PROGRAMMING ASSIGNMENT #1
Reading virtual /proc files

Topics covered:

Obtaining command line parameters
Obtaining files from Project Gutenberg
Basic function handling, including multiple assignment for returning multiple values
Opening a file and reading it with `for vbl in file_handle`
Entering Unicode characters, both as code points and as UTF-8
Strings, including processing with `for vbl in string` (not with subscripts!)
Basic arithmetic
Basic statistics (chi-square)

Specifications:

1. Read two parameters from the command line with the names of two files you have obtained from Project Gutenberg. (I will also load the files on turing.) If the command line does not have exactly two parameters, display an error message and quit.

Some possible test files are:
Pride and Prejudice in UTF-8: `https://www.gutenberg.org/files/1342/1342-0.txt`
Du côté de chez Swann in UTF-8:
`https://www.gutenberg.org/files/2650/2650-0.txt`

2. Write a function that opens a file and does the following:

a) Prints the number of lines, number of characters and number of letters in the file with appropriate rubrics. A letter is any character whose Unicode category code starts with 'L'.
You can see the Unicode categories here:
http://www.unicode.org/reports/tr44/#General_Category_Values
and you can learn how to access them in Python here:
https://docs.python.org/3/howto/unicode.html

b) Print the name(s) of the system utilities you used to verify that the number of lines and number of characters are correct.

c) Count the number of vowels the file contains. (Convert to lower case for simplicity.) We will consider the vowels to be a e i o u and the twelve French vowels with diacritics:

U+00e9 = e acute
U+00e2 - a circumflex
U+00ea = e circumflex
U+00ee = i circumflex
U+00f4 = o circumflex
U+00fb = u circumflex
U+00e0 = a grave
U+00e8 = e grave
U+00f9 = u grave
U+00eb = e dieresis
U+00ef = i dieresis
U+00fc = u dieresis

d) Print the number of vowels, number of consonants, total number of letters and percent of vowels with appropriate rubrics. A consonant is any letter that is not one of the vowels listed above. Percents should be rounded to 2 decimal places.

e) Return the number of consonants and number of vowels.

3. Use the chi-square statistic to investigate the hypothesis that one of these texts has a greater percent of vowels than the other. Look at the number of consonants and the number of vowels. Do the arithmetic as shown on this web page:

http://psc.dss.ucdavis.edu/sommerb/sommerdemo/stat_inf/tutorials/chisqhand.htm

Print the chi-square statistic and your conclusion about whether the result is significant, using the format in the last paragraph of the web page.

Background:

This assignment is mainly a way to do something interesting with real data without using any complex data structures. Even so, counting letters is useful in several real-world contexts:

- Data compression. Zip and related algorithms work by assigning shorter bit patterns to the most common letters in a document.

- Cryptography.

- Language identification, e.g., the “detect language” feature in Google Translate.

- Author identification. Common approaches to author identification use vocabulary, function words (e.g., prepositions), or content words (e.g., nouns, adjectives and verbs), but this paper makes a case for using low-level features like letter counts as well:

Stamatatos, Efstathios. (2009). A survey of modern authorship attribution methods. *Journal of the American Society for Information Science and Technology*, 60(3): 538-556.