Word Translator

In this activity, you will create a word translator that will translate a word in Language A to Language B where A and B are given as input to the program. For simplicity, there are only three languages: “english”, “hindi” and “punjabi” [no testcase will have a language other than these three languages, you may hardcode this as well] and one word in any of these languages can be translated to only one word in any other language. Only the words present in the files are considered to be in the vocabulary of the language. Also note that all words will be in lower case. You are given a file ‘translations.csv’ (comma separated values), which has 4 columns. Column-1 has the name of Language-1, Column-2 has word in Language-1, Column-3 has the name of Language-2, Column-4 has word in Language-2.

For example: hindi,aasmaan,punjabi,ambar means that aasmaan in hindi translates to ambar in punjabi.

LabDirectory contains the following files:

- language.py: This is the python script that you need to complete.

- testcases: This folder contains the testcases for the python script.

- translations.csv: This file contains the translation data.

- problem.pdf: The pdf file you are currently reading

Tasks:

1. Your python script will accept command line arguments and based on the query type provided as first argument to the script, interpret other command line arguments, perform operations and finally print the result on terminal.

2. There are 3 types of queries that you need to handle (language will always be one of english, hindi or punjabi):

* Query 1: Given a language, print its vocabulary i.e all distinct words that appear in translations.csv belonging to the given language in the form of a list sorted lexicographically in reverse order.

For example: ```python3 language.py 1 english``` gets you all english words present in translations.csv.

* Query 2: Given two languages A and B, print all translations available from Language A to Language B in the form of a list of tuples sorted lexicographically based on first element of tuple where the first element of each tuple is a word from Language A and the second element of the tuple is the corresponding translation in Language B.   
  For example: ```python3 language.py 2 hindi punjabi``` gets you all hindi to punjabi translations that can be interpreted from translations.csv.
* Query 3: Given two languages A and B and a word w in Language A, print the translation of word w in Language B. If the translation is not available, print "UNK" without quotes.

For example: ```python3 language.py 3 punjabi hindi ambar``` gets you a hindi translation of punjabi word “ambar” that can be interpreted from translations.csv.

Important Points:

- It might be possible that there is an indirect translation available from Language A to Language B via Language C (check Example given below for clarity). There will be “some” testcases that will particularly check if your program handles such cases and they have a weightage of 2 marks in hidden testcases.

- So if you don’t handle indirect translation, then you will get a maximum of 4 marks in hidden testcases.

- Please run the autograder script to check if your program passes on public testcases.

Marks distribution (out of 6):

- 1.5 marks for “query type 1”, 2 marks for “query type 2” and 2.5 marks for “query type 3”. All these testcases will be according to the above mentioned format of queries with no errors at all.

Submission:

- You need to submit the python script (language.py) only.

DISCLAIMER:

- Marks provided by the autograder are provisional. Final marks will be awarded after evaluation on hidden testcases. So you should not hardcode the answers based on the provided testcases. Try to write your own testcases to test the correctness of the script.

EXAMPLE:

Suppose the file ‘translations.csv’ is as follows:

punjabi,bhain,english,sister

hindi,bhagwan,punjabi,rab

english,mother,hindi,maa  
english,god,punjabi,rab

Test Case 1

$ python3 language.py 1 punjabi

Output:

[‘rab’, ‘bhain’]

Explanation:

* Out of 8 words, 3 are punjabi words and out of them only 2 are distinct. Also in reverse lexicographical order ‘rab’ comes before ‘bhain’.

Test Case 2

$ python3 language.py 2 hindi english

Output:

[('bhagwan', 'god'), ('maa', 'mother')]

Explanation:

* There is an indirect translation of ‘bhagwan’ from hindi to english, bhagwan (hindi) -> rab (punjabi) -> god (english). There is a direct translation of ‘maa’ from hindi to english. Also ‘bhagwan’ comes before ‘maa’ in lexicographical order.

Test Case 3

$ python3 language.py 3 english hindi god

Output:

bhagwan

Explanation:

* There is an indirect translation of ‘god’ from english to hindi, god (english) -> rab (punjabi) -> bhagwan (hindi).