Exercise 1:

Given the following header and source files:

prog.h

```
typedef struct {
  int total_chars;
  int letters_count;
  int words_count;
  int lines_count;
  int max_line_length;
} FileStats;

/* returns 1 if param is a letters
   and 0 otherwise */
int is_letter(char);

/* takes a filename as a string param
and returns a pointer to a FileStats
structure, or NULL on failure */
FileStats* process_file(char*);

/* takes a filename as a string param
and returns an array contains all the
lines of the file, or NULL on failure
*/
char** get_lines(char*);
```

prog.c

```
#include <stdio.h>
#include <stdlib.h>
#include "prog.h"

int main(){

  /* local var declarations */

  fsp = process_file("wcs.txt");

  /* printf the returned stats */

  lines = get_lines("wcs.txt");

  return 0;
}

/* define the functions
   you can write function stubs
    during development */
```

1. Launch the terminal
2. Create a new directory with the name "Lab08" inside "CSC215"
3. Write a C file "prog.c" the contains that:
    a. implements the function `is_letter` — 1 point
    b. implements the function `process_file` — 3 points
    c. implements the function `get_lines` — 3 points
2. Complete the function `main` so the program does what is required. — 1 point

Note: words are separated be no-alphabetic characters.

Assignment:

Add to your program the function:

```
void write_rev(char*, char**, int);
```

Which takes a filename as a string parameter, an array of strings and the number of strings as an integer, and stores the strings in a text file called filename in a reversed order (i.e the first string should be the last line of the file, …).

Then modify your main function to store the lines that was read from "`wcs.txt`" into the file "`wcs-rev.txt`" in reversed order. — 2 points