1. За феновете на A series of unfortunate events:
    * Ava J. sent you a letter for your secret annual meeting.
    * You must decode it to understand when the meeting will take place.
    * Ava never makes mistakes when writing her letters. If you ever see one, consider that a clue.
    * <p>
    * The first 3 letters of Ava's letters always represents the *month*. Be sure of that.
    * <p>
    * Decode the exact *hour* of the meeting by counting the number of hidden paragraphs.
    * If you see a punctuation mistake that would mean the end of a hidden paragraph. *1*
    * If you count more that 24 hours, something is not quite right.
    * <p>
    * Decode the exact *minutes* of the meeting by counting the number of misplaced symbols.
    * For every misplaced symbol add 5 minutes. *2*
    * If you count more than 60 minutes, something is not quite right.
    * <p>
    * Decode the exact *day* of the month Ava wants you to meet her by counting the length
    * of the hidden paragraph which starts with the word "The" (including a period at the end). *3,4*
    * <p>
    * *1* - Ava would never make punctuation mistakes.
    * Normally every sentence in her letters ends with one occurrence of a period, no more no less.
    * *2* - Ava would never replace letters from the alphabet with symbols.
    * Normally she writes S instead of $.
    * *3* - Ava would never start a hidden paragraph with the word "The".
    * *4* - Ava would never leave a lot of unnecessary space between the hidden paragraphs.
    * Either at the start, or the end.
    *
    * @param message The coded message that Ava sent you.
    * @return decoded Message in the format [*day* *month* at *hour*:*minutes*], e.g. "17th October at 9:00".
    * Print "Maybe another *time*" if you feel something is not right.
    * (Replace *time* with the reason you think the meeting might not happen,
    * e.g. "Maybe another day" or "Maybe another minute").
    * If the message is empty, print "Maybe another day".

Function prototype:
char* decode(char* message);

Credits to MJT @ FMI 2020/2021 course.

2. Given two sequences, find the length of longest subsequence present in both of them. A subsequence is a sequence that appears in the same relative order, but not necessarily contiguous. For example, “abc”, “abg”, “bdf”, “aeg”, ‘”acefg”, .. etc are subsequences of “abcdefg”.

Examples:
LCS for input Sequences “ABCDGH” and “AEDFHR” is “ADH” of length 3.
LCS for input Sequences “AGGTAB” and “GXTXAYB” is “GTAB” of length 4.