Files to submit: **read_lines.c, read_lines.h**
Time it took Matthew to Complete**: 15 mins**

- All programs must compile without warnings when using the -Wall and -Werror options
- If submitting multiple files on Kodethon place them into a zip folder
  - Make sure to zip the files themselves and **NOT** the directory they are inside.
- Your program must match the output exactly to receive credit.
  - Make sure that all prompts and output match mine exactly.
  - Easiest way to do this is to copy and paste them
- All input will be valid unless stated otherwise
- Print all real numbers to two decimal places unless otherwise stated
- The examples provided in the prompts do not represent all possible input you can receive.
- All inputs in the examples in the prompt are underlined
  - You don't have to make anything underlined it is just there to help you differentiate between what you are supposed to print and what is being given to your program
- If you have questions please post them on Piazza

Restrictions

- No global variables are allowed
- Your main function may only declare variables, call other functions, and assign variables values.

For this assignment you will be writing only a single function: `read_lines.`

Specifications

1. This function has the following definition:
    1. `void read_lines(FILE* fp, char*** lines, int* num_lines).`
2. This function should read all of the lines contained within `fp` and
    1. Set each row of lines to contain one line of the file.
    2. Set num_lines to be equal to the number of lines that were in the file
3. If the file is empty `lines` should be set to NULL and `num_lines` to 0.
4. You only need to submit read_lines.c and read_lines.h
5. `read_lines.h` must contain at least the definition for read_lines but it is perfectly ok if there are more

Example

1. ./read_lines.out Makefile
    1. read_lines.out: read_lines.o main.o
    2. gcc -g -Wall -Werror -o read_lines.out read_lines.o main.o
    3.
    4. main.o: main.c read_lines.h
    5. gcc -g -Wall -Werror -c -o main.o main.c
    6.
    7. read_lines.o: read_lines.c read_lines.c
    8. gcc -g -Wall -Werror -c -o read_lines.o read_lines.c
    9.
    10. clean:
    11. rm -f *.out *.o