National University Of Computer and Emerging Science Lahore Campus

Programming Fundamental

Section: (BSCS-1C,1E & 1J)

Due Date: **Sat, Nov 25th by 11:55pm.**

Dear students we will be using auto-grading tools, so failure to submit according to the below format would result in zero marks in the relevant evaluation instrument.

1. Make a separate cpp file e.g. you have to make ROLL-NUM_SECTION.cpp (23L-0001_C.cpp) and so on. File that you submit must contain your name, student-id, and assignment on top of the file in comments.
2. Submit the .cpp file on Google Classroom within the deadline.
3. Submission other than Google classroom (e.g. email etc.) will not be accepted.
4. The student is solely responsible to check the final cpp file for issues like corrupt file, virus in the file, mistakenly exe sent. If we cannot download the file from Google classroom due to any reason it will lead to zero marks in the assignment.
5. Displayed output should be well mannered and well presented. Use appropriate comments and indentation in your source code. Five Bonus marks will be awarded to well commented/indented code (for all questions).
6. Total Marks: 100
7. If there is a syntax error in code, zero marks will be awarded in that part of assignment.
8. Your code must be generic

Problems

In this assignment, your task is to write a program that determines if word pairs given in an input are **super anagrams** and if each word is an **imperfect palindrome**.

Super-anagrams are words;

1. that consist of the same letters.
2. that the accumulated frequency difference of the letters in the pair of words differs atmost 2.

For example, “stake” and “takes” uses each letter ‘a’, ‘e’, ‘k’, ‘s’, and ‘t’ once. So they are super-anagrams. “Pumpkin” and “umpire” uses different letters (e.g. with and without ‘k’), so they are not super-anagrams. Words “umpiire” and “umpppire” have the same letters (‘u’, ‘m’, ‘p’, ‘I’, ‘r’, ‘e’), but are not super-anagrams since “umpiire” has two fewer p’s and one more ‘I’: the accumulated frequency difference is 3.

A **palindrome** is a word that is the same reading forward or backward. Examples include “a”, “aba” and “aabb baa”. An imperfect palindrome is not a palindrome, but when reading an imperfect palindrome forward and backward, the number of characters that differ is no more then 2. For example, “abc”, “abca”, and “aaabbaab” are imperfect palindromes. “aba”, “abcd”, and “aaabbabb” are not impact palindromes.

The input will contain many lines (from the input file), each line containing a pair of words to be analyzed. The words only use low-case letters and there is exactly one space between the pair of words. Each word has no more than 20 letters. For each line, your program should output three yes/no answers: the first answers whether the first word is an imperfect palindrome, the second answers whether the second word is an imperfect palindrome, the third answers whether the words are super-anagrams. For example, for the following input:

read dare
stake takes
tofu tofu
aabbba aabbaa
one two

Your program should output:

no no yes
no no yes
no no yes
yes no yes
yes yes no

Honor Policy

This task is a learning opportunity that will be evaluated based on your ability. **Plagiarized reports or code will get a zero**. If in doubt, ask the course instructor.

Grading policy

1. Program with no compiler error (20 points)
2. The program gives the correct number and format of the yes/no answers (20 points)
3. Correctly check imperfect palindromes (30 points)
4. Correctly check super-anagrams (30 points)