HOMEWORK 3
(submit as a single .py file)

1) Construct a Tuple with a single value. (5pts)
A tuple named : Student Name with the value ”Binil”
Print the value to the console and check its type.
Check if the returned Type is a Tuple. If not, how would you fix it?

2) Write a List comprehension that does the following task. (10pts)
Given a list of strings of movies
aList = [“Batman”, “Spiderman”, “Avengers”, “Matrix”]
Create a new list that contains the total number of characters in each word within the list. The output should look like.

Output
charCountList = [6, 9, 8, 6]

3) Take two lists, say for example the two lists below: (10pts)
a = [1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89]
b = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 78, 21]
c = [3, 1, 5, 6, 3, 2, 4, 78, 21, 33]
and write a program that returns a list that contains only the elements that are NOT common between the lists (without duplicates). Make sure your code snippet works on lists of different sizes.

4) (20pts) Given the following paragraph edited from a recent new story:
“British researchers have identified 50 new planets using artificial intelligence and machine learning, marking a technological breakthrough in astronomy. Astronomers and computer scientists from the University of Warwick built a machine learning algorithm to dig through old NASA data containing thousands of potential planet candidates. It is not always clear, however, which of these candidates are genuine. When scientists search for exoplanets (planets outside our solar system), they look for dips in light that indicate a planet passing between the telescope and their star. But these dips could also be caused by other factors, like background interference or even errors in the camera. The researchers used heavy GPU computing to analyze more 10PBs of data that spanned near 4 months of analysis”

   A. Create a list that contains individual words in the paragraph. Strip away punctuation marks.

   B. Clean the list to create a new list called “CleanedList” that has the following attributes: 1) Each entry is unique; 2) Must not contain the following stopwords = [‘and’,’this’,’is’,’are’,’its’,’when’,’like’,’or’,’in’,’on’,’because’,’But’]

   C. Create a second data structure that reports for each unique word from CleanedList, the frequency with which it appears in the paragraph. For example, scientists appear 2 times, whereas months appear 1 time.

5) (10pts) A user has two standard dices. Each dice has values from 1 through 6. The user throws each set of 2 dices 20 times. Simulate this action using a Python script. For the 20 throws, print the total sum of the scores obtained on the dices. For Loops are required.

6) (25pts) Let’s play Rock, Paper and Scissors. Devise a program that lets the user play the game with the Computer for a specified number of attempts. For each attempt, the user inputs his/her hand and the computer randomly picks one from the list. If you know the game – Rock beats Scissors; Paper beats Rock; Scissors beats Paper. The algorithm should also track the score for the user and the Computer until the end of the game.

A sample I/O is shown as follows:

Input:
Welcome to the Game. My name is “RoPaSc” Gamer
Enter number of attempts: 5
Enter your name: Binil

Input:
Attempt 1: Show your Hand: Rock

Output
Sorry, you lost. Computer picked Paper.
Score: User: 0; Computer: 1

Input
Attempt 2: Show your Hand: Paper
Output
Sorry, you lost: Computer picked Rock
Score: User: 0; Computer: 2

..
..

(After the 5 attempts are over, the algorithm should display who won the game)

Output:
Congratulations Computer, you won the game! Sorry, Binil, you lost.

Final Score: Binil - 2; Computer - 3

7) (20pts) Evan is fond of playing with strings. He discovered his own kind of string called Evan’s String Test. If the string is split in the middle, the two halves will have the same exact characters and in the same frequency. If the number of characters is even, such as ‘papa’ and ‘mama’, there is an exact split and each half passes the Evan’s String Test. If the string has odd number of characters such as ‘pappa’ and ‘mamma’, this also passes the Evan’s string test, since the middle character – p and m are omitted, and the two halves are exactly the same.

Note that strings like ‘lala’, ‘eegege’, ‘pqooqp’, ‘acdca’ also pass Evan’s String Test. Write a python function to return a list of ‘Pass’ or ‘Fail’ for each input string. (15 points)

testWList=['papa', 'paapaaa', 'Mama', 'eefffe', 'danda', 'paattaa', 'lala', 'eefefe', 'abdba', 'RoaaoR', 'Ululu', 'Ratatat']

The output should be as follows for the given list in the question = ['Pass', 'Fail', 'Pass', 'Fail', 'Pass', 'Fail', 'Pass', 'Pass', 'Pass', 'Pass', 'Pass', 'Fail']