DSCI 510  
Spring 2021

Homework #2 (100 points)

**DUE Friday, November 5, 2021, at 11:59PM Pacific Time on Blackboard**

Please note that this assignment is to be turned in via python script files **not** Jupyter notebooks. You are free to develop and debug the code in notebooks if you wish, but your final submission deliverable must be python script files.

The assignment should be delivered in a zip file to be called:

[your\_firstname]\_ [your\_lastname]\_homework2.zip.

(For example, I would turn in a file called “Yigal\_ Arens\_homework2.zip”.)

In this zip file should contain 4 files, named

1\_firstname\_lastname.py

2\_firstname\_lastname.py

and so on.

(In my case, 1\_Yigal\_Arens.py, 2\_Yigal\_Arens.py, etc.)

You may lose points if you do not follow this submission procedure.

**Question 1 (15 points)**

Write a python function that creates a copy of an input list and converts all negative numbers in the input list to their absolute values in the returned output list. If an item in the input list is a non-numeric value, the function should not include it in the output list, unless it can be converted to a numeric value. As an example, if the input is [‘DSCI-510’, -1, 0.1, 2, ‘US’, 3, ‘-3’] the output should be [1, 0.1, 2, 3, 3].

Note that the last element in the list above is a string, but is still converted to a number.

Call your function with a number of examples in the program you submit.

Hint: Use exception handling when appropriate, and remember to limit the code in the *except* clause.

**Question 2 (30 points)**

Suppose you have a table represented in a CSV file. Each element of the first row represents the header/name of a column.

Write a Python function ​**read\_csv(filename)** that reads in the CSV file (do this by writing your own code, without using libraries). For each element in the header (the first line of the CSV), make a key in a dictionary. Then, for each subsequent row of the file, add each cell to a list that is the value of the keys you defined for the headers. The value the function returns should be the dictionary, but it should also print it once you’ve read the whole file in. Make sure that the list elements are appended in row order.

Example.csv

|  |  |  |
| --- | --- | --- |
| key1 | key2 | key3 |
| key1’s value 1 | key2’s value 1 | key3’s value 1 |
| key1’s value 2 | key2’s value 2 | key3’s value 2 |

Result for Example.csv:

{key1 : [key1’s value 1, key1’s value 2], key2 : [key2’s value 1, key2’s value 2], key3 : [key3’s value 1, key3’s value 2]}

**Question 3 (25 points)**

Write a separate program that uses your function above to read in the table, and then computes the following transformation of your dictionary: The keys of the new dictionary you create should be the *last* elements of each column. The values of the new dictionary are lists containing the rest of the column that each new key belonged to in the original dictionary.

So, for the above example, the final dictionary would be:

{key 1’s value 2: [key 1, key 1’s value 1], key 2’s value 2: [key 2, key 2’s value 1], key 3’s value 2: [key 3, key 3’s value 1]}

Assume that the last values for all columns are unique with respect to each other.

These lists can be constructed in any order, but be sure not to include the new key itself in the value that is associated with it.

**Question 4 (30 points)**

Write a program that creates the transpose of a given CSV file's contents. In other words, if your program reads in a file that looks like:

abc,def,ghi

123,345,567

do,re,mi

x,y,z

The program should write a file that looks like:

abc,123,do,x

def,345,re,y

ghi,567,mi,z

You can use CSV parsing libraries to read and write your CSV file, but you may *not* use zip or any other variants. Read the file name your program reads and writes from the *command line*. Your program should be executed as (for example):

$ python 4\_*Yigal\_Arens*.py in\_file.csv out.csv

There are numerous tutorials on command line processing on the web (e.g. <https://pythonprogramminglanguage.com/command-line-arguments/>). You don’t have to use a fancy library, sys.argv will work.

NOTE: There are corpuses of common solutions to these problems (from google, stackoverflow, previous semesters, etc.). If your solution is too similar to some other solution (i.e. if you cheat), you will get a zero for this assignment and be subject to potential USC sanctions. Please don’t cheat.