CSSE 132 – Introduction to Systems Programming
Rose-Hulman Institute of Technology

Homework 7

On your Linux, pull changes to your Git repository. In the `homework7` directory:

1. (20 points) Complete the functions in the `problems.c` file. The comments in the file tell you what needs to be completed.

   Test them using the provided tests: `make test` and `./test`. Note that, in grading, we will also manually check for proper amounts `malloced`, properly closing files, etc.

2. (20 points) `Censor`: Create a program called `censor` that opens and reads a file, printing a censored version of the file as it reads. A censored version has changed all four-letter words `ABCD` so the middle two characters are asterisks: `A**D`. A more explicit example can be found below when handling a specific file:

   ```
   usr@linux: $ cat test.txt
   This is a test file. "-test-", 'test', text?
   usr@linux: $ ./censor ./test.txt
   T**s is a t**t f**e. "-t**t-", 't**t', t**t?
   usr@linux: $
   ```

   Note that a word can consist of alphabetic characters. And any non-alphabetic characters separate words.

   To complete this problem, you must:

   - ☐ Implement the code in a new file called `censor.c`
   - ☐ Use `make censor` to compile your program
   - ☐ Read the file specified by an argument passed to the program. For example: `./censor file.txt`
   - ☐ To test your program, you can run the test script `./test_censor.sh`

   Suggestions and hints:

   - In this part, you can assume no word will be split into different lines. In addition, the number of characters in each line must be less than 1024.
   - Test the `censor` program on the command line by running it with various files. See the `censortests` subdirectory provided with this homework for example inputs (end in `.txt`) and outputs (end in `.censored`).
   - Consider copying one or more of the functions from `problems.c` into your new file.

When finished, submit your `problems.c` and `censor.c`to **Gradescope**.