Using dictionaries.

(1) Download the file 'getty.txt' save it in the directory for this lab. This file contains the text of the famous Gettysburg Address, made by Abraham Lincoln, on November 19th 1863. I want you to write me a some code which:

(a) - Contains a function (with one parameter, the filename as a string) that opens this file, reads in the text as a single string and returns this string.

(b) Write a function that takes a string, and returns a string that is all lower case, and have no punctuation. That means we're going to need to replace all the punctuation with...nothing.

You could manually list all punctuation, but there's an easier way. At the top of your file, add a line that says

import string

Then inside your function, do

print(string.punctuation)

to find out what string.punctuation contains. Remember you'll have to call your function to see it.

You've should have seen that string.punctuation is a single string, that contains all the known punctuation.

Now, how can it help you? You can go through string.punctuation one character at a time, (using a loop of some sort) and use the string.replace() method. Replace each character from string.punctuation in the string you pass to the function with nothing.

You DO NOT need to check to see if each character exists, you can simply call the string method replace. There is no error caused if you try to replace a character that doesn't exist in your text. For example:

gettyString = '!!nick!'

gettyString = gettyString.replace('#','')

does not generate an error. Nothing happens. The string in blah doesn't change. But:

gettyString = gettyString.replace('!','')

means that blah now contains 'nick', with all the '!' removed.

HINT: Notice how I'm keeping the same variable name, so that changes I make are kept

HINT: You MUST use a loop to go through the elements of string.punctuation one at a time, you cannot use the characters in string.punctuation all in one go.

(c) TEST your code from (b), by passing it the string "NiCk!!! iS, AWEsome??,". It should return "nick is awesome"

(d) - Write a function which takes one argument, a string, and constructs a dictionary of words contained in that string as keys, and a count of the number of occurrences of each word as values. Return the dictionary.

The string argument here could be a word, a sentence, or the entire string contents of a file. You CANNOT use any methods called count (string OR list) to do this. You DO need to split your string into words.

HINT: This isn't hard, but think about the steps the function needs to take. You need to create an empty dictionary. THEN you need to go through the words of your text one at a time. Then....what? If the word you're currently looking at is already a key in your dictionary, you want to add one to the count, which is stored as the value.

HINT: How do you know if a word you're looking at is a key in your dictionary? Do you remember the 'in' operator? How does that work? What does it return? Look back at your notes, and try some stuff with a simple dictionary in the command window. Again - you want to know IF the key is IN your dictionary.

If on the other hand, if the word ISN'T yet a key in your dictionary, you need to create an entry in the dictionary with the word as the key, and set the value to 1.

This means that when you've gone through every word in the string input, there should be an accurate count in your dictionary of how many times each word appears in the text.

Have this function print the TOTAL NUMBER of words in your text. Then return the dictionary.

(e) Test your function from (d) by passing it the string "cat cat cat zebra zebra dog dog". Make sure the counts are correct.

(f) Write a function that takes a dictionary of word counts, and prints the most frequent word, and its count, AND returns the count. HOWEVER, many frequently occurring words aren't very interesting. So, add a condition that the word has to be longer than 5 characters in length. That is, print the most frequent word longer than 5 characters, and its count.

(g) Write a function called lab10(). Have it use your function from (a), to open the text of the speech. Then use your function from (b) to strip punctuation, and then use your function from (d) to count the words. Then use your function from (f) to print the most frequent word with more than 5 characters.

(h) Write a function to output the top 8 words in order of count from a text. You MUST implement it like this:

The function takes a dictionary of counts as an argument

Get the maximum count, using the function from (f).

Set a counter to zero.

WHILE the number of words printed is LESS than 8

Go through all the words in dictionary, and print any word with the max count value that is ALSO longer than 5 characters

For each word you print, increment the number of words printed.

Once you've iterated through and printed ALL the words with the max count value

Take one away from the max count, and repeat the process.

Add a call to (h) in your function from (g).

For instance, when I run it with the Gettysburg address, I get the following output:

nation 5

dedicated 4

people 3

conceived 2

dedicate 2

living 2

rather 2

devotion 2

(i) In lab 7, you imported the urllib.request library, by writing:

import urllib.request

at the top of your file.

Let's do it again. We're going to download ALL the text of the novel Dracula from Project Gutenberg.

There's another issue with decoding text, just like with the sharks CSV file.

Copy the following function:

```
def get_dracula():
    response = urllib.request.urlopen('https://www.gutenberg.org/files/345/345-0.txt')
    data = response.read()      # a `bytes` object
    dracula_text = data.decode('utf-8')
    return(dracula_text)
```

Add in a call to this function to (g), and comment out the call to your function that gets text from the gettysburg address. Make sure all your functions work with the text of Dracula.