CSC Lab 8 Dictionary and File I/O

Prof. Xiao

Assigned: November 16th, Tuesday.

Due: November 21, Sunday

This lab is extra credit. However, it is very useful for the upcoming mid-term exam.

We will also discuss some of them on Wed’s lectures. So it is encouraged you get a head start to work on them in Today’s class.

Bonus 16 points will be added to the Lab portion.

You might find the following reading materials useful:

<https://www.greenteapress.com/thinkpython/html/thinkpython012.html>

1. (8 pts) Please write code to do the following:

Here are 5 tallest mountains in the world, the names and the elevations. Save all of the following code in one file: Question1.py

Mount Everest 8445m

K2 8611m

Kangchenjunga 8586m

Lhotse 8616m

Makalu 8516m

a). Create a dictionary with the mountain's name as key and elevation as values.

b). Print out just the names of the mountains by looping through the keys (using for loop)

c). Print out just the values of the mountains by looping through the values (using for loop)

d). Using for loop, print out series statements of how tall each mountain is: "Everest is 8848 meters tall." Make sure the output is **in alphabetical order.**

e). Create a new dictionary, where the keys of the dictionary are still the mountains' names. This time however, the values of the dictionary should be a list of each mountain's elevation in meters, and then in feet: {'everest': [8848, 29029]}. A meter is approximately **3.28 feet**.

f). Print out just the mountains' elevations in meters, by looping through the values of your dictionary and pulling out the first number from each list.

g). Print out just the mountains' elevations in feet, by looping through the values of your dictionary and pulling out the second number from each list.

h). Print out a series of statements telling how tall each mountain is: "Everest is 8848 meters tall, or 29029 feet.”

1. (8pts) This is last year’s mid-term exam question. So, solving this question will help you prepare the mid-term exam.

Download the file “election.txt” from canvas.

This file contains some comments from Twitter about the election. Please save the answers as question2.py and use comments in your code to label each sub-questions.

A sample text from the .txt file could be the following:

Meanwhile, another 4,500 votes came in from Georgia, which went 63 percent to 36 percent for Biden, but given the small number, they moved things only very slightly in Biden’s direction. According to ABC News, Trump now leads Biden by 0.27 percentage points, or 13,540 votes. Earlier today, we heard there were about 61,000 votes left to count, but since then, around 10,500 more votes have come in, and Georgia’s secretary of state just said that 50,401 ballots are still outstanding.

1. (2pts) Open the file using Python and read the file line by line and convert the words in the file into a list of words. (see previous lectures Lecture 20 and 21 for demos).
2. (2pts) Sort the word list from longest length to shortest length, and print out the sorted word list excluding the punctuations. (see Lecture 21 and demo codes)
3. (2pts) Find out whether the sample text mentioned a battleground state in the election such as “North Carolina”, “Georgia”, “Arizona” or “Nevada”, and  
   “Pennsylvania”. Report the name of the State and how many times it appears in the text.

Do the same thing with the candidates “Trump” and “Biden”.

**For example, for the above text, it could return “Georgia, 2", "Trump, 1", "Biden, 3" . The format of your result could look different from the example.**

1. (2pts) Find all the numbers in the list and print out the number and the word right after the number. E.g. 61,000 votes, 10,500 more, 50,401 ballots.

**Hint**: string.isnumeric() checks if a string is a number.

Note that the actual file you received might not be the same as the one written here. This is only an example.

Here are some sample code of how to read texts from a file into Python Dictionary

wordcount = {}. # creating an empty dictionary

with open('test.txt', 'r') as f:

for line in f:

line = line.lower() #I suppose you want biden and Biden to be the same word

for word in line.split():

#if it's already in the d increase the number, instead of try or except, you can use “ if word in d:” instead.

try:

wordcount[word] += 1

#if it's not this is the first time we are adding it

except:

wordcount[word] = 1

you might want to find the starter code lab8\_q2.py helpful.