CS19003: Programming & Data Structure Lab, Section 15
Autumn 2020-21
Lab Test 1, January 14, 2021

PART – 2

Time: 10-30 am to 11-40 am **Total Marks: 10**

Instructions (Read carefully)

1. **Your C programs must first contain comments with your name, roll no., and Labtest no. (=1), as done in class.**
2. **Name your C file LT1_2_<your roll no>.c for (For example LT1_2_20ME30006.c)**
3. **Submit through the links (Intermediate and Final) for Part 2 in moodle. MAKE SURE TO VERIFY YOUR SUBMISSION after the final submission.**
4. **All instructions given in the slides of last class (January 7) to be followed strictly**

---

1. Write a C program that does the following **exactly in this order**:
    - Reads a string *S* from the keyboard using %s in scanf. Assume that the string will contain less than 100 characters, and will have all lowercase alphabets only (you do not need to check this, just enter strings with all lowercase alphabet).
    - Computes the length of the string *S* and stores it in a variable named *len*
    - Reads two integers from the keyboard in two variables, *low* and *high*. (Input them **exactly in this order**, i.e., *low* first and then *high*).
    - If either the value of *low* is < 0 **or** the value of *high* is ≥ *len*, print a message "Values entered are incorrect" and return (nothing else to be done). Otherwise go to the next step.
    - Count the number of occurrences of the characters 'a', 'b', 'c', 'd', ...,'x', 'y', 'z' (i.e., all lowercase alphabets) between index *low* to index *high* of the string S, both included (i.e., how many times the character occurs between *S*[*low*] and *S*[*high*]). Store the counts in another integer array X of size 26 (one each for the 26 alphabets). So the number of times 'a' occurs should be stored in X[0], the number of times 'b' occurs should be stored in X[1], the number of times 'c' occurs should be stored in X[2], …the number of times 'z' occurs is stored in X[25].
        - **You must use two loops, one nested inside another, for this part. Do not use a one-loop solution even if you can do it. 50% penalty if you do so.**
        - **You cannot use any ASCII code values directly in your code. 50% penalty if you do so.**
        - Use the fact that ascii codes of the lowercase alphabets are contiguous.
    - Print each alphabet and its count from the array X after counting of all characters are over. You should print only the characters that have a non-zero count in the format like "a = 3, b = 4, x = 1" etc. (see example below for exact format).

Example 1:

S = "assassination", *low* = 3, *high* = 17

Your program should print

Values entered are not correct

as the length of *S* is 13 and *high* = 17 ≥ 13.

Example 2:

S = "assassination", *low* = 2, *high* = 10

Your program should print

***a* = 2, *i* = 2, *n* = 1, *s* = 3, *t* = 1**

as 'a' occurs 2 times, 'i' occurs 2 times, 'n' occurs 1 time, 's' occurs 3 times, and 't' occurs 1 time in the part of the string from *S*[2] to *S*[10], i.e., "sassinati"