CSCI 1300 - Intro to Computer Programming
Instructor: Fleming/Gupta
Homework 9

Due Sunday, April 29th, by 6 pm
+5% bonus if submitted by Friday April 27th 11:55 pm,
+2% bonus if submitted by Saturday April 28th 11:55 pm

This assignment is due **Sunday, April 29th 6pm**.

- ***All components*** **(Cloud9 workspace, moodle quiz attempts, and zip file) must be completed and submitted by Sunday, April 29th 6:00 pm for your homework to receive points.**
- Complete submissions (Cloud9 workspace, moodle quiz attempts, and zip file) before **Friday April 27th 11:55 pm** will receive a 5% bonus, and complete submissions before **Saturday April 28th 11:55 pm** will receive a 2% bonus.

The De-slang-ifier

OMG! In this assignment, you are writing a program that converts common texting abbreviations to English words. For the assignment, you are provided with a .txt file, called *TextToEnglish.txt*, that provides the abbreviations and their English translation. (The file contains G-Rated translations only). The txt file has comma separated values that list the key as the first item, a comma to separate the key from the value, and then the value.

TextToEnglish.txt

```
ikr,I know right
b4,before
gtg,got to go
l8,late
l8r,later
nvm,nevermind
nm,nevermind
omw,on my way
y,why
r,are
u,you
```

For this assignment, you need to submit one file called **Homework9_LastName.py**, where LastName is your last name.

- 4 points for code running
- 6 points for two different sample runs (for questions 3 -5)
    - add your sample runs as a separate .txt file or as comments in your Homework9_LastName.py
- 10 points for an algorithm description for each function
- 10 points for comments
    - required comments at top of file

      # CS1300 Spring 2018
      # Author:
      # Recitation: 123 – Favorite TA
      # Cloud9 Workspace Editor Link: https://ide.c9.io/…
      # Homework 9

    - comments for functions

Question 1

Write a function called **countNames** that takes in a list of names (each element being first name last name), and returns a dictionary with key as first name and it's value as a list of last names.

e.g.

```
name_list = ["john oliver", "emma roberts", "john mccain", "emma watson", "emma stone", ]

nameDictionary = countNames(name_list)

nameDictionary: {"john" : ["oliver", "mccain"],

                                "emma": ["roberts", "watson", "stone"]
```

Question 2

Write a function, **compute_grade(...)**, that takes a parameter, file_name, and computes the final grade of class given the grades of each section of the course and the weights associate with the section. Your function should read each line of a file, compute the average of the values associated with the section and weight the average by the weights indicated. The final grade is the sum of the weighted averages. The function should return the final grade of the class.

Notes:

1) The values are comma separated.

2) If the file is not found return "File not found".

Example:

If the content of the file "csci1300_grades.txt" is:

```
90,90,90,Practicum,0.2

90,90,90,Recitation,0.3

90,90,90,Assignment,0.2

100,Assignment4,0.05

100,Assignment7,0.1

100,Assignment8,0.15
```

When following code is executed:

```
file_name = "csci1300_grades.txt"

print( compute_grade(file_name) )
```

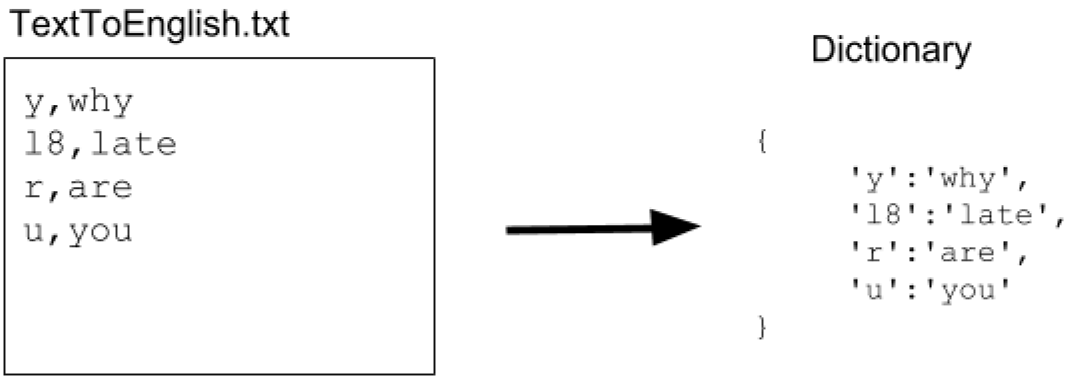
This should be the output:

```
93
```

Question 3

Write a function, called **update_dictionary** that has two parameters: a filename and a dictionary. The function should:

1. Open the given file
    a. If the file open succeeds print `"TextToEnglish.txt loaded successfully."`
    b. If the file open fails: print `"TextToEnglish.txt does not exist."`
2. Read in the data from the file and fill the dictionary.



