CS 280: Program #1

Fall 2015

Due September 23 at 11:55PM, via Moodle

For this assignment, you will write a C++ program that "filters out" certain things from input. The following definitions apply:

- letter: upper or lower case a through z
- vowel: the letters a, e, i, o or u
- consonant: letters that are not vowels
- word: a sequence of sequential letters and/or numbers separated by punct or space
- number: zero through 9
- space: whitespace characters such as space, tab, or newline
- punct: punctuation marks such as quote, comma, period, etc.

- upper: uppercase letter, vowel, consonant or word
- lower: lowercase letter, vowel, consonant or word

In this assignment, the user is going to provide the name of a file and one or more of the items in the list above. The items tell the program to filter out any characters that match the item provided. Note that upper and lower are optional modifiers for letter, vowel, consonant or word. If present, the modifier must appear immediately before letter, vowel, consonant or word.

All input to the program is provided as command line arguments. The first command line argument is taken to be the name of the file and all subsequent arguments are from the items list, above. The program should generate its output to the standard out.

As an example, let us assume we are given a file named input.txt which contains:

> This is a demo, 100 times over,
> for this assignment.

The table below shows command line arguments and the result of running your program. For this example, I show your program being named "filt"

command	meaning	output
filt input.txt letter	Remove all letters (leaving spaces, punctuation and numbers	, 100 , .
filt input.txt lower consonant	Remove all lowercase consonants	Ti i a eo, 100 ie oe, o i aie.
filt input.txt upper letter number punct	Remove upper case letters, numbers and punctuation	his is a demo times over for this assignment

Your program should accept command line arguments. The first argument is the name of a text file containing the input text. There must be AT LEAST one additional argument provided. Your program should read the file named in the first argument as input and generate its output to standard out. The program finishes when the end of the input file is reached.

Remember that your program needs to check for errors and react appropriately. What happens if a proper number of command line arguments is not provided? What if the text file cannot be opened? What if an unrecognized command is given?