```
/* CMPUT 201 Assignments                                                    */
/* Dues:                                                                    */
/* #1:    11:55pm, September 29, 2019;                                      */
/* #2:    11:55pm, November 3, 2019;                                        */
/* #3:    11:55pm, December 6, 2019                                         */
```

(Mandatory assignment cover-sheet; without it, your work will not be marked.)

Submitting student: _____

Collaborating classmates: _____

Other collaborators: _____

References (excluding textbook and lecture slides):_____

_____

_____

_____

Regardless of the collaboration method allowed, you must always properly acknowledge the sources you used and people you worked with. Your professor reserves the right to give you an exam (oral, written, or both) to determine the degree that you participated in the making of the deliverable, and how well you understand what was submitted. For example, you may be asked to explain any solution that was submitted and why you choose to write it that way. This may impact the mark that you receive for the deliverable.

So, whenever you submit a deliverable, especially if you collaborate, you should be prepared for an individual inspection/walkthrough in which you explain what every line of assignment does and why you choose to write it that way.

The following is from a sample input file named "`sequences.txt`" (also posted in eClass and Public Webpage under Week 1):

```
3110283025318292818318143
1311230172803163125184 1324
```

This file contains two sequences over the numerical digit alphabet $\Sigma = \{0, 1, 2, \ldots, 9\}$, of length 25 and 26 respectively. Let us use an array to represent such a sequence, and further use $S_1[25]$ and $S_2[26]$ to denote the above two sequences.

Given the sequence $S_1[25]$, one may delete any number of members from $S_1[25]$ (and then concatenate the remaining members in the original order) to achieve a *subsequence* of $S_1[25]$; for example, $T[10]$ ="3108025843" is a subsequence of $S_1[25]$. One sees that $T[10]$ is also a subsequence of $S_2[26]$, and in such a case $T[10]$ is called a *common subsequence* of $S_1[25]$ and $S_2[26]$. Our goal is to compute a longest common subsequence (LCS for short) for any two input sequences (over $\Sigma = \{0, 1, 2, \ldots, 9\}$, for now).

The following list contains the specifications for Assignment #1 (10 marks in total):

1. Write a single-file C program with multiple functionalities (i.e. objectives). Suppose your executable name (after compilation) is "`myprogram`":

   ```
   >myprogram
   ```

   Running your program through command-line as in the above will read in the two input sequences from the `stdin` (that is, the terminal, and the two sequences are separated by the return key), by providing an interface as follows (functionality 1):

   ```
   To compute an LCS,
   enter the first sequence:
   enter the second sequence:
   ```

   and then proceed to compute an LCS.

2. Your program needs to do a sequence check, that is, if there is any non-digit character, then prints an error and re-provides the interface, as follows (functionality 2):

   ```
   Error, non-digit character detected!

   To compute an LCS,
   enter the first sequence:
   enter the second sequence:
   ```

3. After successfully reading in the two input sequences, your program prints out the two input sequences to the `stdout` (that is, the screen), as follows (functionality 3):

   ```
   # Two input sequences (length = 25, 26) are:
   3110283025318292818318143
   1311230172803163125184 1324
   ```

4. After an LCS has been computed (functionality 4, for which there will be a lecture to explain how to compute an LCS), your program prints it out to the `stdout` as well (functionality 5, the following `311230128131814` is indeed an LCS).

   ```
   # an LCS (length = 15) is:
   311230128131814
   ```

   //End of description of Assignment #1.