TextToEnglish.txt

```
y,why
18,late
r,are
u,you
```

Dictionary

```
{
    'y':'why',
    '18':'late',
    'r':'are',
    'u':'you'
}
```

The dictionary key is the text abbreviation and the value is the English translation. For example, one entry in your dictionary will be, ‘l8’: ‘late’ because one of the rows in the txt file contains ‘l8’ and ‘late’.

3. Provide the new size: print "`The dictionary has 4 entries.`"
4. Return the updated dictionary.

Test your function before moving on to the next question!

Question 4

Write a function **deslang** that takes two parameters, a string and a dictionary, and returns the deslanged string. Each word in the string will be replaced if it has an entry in the dictionary. Any word not in the dictionary should be copied to the results.

```
deslanged = deslang(slang, wordDictionary)
```

For example, if the slang string is: `“David, y r u l8”`
your function should return: `“David, why are you late”`

Question 5

Your **main** function in the .py file containing your functions, **update_dictionary**, *and* **deslang**, should provide the following menu:

```
Would you like to (a)dd words to the dictionary, (d)e-slang a
sentence, or (q)uit?:
```

For this assignment, will be testing your code by running your *entire* file (as we did in Homework 7). Please reference the following sample runs for your menu functionality:

Present the menu until the user chooses 'q'

```
Would you like to (a)dd words to the dictionary, (d)e-slang a
sentence, or (q)uit?: q
Would you like to (a)dd words to the dictionary, (d)e-slang a
sentence, or (q)uit?: r
Would you like to (a)dd words to the dictionary, (d)e-slang a
sentence, or (q)uit?: s
Would you like to (a)dd words to the dictionary, (d)e-slang a
sentence, or (q)uit?: q
```

Add words (prompt user for filename until they provide a non-empty string)

```
Would you like to (a)dd words to the dictionary, (d)e-slang a
sentence, or (q)uit?: a
Enter a filename:
Enter a filename:
Enter a filename:
Enter a filename:
Enter a filename:
Enter a filename:
Enter a filename:
Enter a filename: textToEnglish.txt
textToEnglish.txt loaded successfully.
The dictionary has 11 entries.
Would you like to (a)dd words to the dictionary, (d)e-slang a
sentence, or (q)uit?: q
```

Add words (file open fails)

```
Would you like to (a)dd words to the dictionary, (d)e-slang a
sentence, or (q)uit?: a
Enter a filename: doesNotExist.txt
doesNotExist.txt does not exist.
The dictionary has 0 entries.
Would you like to (a)dd words to the dictionary, (d)e-slang a
sentence, or (q)uit?: q
```

Add words (file open succeeds)

```
Would you like to (a)dd words to the dictionary, (d)e-slang a
sentence, or (q)uit?: a
Enter a filename: textToEnglish.txt
textToEnglish.txt loaded successfully.
```

```
The dictionary has 11 entries.
Would you like to (a)dd words to the dictionary, (d)e-slang a
sentence, or (q)uit?: q
```

Add words (dictionary should grow if new words are added)

```
Would you like to (a)dd words to the dictionary, (d)e-slang a
sentence, or (q)uit?: a
Enter a filename: textToEnglish.txt
textToEnglish.txt loaded successfully.
The dictionary has 11 entries.
Would you like to (a)dd words to the dictionary, (d)e-slang a
sentence, or (q)uit?: a
Enter a filename: text2English.txt
text2English.txt loaded successfully.
The dictionary has 13 entries.
Would you like to (a)dd words to the dictionary, (d)e-slang a
sentence, or (q)uit?: q
```

Deslang a sentence (prompt user for sentence until they provide a non-empty string)

```
Would you like to (a)dd words to the dictionary, (d)e-slang a
sentence, or (q)uit?: d
Enter a sentence:
Enter a sentence:
Enter a sentence:
Enter a sentence: hello world!
hello world!
Would you like to (a)dd words to the dictionary, (d)e-slang a
sentence, or (q)uit?: q
```

Deslang a sentence (slang in the dictionary should be replaced with the de-slanged version)

```
Would you like to (a)dd words to the dictionary, (d)e-slang a
sentence, or (q)uit?: a
Enter a filename: textToEnglish.txt
textToEnglish.txt loaded successfully.
The dictionary has 11 entries.
Would you like to (a)dd words to the dictionary, (d)e-slang a
sentence, or (q)uit?: d
```

```
Enter a sentence: hey david, y r u late?
hey david, why are you late?
Would you like to (a)dd words to the dictionary, (d)e-slang a
sentence, or (q)uit?: q
```