CS/CYCS1110 (Python) –Fall 2021
Programming Project #3

Due Date (A Two-week Project) 100 points	
Tueday Labs	**10/?/21 @ 11:59 pm**
Wednesday Labs	**10/?/21 @ 11:59 pm**

Project Objectives

- Use loops
- Use functions, parameter-passing and modules for a very modular program
- Use incremental development to write and test your program
- NO GLOBAL VARIABLES ALLOWED

Project Overview & Structure

This Project contains 2 separate .py files: the main program and one module.

1. **frequencyCalculate.py**: contains 8 functions (& the call to main to start program execution).
    a. **main()**: This function asks the user to input a positive integer, and then read that integer m. Use input() built-in function to read this integer. Note: validate the integer number; The program accepts only positive numbers.
    b. **userString():** This function asks the user to input m strings. As the strings are read, they should be concatenated into a single string st. After reading the m strings and forming the single string st, it should return the concatenated st to the main() function. This function has one parameter, an integer to receive m.
    c. **showString():** This function prints the concatenated string received from the main() function. This function has one parameter, the concatenated st.
    d. **convertStringLower()**: This function coverts the string object to lower case. it should return the lower case string to the main() function. This function has one parameter, the concatenated st.
    e. **stringToList()**: This function converts the lower case String object to a list of characters. (All letters will be lower case). Then, this function prints the character list just created and returns the character list to the main() function. This function has one parameter, the concatenated st.
    f. **compressList():**.Creates a new compressed list, and copies only the letters of the original compressed list into the new compressed list. This function prints compressed list and returns the new list to main() function. This function has one parameter, the list of characters._
    g. **frequencyCount():** In main() declare two lists: an integer list of length 26 (set all entries in the integer list to zero), and a character list (which contains 26 lower case letters). Call the frequencyCount() with three arguments, the integer list and the character list(those will declare in the main()), and the compressed list. This function counts the number of times each letter occurs. The integer list will contain a frequency distribution for the letters in the list of lowercase letters. This function returns the integer list to main() function.

h. **showFrequency():** This function receives the frequency distribution list and prints each letter on a new line followed by the number of stars equal to the integer value in that list element. This must be neatly aligned.

2. **testModule.py** - contains 1 function:
    a. **Checkplaindrome():** This function tests the compressed list to see if it forms a palindrome, and returns a Boolean value: **true** if it does and **false** otherwise, then in main() displays a message indicating whether or not the list is a palindrome based on the returned value. It has one parameter, the compressed list.

Deliverables

The deliverable for this project is the following file:

proj03_LastName

A zip file that contains 2 separate .py files. Be sure to use the specified file(s) name(s) and to submit it for grading via eLearning before the project deadline.
Provides this information in frequencyCalculate.py:
Project No.:
Author:
Description:

Sample Output (use EXACT format – and NO HARDCODING of the data itself)

```
Enter the number of strings you want to read:
3
Enter string 1:
Listen my children and
Enter string 2:
you shall hear of the midnight
Enter string 3:
ride of PAUL REVERE!!
Listen my children and you shall hear of the midnight ride of PAUL REVERE!!
listen my children and you shall hear of the midnight ride of paul revere!!
listenmychildrenandyoushallhearofthemidnightrideofpaulrevere
These letters do not form a palindrome.

Following is the frequency distribution for the letters in the list.

a  ****
b
c  *
d  ****
e  ********
f  **
g  *
h  *****
i  *****
j
k
l  *****
m  **
n  ****
```

o ***
p *
q
r *****
s **
t ***
u **
v *
w
x
y **
z