Important Notes

- Deadline:
  Electronic submission is due on Monday, Sept 15th, 2014 11:59pm.
  Hard copy is due on Tuesday, Sept 16th, 2014 in class.
- **This assignment is to be done on your own.** If you need help, see the instructor or TA.
- Please start the assignment as soon as possible and get your questions answered early.
- The purpose of this assignment is to get you started in writing programs in C.
- Read through this specification completely before you start.
- Some aspects of this specification are subject to change, in response to issues detected by students or the course staff.

1. **Counting (30pts):**
   Using only the features of C you have learned in so far write a C program that takes text as input, and outputs the following:

   - the number of lines in the input
   - the number of characters in the input
   - the number of words in the input

   All characters, including the newline ('\n'), should be counted.

   Assume that only space(s), newline characters, and tab characters seperate words. That is: ' ', '\n', and '\t'.

   Input is terminated by entering CTRL-D at any time.

   Instead of typing long paragraphs to test your program, you can use text file using I/O redirection. I/O redirection on Unix converts text files to standard input for programs. For example, if the text is stored in a file, textinput, you can input it into the program (a.out) by using I/O redirection as follows:

   $ ./a.out < textinput

   The results will then be printed on the screen. Alternately, you can do:

   $ ./a.out < textinput > results

   The results will be printed in a file called, results

   Use the unix program *wc* to verify the correctness of your program.

2. **Sum of numbers as characters (70pts):**
   Write a program that reads numbers in one by one, and then add them up. The numbers are read in as characters.

   Your program must do the following:

- Take input from the user, one number (both integer and floating point numbers) on a line, until user types CTRL-D on its own line.
- Use EOF to detect end of input. In keyboard input, EOF is signaled by typing CTRL-D.
- Read the input as characters, one character at a time (i.e. use getchar() only).
- Remember that character digits are not integer digits, i.e. you must perform conversions between these.
- Once you have the digits in proper integer format, think of a way to convert a collection of digits into proper numbers, i.e. multiplying powers of 10 correctly.
- You should perform adquate error-checking to guarantee program robustness. Your program should not crash under any circumstances, this includes invalid or unreasonable inputs.
- You may assume however, that the user will not enter integers or floats in sizes bigger than machines can handle.
- Reading the numbers as doubles or integers and then adding them up earns 0 credit on this part.

Extra Credit (15pts):

- Leading 0 in a floating point number is optional, that is, .4 and 0.4 are equivalent and both acceptable.
- Handle negative numbers correctly.

3. **Submission:** Copy all your files (including source code and executables) into a directory named YourID_P2. Run your program to make sure that it behaves as expected and then copy the whole directory to the CS server powerpuff.cs.brynmawr.edu. Submit the directory using the following command:

```
$ submit -c 246 -p 1 -d YourDirectory
```

For more information about submission, see the submission instruction online:

`https://systems.cs.brynmawr.edu/wiki/Submit_assignments`