Bhavyansh Mishra

Data Structures and Algorithms II

Project 5

User’s Manual

Dynamic Programming

*(Longest Common Subsequence)*

*Instructions for Setup and Compilation*

1. Download and Unzip the submission from the eLearning dropbox onto a Linux box in the Multi-Platform Lab.
2. Change directory into the unzipped directory
3. The extracted/unzipped directory must include the following (at a minimum):
   1. Functional Decomposition document (in .docx format)
   2. User’s Manual (in .docx format)
   3. Makefile
   4. Main.c
   5. Similarity.c
   6. Lcs.c
   7. Similarity.h
   8. Lcs.h
   9. twoSequences.txt
   10. multipleSequences.txt
4. To compile and build the program, use the Makefile provided. Run the command “make” once inside the unzipped folder.

(*Note: This program has been tested on Ubuntu bash on several machines using make*)

**(Remember: twoSequences.txt and multipleSequences.txt must in the UNIX format with 1 Byte for newline character. The two files must be present in the same directory as the Makefile for proper functioning. To use a Windows generated .txt files, change the #define NL 1 to #define NL 2 to represent a newline using 2 Bytes as is done by Windows for CR and LF)**

1. To run the program, execute the command “./dp” to execute the executable named “dp” created in the same directory.

**User Input**

This program does not require any user input.

**Output**

All the output will go to the standard output, no relevant external files will be created during the runtime of the program. The output of the program must in two parts:

Part 1 and Part 2 as following:

PART 1

Reading local file: twoSequences.txt

String 1: chocolateclvpk

String 2: colldegeclavpelk

Length of LCS: 9

LCS: coleclvpk

PART 2

Reading local file: multipleSequences.txt

Performing similarity comparison on all unique pairings: multipleSequences.txt

1 2 3 4 5 6 7

1 - M D D D D D

2 - - M L L D D

3 - - - D M L D

4 - - - - D D L

5 - - - - - D D

6 - - - - - - D

7 - - - - - - -