CS 1301 – Homework 9

Extra Credit

Due Monday, December 5

File to submit to T-Square: HW9.py

You should work individually on this assignment. You may collaborate with other students in this class. Collaboration means talking through problems, assisting with debugging, explaining a concept, etc. Students may only collaborate with fellow students currently taking CS 1301, the TA's and the lecturer. You should not exchange code or write code for others. For individual assignments, each student must turn in a unique program. Your submission must not be substantially similar to another student's submission. **Collaboration at a reasonable level will not result in substantially similar code.**

For Help:

- TA Helpdesk – Schedule posted on class website.
- Email TA's or use Piazza Forums

Notes:

- **Do not forget to include the required collaboration statement** (outlined on the syllabus).
- **Do not wait until last minute** to do this assignment in case you run into problems.
- **Read the entire specifications document before starting this assignment**

Function Name: abbreviator (10 points)

Description: Write a function called abbreviator that takes in one string as a parameter. The function should return a new string. The new string should only contain the first letter in every word that is capital or any lone number in the string. Make sure that your returned string does not contain any special characters such as periods, semicolons, apostrophe's, spaces, etc. (Hint: the isalpha() and isupper() functions may help)

Parameter(s):

1. Some string of characters

Return Value: Some string of capital letters and/or numbers

Example(s):

```
>>> abbreviator("Chet just stole my girl again!")
'C'

>>> abbreviator("The Life of Pablo")
'TLP'

>>> abbreviator("30 Rock")
'30R'
```

Function Name: union (10 points)

Description: Define a function called union that takes in two lists that contain numbers – and returns the union of the two lists. The function should not change either of the input lists. The elements in the returned list must be in ascending order, and there should be no repeats in the returned list. Note: Numbers includes both floats and integers, although our example below only shows integers. (Hint: The sort() function may help)

Parameter(s):

1. An unsorted list of numbers
2. Another unsorted list of numbers

Return: A list of sorted numbers

Example(s):

```
>>> a = [1,3,2]
>>> b = [4,2,5]
>>> c = union(a,b)
>>> print(c) [1,2,3,4,5]
```

Function Name: tupleMagic (15 points)

Description: Write a function called tupleMagic that takes in a list of tuples. Each tuple contains some numbers. Without creating a new list, change each tuple to have the average of all the numbers in it at the beginning of the tuple. This function shouldn't return anything.

Parameter(s):

1. A list of tuples

Return: None

Example(s):

```
>>> aList = [(1,1,1), (1,2,4,5,6), (2,1,3)]
>>> tupleMagic(aList)
>>> print(aList)
[(1.0, 1, 1, 1), (3.6, 1, 2, 4, 5, 6), (2.0, 2, 1, 3)]
```

Function Name: reverseMultiTable (10 points)

Description: Write a function that takes in the number of the times table (up to 9). And print a reverse
multiplication table. DO NOT HARD CODE THE PRINTOUTS. Using a nested loop it's not the only way to do it but definitively the easiest. Check the test cases for the EXACT format on how to print the table. You must follow this format all the columns and rows must be indented equally.

Parameter(s):

1. Some integer from 0 to 9

Return: None

Example:

```
>>> reverseMultiTable(5)

25   20   15   10   5

20   16   12   8    4

15   12   9    6    3

10   8    6    4    2

5    4    3    2    1
```

```
>>> reverseMultiTable(7)

49   42   35   28   21   14   7

42   36   30   24   18   12   6

35   30   25   20   15   10   5

28   24   20   16   12   8    4

21   18   15   12   9    6    3

14   12   10   8    6    4    2

7    6    5    4    3    2    1
```

Function Name: charCount (15 points)

Description: Write a function called charCount that takes in a string and returns a dictionary where the key is a character in the string and the value is the amount of times that character has shown up the in the string. The key for a space character should be None, but using None directly as a key is prohibited (e.g. aDict[None] is not allowed). For the first space, you should print "First Space", but for every space after you should print "ANOTHA ONE".

Parameter(s):

1. Some string of characters

Return: A dictionary containing the character count of the string

Examples:

```
>>> charCount("")
{}

>>> charCount("hello")
{'h': 1, 'o': 1, 'e': 1, 'l': 2}

>>> charCount("Yung Coco Butter")
First Space
ANOTHA ONE
{'g': 1, 'r': 1, 't': 2, 'c': 1, 'e': 1, 'C': 1, 'n': 1,
'u': 2, 'B': 1, 'Y': 1, None: 2, 'o': 2}
```

Function Name: **chetify** (20 points)

Description: Write a function called chetify that takes in a file name and returns nothing. This function should go through a text file and change every "Young" to "Yung", "you" to "we", "yea" to "yee", and "like" to "about". (Hint: use a dictionary)

Parameter(s):

1. Some string that represents a file name

Return: None

Example:

```
Yo what's up Young Margs.
Nothing just chillin', did you checkout Paper Bois new
mixtape?
Yea, you like it?
It was pretty good, I'd give it a solid 10/10.

>>> chetify("sample.txt")

Yo what's up Yung Margs.
Nothing just chillin', did we checkout Paper Bois new
mixtape?
Yea, we about it?
It was pretty good, I'd give it a solid 10/10.
```

(20pts) Write what you think the big-O of your each of your functions is and give your reasoning on why. Points will be awarded more based on your explanation rather than correctness of your answer.

Rubric:

abbreviator……10

- Correct function header……...2
- Returns a string……....2
- Returns the correct string……6

union……10

- Correct function header……...2
- Returns a list……....2
- Returns the correct list……6

tupleMagic……...15

- Correct function header……...2
- Iterates through the list……2
- Does not create a new list……...……5
- Tuples are replaced in the list……..1
- Tuples are changed with the correct averages…...……5

reverseMultiTable……10

- Correct function header….. ……2
- Correctly calculates numbers (e.g. using a nested loop)…….....2
- Print the exact format of table.……..…….……6

charCount……..15

- Correct function header….. ……2
- Doesn't use None as a key directly……5
- Prints the right amount of times and prints correct statements……...3
- Returns correct dictionary……...5

chetify…….……....20

- Correct function header….. ……2
- Opens files correctly…….…..……2
- Iterates through file……..……4
- Writes to a new file……..2
- Output is correct……..……8
- Closes both files……...2

Big-O……20

- Full credit is given as long as a reasonable explanation is given for each function
- -3 for each function that does not have a reasonable explanation