

Programming Assignment 4 Spell Checker

Due: 12/1/2021 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 runner 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 runner file in your implementation! Changing names will cause the runner file not to run or even compile which will result in a low score on the assignment!
 - a. In order to 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 runner 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 runner file in your implementation! Changing names will cause the runner 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 runner file. Adding extra methods to call in the runner file will not match to what the graders will use to evaluate your code. This will result in a low score with no change to be applied!

A runner file (`SequenceAlignmentRunner.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 runner file as that will be provided for the graders to test your code. Please make sure the runner 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 you will need to separate the two files. In other words, do not put the runner file in the same package as `SequenceAlignment.java`. If you put both files in the same package, the command line will have issues with compiling. The workaround this is putting your `SequenceAlignment.java` file in a package called “sequencealignment” and have the runner file import it (see the runner file for clarification). **Please make sure to name the package sequencealignment. Check out the runner file!!!**