CSC/DCSCI 2720: Data Structures
Lab 4

Instructor: Shiraj Pokharel

Due : 48 hours after release
Late Submission deadline (with 25% penalty) 24 hours after due date

Answer the below questions. You must submit your responses as a SINGLE Jupyter Notebook, where each program is put in separate Jupyter Notebook cells within that SINGLE Jupyter Notebook. Do NOT submit Colab links. Failure to comply with this simple requirement will result in a score of Zero. Please, be careful not to be assigned a Zero score this way.

Few Rules to be followed, else will receive a score of ZERO

(1) Your submissions will work exactly as required.

(2) Your files shall not be incomplete or worse corrupted such that the file does not compile at all. Make sure you submit a file that compiles.

(3) Your submission will show an output. Should you recive a Zero for no output shown do not bother to email me with "but the logic is perfect" !

Note that your program's output must **exactly** match the specs(design , style) given here for each problem to pass the instructor's test cases .
Design refers to how well your code is written (i.e. is it clear, efficient, and elegant), while *Style* refers to the readability of your code (commented, correct indentation, good variable names).

[100 points] Design and code in Python, a solution such that given two user input strings s and t, return true if t is an anagram of s, and false otherwise.

Definition : An Anagram is a word or phrase formed by rearranging the letters of a different word or phrase, typically using all the original letters exactly once.
Example 1:
Input Strings: s = "anagram", t = "nagaram"
Output: true
Example 2:
Input: s = "rat", t = "car"

Output: false

Solution Approaches :
To examine if String t is a rearrangement of String s, we can count occurrences of each letter in the two strings and compare them using an Array. Since both String s and String t only contain alphabets, an array of size 26 will suffice.

The question then automatically comes-up : Do we need two counters for comparison? Actually no, because we can increment the count for each letter in String s and decrement the count for each letter in String t, and then check if the count for every character is zero.

A second approach to the solution is also available. We could first increment the counter for String s, then decrement the counter for String t. If at any point the counter drops below zero, we know that String t contains an extra letter not in String s and return false immediately.