reverse function.