COP3503 Computer Science 2
Dr. Andrew Steinberg
Fall 2021

Programming Assignment 3
Secret Messages
Due: 11/14/2021 at 11:59pm

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

Assignment Description: You have intercepted a message that is riddled with random characters. You try really hard to decipher the message, but couldn't. Later you showed a computer science friend the message and your friend tells you they are using subsequences to send secret messages. You both learned that sending the message "g1o1o4dasdfhad5aaytwy" can translate into "good day". Remember subsequences is different than substrings as character positions don't have to be consecutive. In this assignment, you have intercepted the document and are going to decipher the message. In order to do this, you will need to apply the Longest Common Subsequence Algorithm discussed in class. You must apply the dynamic programming solution. Using another variation will result in losing points.

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

1. Create a class called `LCS`. In this class, you will implement the dynamic programming algorithm.
2. The class constructor for LCS takes two string objects.
3. In the runner file, you will notice the method "`lcsDynamicSol`" 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!
4. In the runner file, you will notice the method "`getLCS`" is invoked. This is the method that will access the subsequence computed by the dynamic programming solution. This is the Print-LCS algorithm we discussed in class. 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 evaluate your code. This will result in a low score with no change to be applied!

A runner file (`LCSRunner.java`) has been provided for you to show you how the methods are called along with 4 test cases. The text file is also provided for you that you will need to decipher using the LCS algorithm. The text file itself must be in the same directory as the runner file. The number of characters to be read from the file on each line is 1000.

COP3503 Computer Science 2
Dr. Andrew Steinberg
Fall 2021

What to submit: Submit a file called `LCS.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 `LCS.java`. If you put both files in the same package, the command line will have issues with compiling. The workaround this is putting your `LCS.java` file in a package called "`lcs`" and have the runner file import it (see the runner file for clarification). **Please make sure to name the package `lcs`. Check out the runner file!!!**