BIOS 6642 – Take-Home Assignment Two

Due at 11:59PM (Mountain Time), Friday, March 19, 2021

Please submit your source codes (.py or .ipynb file) and screenshots of the output of your codes. Your codes should be properly documented or commented.

- Q1 (30%). Please write a function that takes a non-empty string, e.g., *my_str*, as the input parameter. The function needs to find the length of a substring, *sub_str*, in *my_str* such that (1) *sub_str* does not have repeating characters, and (2) the length of *sub_str* is maximum. The function needs to print the length of *sub_str (*do not need to return *the sub_str)*.

    - A substring is a contiguous sequence of characters within a string. For example, '*py*' is a substring of '*python*', but '*pt*' is not.
    - If *my_str* = '*aaa*', then the function should print 1, because *sub_str* would be '*a*'.
    - If *my_str* = '*aa*b*', then the function should print 3, because *sub_str* would be '*a*b*'.
    - If *my_str* = '*asdfsab*', then the function should print 5, because *sub_str* would be '*dfsab*'.
    - Note that uppercase and lowercase letters are considered different letters, e.g., '*A*' and '*a*' are two different characters.
    - In addition to the function, please write some code to test the function.

- Q2 (35%). Suppose you want to move an object from the origin to a position that has a distance, ***x*** units (a positive integer), from the origin. Each time you can move only 1 or 2 units. Please write a program to print in how many different ways you can move the object to the position. The distance, ***x***, is provided during runtime. For example,

    - If ***x*** = *2*, then your program should print 2, because there are 2 different ways: (1) 1 + 1 and (2) 2
    - If ***x*** = *4*, then your program should print 5, because there are 5 different ways: (1) 1 + 1 + 1 + 1, (2) 1 + 1 + 2, (3) 1 + 2+ 1, (4) 2 + 1 + 1, and (5) 2 + 2

- Q3 (35%). Please write a function that takes a non-empty string as input (assume the string is called *my_str*). For this question, the string consists of only English letters, and there is no difference between uppercase and lowercase letters, e.g., 'A' and 'a' are considered the same. The function repeatedly deletes two adjacent and identical letters. The function returns the final string after the deletion (the letters in the returned final string can be all lowercase). For example,

- If *my_str* = '*xyzzya*', the function should return '*xa*'.
- If *my_str* = '*xyx*', the function should return '*xyx*'.
- If *my_str* = '*xxx*', the function should return '*x*'.
- If my_str = '*xXx*', the function should return '*x*'.
- If *my_str* = '*xxxYyxZ*', the function should return '*z*'.
- In addition to the function, please write some code to test the function.