PART 1

Input: Modify `in.txt` with the phrase to be tested
Compile: `gcc lab1p4_1.c -o lab1p4_1`
Run: `./lab1p4-1`
Output: see `out.txt`, format gives word and its count in each line

Notes:
Individual words must be 254 characters or less.

Citations:
Lines 8-11 inspired from
https://www.tutorialspoint.com/data_structures_algorithms/hash_table_program_in_c.htm

Line 34 comes from
https://stackoverflow.com/questions/3463426/in-c-how-should-i-read-a-text-file-and-print-all-strings

Lines 38-40 come from
https://stackoverflow.com/questions/2661766/how-do-i-lowercase-a-string-in-c

PART 2

Input: Modify `in.txt` with the phrase to be tested
Compile: `gcc lab1p4_2.c -o lab1p4_2`
Run: `./lab1p4_2`
Output: see `out_improved.txt`, format gives word and its count in each line

Notes:
Individual words must be 254 characters or less.
Words with the same alphanumeric characters are considered the same word.
The output file may print non-alphanumeric characters, but the program will ignore them when testing equivalence.

Citations:
Lines 8-11 inspired from
https://www.tutorialspoint.com/data_structures_algorithms/hash_table_program_in_c.htm

Lines 30 and 35 based on
https://www.geeksforgeeks.org/isalnum-function-c-language/

Line 74 comes from
https://stackoverflow.com/questions/3463426/in-c-how-should-i-read-a-text-file-and-print-all-strings

Lines 78-80 come from
https://stackoverflow.com/questions/2661766/how-do-i-lowercase-a-string-in-c