COP3503 Computer Science 2
Dr. Andrew Steinberg
Fall 2023

Programming Assignment 4
Spell Checker
Due: 11/17/2023 at 11:59pm

Objective: Students will apply concepts of dynamic programming and the Sequence Alignment problem.

Assignment Description: You are currently using a computer that doesn't use a spellchecker and have decided to create your own version using the sequence alignment algorithm you learned in class.

For this assignment, you must follow these **requirements**.

1. Create a class called `SequenceAlignment`. In this class, you will implement the dynamic programming algorithm.
2. The class constructor for `SequenceAlignment` takes two string objects.
3. In the driver file, you will notice the method "`computeAlignment`" is invoked. This is the method that will invoke the dynamic programming solution. Make sure the method name matches as the driver file in your implementation! Changing names will cause the driver file not to run or even compile which will result in a low score on the assignment!
    a. In order the compute the minimal alignment, you have decided on the following mismatch costs.
        i. $\delta = 2$
        ii. Same symbol costs is 0
        iii. Vowel and different vowel will cost 1
        iv. Constant and different constant will cost 1
        v. Vowel and constant will cost 3
    b. For this assignment, you only need to worry about lower case alphabet letters. You do not need to consider uppercase.
4. In the driver file, you will notice the method "`getAlignment`" is invoked. This is the method that will return both alignments as one string separated by a whitespace character (' '). Make sure the method name matches as the driver file in your implementation! Changing names will cause the driver file not to run or even compile which will result in a low score on the assignment!
5. Make all methods public and class attribute private. It's good practice!
6. You may create additional helper methods and attributes if needed as long as they are implemented and called in your solution file and NOT called from the driver file. Adding extra methods to call in the driver file will not match to what the graders will use evaluate your code. This will result in a low score with no change to be applied!
7. If your code does not compile or is stuck in an infinite loop, you will receive no credit on any of the rubric categories except for code style/comments, assignment directions, and code submission/header.

A driver file (`SequenceAlignmentDriver.java`) has been provided for you to show you how the methods are called along with 5 test cases.

What to submit: Submit a file called `SequenceAlignment.java` to webcourses. You are not required to submit the driver file as that will be provided for the graders to test your code. Please make sure the driver file provided works for your code. Any name changes may cause your program not to work when graded, which will result in a lower score on the assignment and would not be changed.

Important Note for running Eustis: Many of you are probably using IDEs like Netbeans and Eclipse to build your Java Solutions. Please note that some of these IDEs will automatically place your Java file in some sort of package. Please make sure your Java file is not defined in some package as this can result package private errors. Any such error that occurs during the grading will not be fixed and points will be deducted as such in accordance with the respective categories in the rubric. Also, DO NOT create a main method in your solution file!! This will result in your code not running properly with the driver file which will result in points being deducted from the respective categories.