CIS*2500 W20: Assignment 1 Grading Scheme

Part 1: `file_size()`	2%
Part 2: `read_words()`	10%
Part 3: `print_words()`	10%
Part 4: `alphabetical_word_count()`	10%
Part 5: `print_alphabetical_word_count()`	6%
Part 6: `create_alphabetical_array()`	18%
Part 7: `free_alphabetical_array()`	8%
Part 8: `print_words_alphabetically()`	10%
Part 9: `main()`	6%
Functions should be callable by other programs (*i.e. not in same file as main()*)	2%
Consistent Indentation	6%
Good variable names	3%
Good use of white space	1%
Use of #defines for constants	2%
Proper use of headers	2%
Commenting	4%
Read.me	2%

Notes:

1. Your main program should be called `alphabetical_printing.c`
2. List all of the other file names used for your source code in your readme file.
3. You must use dynamic memory (malloc) to create the arrays in this assignment.
4. You cannot change the function names or the number, type or order of the parameters to the functions.
5. Ensure there is a copy of your source code files in a separate directory where they will be safe in the event that you accidentally remove or overwrite everything from your working directory. Alternatively, use your school Git repository.
6. The functions you write should NOT print anything to stdout except for the functions that explicitly are supposed to print. All printf calls for any other function that may have been used for debugging should be removed, #IFDEF'd, or commented out before submission.
7. Your code must compile cleanly with no error or warning messages using the -Wall flags in gcc.
8. The assignment must be written in C and compile and run on the school's Linux server.
9. Your source code should contain brief comments describing the functionality and the major components of each procedure. Any complex structures should also be commented. Your source code should be properly formated and meaningful variable names should be used.
10. **If you hand in an assignment that does not compile you will get a mark of zero.**
    - If you only complete part of the assignment, make sure that you write a main() that demonstrates the abilities of the code you wrote
    - Document which parts of the assignment is functioning and how your mainline demonstrates it, both in comments in your source code and in your readme file.
11. **You should hand in your source code file (.c and .h files) as well as a makefile.**

All work in this course is to be done independently.
Submissions will be electronically examined for similarity.