

CSC 362 Programming Assignment #1  
Due Date: Friday, August 28

This assignment tests you over char-by-char file input/output and logic. The program operates as follows. Input from the user an int value; this number determines how often to “skip” a character from the input. Open both an input and output file. Read from the input file char-by-char using a while loop, outputting all whitespace and punctuation to the output file and all letters/digits except for those indicated by the skip value. After the loop exits, close both files and output a report (see below). The while loop to input will be similar to those we looked at in class. The first input file is shown below. Assume the user enters 5 for the skip value. Every 5<sup>th</sup> letter/digit is skipped, all other characters are output to the output file.

Input file:

The Bengals lost 3 games in a row to have a 9-7 record for the 2005 season.  
The 3-game losing streak was their longest in 3 years!

Output file:

The Bngal lost game in a rw to hve a 9- recod for he 205 seaon.  
Th 3-gam losig strak wa thei longst in year!

The first four letters/digits are output, the fifth (the second ‘e’) is skipped, the next four (“ngal”) are output, the fifth (‘s’) is skipped. Only digits and letters are skipped (above, aside from letters, the first ‘3’, the ‘7’, the second ‘0’ and the last ‘3’ are all skipped) and white space (spaces, enters, tabs) and punctuation are not part of the count used to determine what characters to skip. You might notice in skipping both 3s that there are two blank spaces between the words (“lost” and “game”, “in” and “years”) as these are the spaces on either side of the omitted digit.

Also count the number of characters input from the input file, the number output to the output file, and use these to compute the percentage size difference between output and input files:  $\text{outputsize} / \text{inputsize} * 100$ . Output a report (to the console window using printf) of the name of both files and their sizes, and the percentage difference. These should be formatted so that the filenames line up, the sizes line up, and the percentage is output to 2 decimal points of accuracy, with a % at the end. Use proper formatting as in “%15s” to control how things line up. Store the filenames in string variables rather than hardcoding them in both the fopen and printf statements. You CAN hardcode them when declaring the strings as in `char infile[] = "input.txt";` Below is a sample of the output when run on the above input.

```
Enter the skip number: 5
Size of infile.txt :      133
Size of outfile.txt:      113
File size change:        84.96%
```

Below this document in Canvas are three text files. Run your program on the first textfile using 5 as the skip number and test your results to those shown above. Once your program is running correctly, run it on the other two textfiles using 7 and 3 respectively for the skip numbers. Submit your source code, the output file created and the output from the console window when run on the second and third input files. This amounts to 5 items. For convenience, you can copy and paste all five items (source code, output files \* 2, console output \* 2) into one word/pdf document or you can copy the output files’ contents and the console output into your source code file (in comments) at the bottom and submit a single file. Email your file to [foxr@nku.edu](mailto:foxr@nku.edu).

NOTES: Aside from the filenames, there is no reason to use an array in this program. To determine if a character is punctuation, whitespace or other, use the functions available in the ctype.h library. Make sure you comment your code well. See <http://sappho.nku.edu/~foxr/CSC362/sample.c> for an idea of how you should comment your code.