CHRISTOPHER NEWPORT UNIVERSITY
DEPARTMENT OF PHYSICS, COMPUTER SCIENCE & ENGINEERING
CPSC 360 - CONCEPTS IN PROGRAMMING LANGUAGES

* Assignment 2: C++: Hyphenation *

Instructor: Dr. Roberto A. Flores

Goal

Use C++ for problem solving.

Description

Word processors can break a word across lines using hyphenation, a technique that requires knowledge of where syllables in a word are divided.

Write a program receiving a string with words (where a word is a sequence of non-whitespace characters) and returns another string with these words hyphenated as defined by the following 4 rules:

1. If a **vowel-consonant-consonant-vowel** pattern is found, hyphenate between the two consonants. Letters 'a', 'e', 'i', 'o', 'u', and 'y' are vowels; all other characters are consonants.
2. If a **vowel-consonant-vowel** pattern is found, hyphenate before the consonant unless the second vowel is an 'e' and occurs at the end of the word.
3. The following character sequences are never divided by hyphens: "qu", "tr", "br", "str", "st", "sl", "bl", "cr", "ph", "ch". For the purposes of rules 1 and 2, each of these are single consonants.
4. Upper and lower-case distinctions are ignored for the purpose of applying the above rules. However, the case in the input must be preserved in the output.

Implementation

Write a C++ program implementing the function below (this exact definition must be used).

```
char* process(const char* input)
```

This function receives a C-style string named *input* containing a list of words and returns another C-style string with all pertinent words hyphenated. For example, given the input "Lorem ipsum dolor sit amet" the function returns "Lo-rem ip-sum do-lor sit a-met".

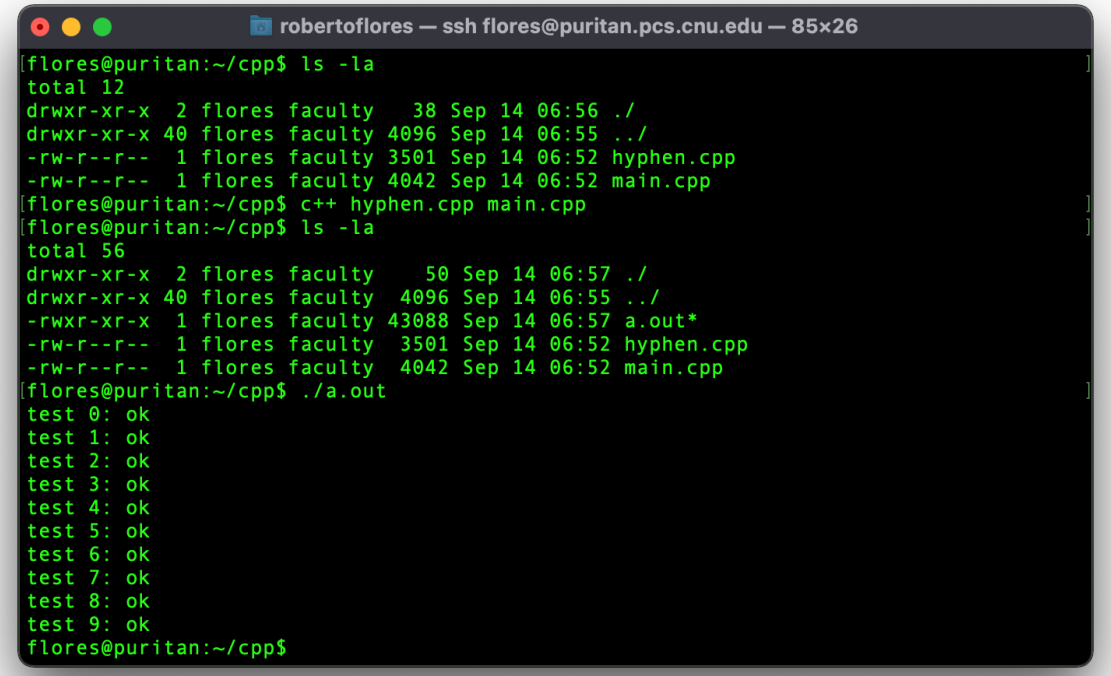
Initial Files

The initial C++ files *hyphen.cpp* and *main.cpp* can be downloaded from Scholar.

- The file *hyphen.cpp* has an empty *process* function, where your implementation begins (you can create other functions if needed).
- The file *main.cpp* has a *main* function with test cases invoking *process*. This file is complete and should not be modified.

Grading

As shown below, your program **must** compile (using *c++*) and run (using *a.out*) in the **PCSE UNIX servers**.



Programs shall run without segmentation faults, bus errors, assertion failures or any other malfunction.

Your program will be tested according to the functionality specified in this description and will be graded according to the number of test cases it passes. Answers must not be hard-coded.

Each of the tests is worth and equal amount of points, up to a maximum of 100.

Groups

This assignment is done individually or in teams of 2 people as assigned by the instructor.

No code (partially or totally) should be **taken** or **disclosed** to/from other people (except your partner, if you have one), including online resources. If in doubt, read the “Honor Code” section in the syllabus and contact your instructor.

Submission

Source code (i.e., your **.cpp* files only) must be submitted through Scholar by its due date.

Partners will submit one copy only.

•●•