CIS 41A - Lab 3: List, Tuple, Set, Dictionary, File IO, Exception

Note: you have about 2 weeks to complete this lab because we're moving into Python specific topics and working with many data values, therefore the programming logic is more complex than in the first 2 labs. Please work on the lab at a steady pace as discussed in class, and don't wait until the last days to do it or the lab will be more stressful than it should be.

Write a program that reads in a file of world countries and the languages spoken in each country. The program prints the most common languages and lets the user: search for the languages of a country, or compare languages between 2 countries.

A. Input file

The input file is languages.txt. Each line of the file has 2 or more fields separated by a comma: a country name, followed by the languages spoken in the country, with the most common languages listed first. An example line:

Algeria,Arabic,French,Berber dialects

The input file is a simplified version of the table found at the [Infoplease](https://www.infoplease.com/countries/languages-spoken-in-each-country-of-the-world) webpage.

B. Function description

1. main function:

* Call a function to read in the input file and store the data in your choice of container(s).

Choose wisely based on how the data will be searched so that your code can be shorter and simpler. For example, storing all lines in a list is not the most efficient way to search for data later.

* Call a function to find and print the most common languages.
* Create a loop to:
  + Let the user choose from a menu: 1. Print languages of a country  
     2. Compare languages of 2 countries  
     3. Quit

Keep prompting the user for a choice until the user enters a choice of 1-3. Make sure to handle the case when the user enters a string instead of a number.

*[1pt EC if you don’t use while True or a break statement for the loop control.]*

* + Call an appropriate function to process the first 2 user choices.

2. Function to read the input file:

* Let the user enter a filename or press the Enter key for the default languages.txt file.
* Read in each line of country name and its languages and store the data in the appropriate container(s).
* Print the total number of countries read in.
* You can assume the input file will be opened and read in successfully, so no need to check for valid input file.

3. Function to print the most common languages in the world:

* A "most common language" is defined as a language that is spoken in more than 12 countries.
* Print the most common languages and their corresponding number of countries, sorted by language, and lined up in column format (see sample output).  
  Please *do not use lambda (in any format)* to solve this problem. This is an exercise in working with dictionaries
* If there are non-languages (such as the word "other") that occur in more than 12 countries, don't count it as one of the most common languages.

4. Function to let the user see languages of one country:

* Prompt the user to get a country name. The input name can be in any combination of upper and lower cases.
* Print all languages of the country on one line, separated by comma  
   or print “no such country” if the name doesn’t exist.
* When printing languages, make sure there’s a comma in between the languages, but not after the last language.

5. Function to let the user compare 2 countries' languages:

* Prompt the user twice to get 2 country names. The input name can be in any combination of upper and lower cases.

Keep prompting if a country name doesn’t exist, but only prompt for the invalid name. Making the user enter both country names when only one name is not valid will make your app unpopular.

* Print the languages, if any, that are common to the 2 countries. The languages should be printed on one line, separated by comma, with no comma after the last language.
* Print all the languages of both countries, sorted by language. Each language should be printed one time only. The languages should be printed on one line, separated by comma, with no comma after the last language.

C. Additional requirements

* At the top of the file, put your name and the lab number. You can optionally have a short description of what the program does.
* Have a docstring for each function.
* There should be 5 functions as described above, and each function should do the exact tasks described above. You can have additional support functions if you like.
* Before you dive into the coding phase, it's important to think about what containers are best to store the data, such that you can easily fetch data for the user. The appropriate containers can simplify and shorten your code.

In the words of a CS superstar:  
*"I will, in fact, claim that the difference between a bad programmer and a good one is whether he considers his code or his data structures more important. Bad programmers worry about the code. Good programmers worry about data structures and their relationships." --* [*Linus Torvalds*](https://lwn.net/Articles/193244/)

* *[1pt EC: Use comprehensions when appropriate.]*

*[2pts EC:*

*Do all the work described in green above, which is about half of the lab.*

*Then demo your work before class, after class, or during office hours before or on Thu, Feb 1, office hour.]*

D. Sample output and some sample test cases that you can use. The user input is in blue to make it easier to see.  
(output numbers below are not guaranteed, trust your code instead)

Enter a filename or press Enter for default file: <enter key>

There are 198 countries in the database

The most common languages

Language Num of countries

English 85 Descending order,

French 46 in column format

Arabic 29

Spanish 23

Russian 16

1. Print languages of a country

2. Compare languages of 2 countries

3. Quit

Enter your choice: 1

Name of country: moroco invalid name

Moroco doesn't exist in database

1. Print languages of a country

2. Compare languages of 2 countries

3. Quit

Enter your choice: 1

Name of country: moroCCo case insensitive

Arabic, Berber dialects, French

1. Print languages of a country

2. Compare languages of 2 countries

3. Quit

Enter your choice: 1

Name of country: united STATES

English, Spanish

1. Print languages of a country

2. Compare languages of 2 countries

3. Quit

Enter your choice: 2

Compare languages

Name of first country: morcco

Morcco doesn't exist in database

Name of first country: Morocco re-prompt for invalid name only

Name of second country: united state

United State doesn't exist in database

Name of second country: united states re-prompt for invalid name only

No common language no common language

All languages: Arabic, Berber dialects, English, French, Spanish alphabetical order

1. Print languages of a country

2. Compare languages of 2 countries

3. Quit

Enter your choice: 2

Compare languages

Name of first country: switzerland

Name of second country: italy

Common language: French, German, Italian multiple common languages

All languages: French, German, Italian, Romansch, Slovene

1. Print languages of a country

2. Compare languages of 2 countries

3. Quit

Enter your choice: quit

1-3 only

1. Print languages of a country

2. Compare languages of 2 countries

3. Quit

Enter your choice: 3