CSC 323 Algorithm Design and Analysis, Fall 2019
Instructor: Dr. Natarajan Meghanathan

Project 6: Dynamic Programming-based Solution for the Longest Common Subsequence Problem

Due: Oct. 29th, 2019: by 11.59 PM (in Canvas)

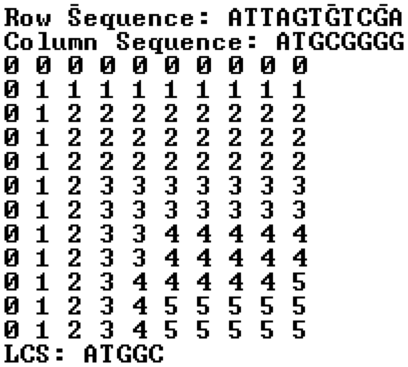
In this project, you will implement the dynamic programming-based solution to find the longest common subsequence (LCS) of two sequences.

Your inputs will be the two sequences (as Strings) and the outputs are the longest common subsequence (printed as a String) and the final matrix (printed as a two-dimensional array) depicting the length of the longest common subsequences (as shown in the slides) for all possible subsequences of the two input sequences.

The two input sequences to be used by each student are shown below. The LCS expected for the two sequences is also shown.

Student Name	Row Sequence	Column Sequence	LCS
Perry Butler	TCGCCTT	GGGGTAACT	TCT
Latamla Culley-Triggs	TAAAATCTAG	CTTGGATC	TATC
Justin Epps	GTGTGGAAAC	GCTTCTTTCT	GTTC
Kalil-Dan Ford	AGGACGGTGAA	AATTTTTA	AATA
Chawne Harris	CGGCCAGGCGAT	CGAGGTAAGTAG	CGAGGGA
Ashly Horner	GCTATTAT	ATAGAAATC	GAAT
Martice Jackson	TTCTGATGTT	TCGGGAT	TCGAT
Jorian Lenard	CAGATGTATCTG	GAGACAGGAT	CAGGAT
Damian Patterson	CTCAGGT	GTGAGGGGGA	TAGG
Brandon Redmond	GATTGCACTA	GTAGCAGT	GAGCAT
Daren Washington	GCTAAGC	AGTGCCG	GTGC
Alicia Wells	ATCACC	GCTCGATCTGCA	TCACC
Marcus Wynn	TTTTAATCCAGC	TGCAGAGAACTA	TAATA
	GAGTAAG	GCGACG	GGAG
	CCCCTATAGT	CTGACG	CTAG
	AGAGGC	CAATCGCAACGC	AGAGC
	TATCAA	TGGACTCCGCAC	TATCA

A sample output is shown below.



```
Row S̄equence: ATTAGTḠTCḠA
Column Sequence: ATGCGGGG
0 0 0 0 0 0 0 0 0 0
0 1 1 1 1 1 1 1 1 1
0 1 2 2 2 2 2 2 2 2
0 1 2 2 2 2 2 2 2 2
0 1 2 2 2 2 2 2 2 2
0 1 2 3 3 3 3 3 3 3
0 1 2 3 3 3 3 3 3 3
0 1 2 3 3 4 4 4 4 4
0 1 2 3 3 4 4 4 4 4
0 1 2 3 4 4 4 4 4 5
0 1 2 3 4 5 5 5 5 5
0 1 2 3 4 5 5 5 5 5
LCS: ATGGC
```

Submission (through Canvas):

1) The entire .cpp code file
2) A word or PDF document containing the following:
(i) Screenshot of the outputs showing the final dynamic programming table (of the lengths of the longest common subsequences of all possible subsequences of the two input strings) and the longest common subsequence
(ii) the final alignment (determined manually by working out the alignment based on the table printed in ii, and typed in the word document) of the two input strings and the gaps (-) that need to be introduced in order to facilitate an alignment that matches with the LCS obtained.