Recall the Longest Palindromic Subsequence problem from the DPV text (exercise 6.7) and discussed in class, which is to find the length of the longest subsequence of \( x = x_1x_2 \cdots x_n \) which is a palindrome. (Note that it is a subsequence, so the chosen characters do not have to appear next to each other in \( x \).) To find the longest length, we define the subproblem \( LP(i, j) \) to be the length of the longest palindromic subsequence of \( x_i \cdots x_j \). The recurrence we derived in class looks like

\[
LP(i, j) = \begin{cases} 
0 & \text{if } i > j \\
1 & \text{if } i = j \\
2 + LP(i + 1, j - 1) & \text{if } i < j \text{ and } x_i = x_j \\
\max(LP(i + 1, j), LP(i, j - 1)) & \text{if } i < j \text{ and } x_i \neq x_j
\end{cases}
\]

The length of the longest palindromic subsequence of \( x \) is thus \( LP(1, n) \).

You are to write pseudo-code which will fill up an array \( LP \) in either a bottom-up (iterative) manner or in a top-down (memoized) manner.

**PART I:** My pseudo-code is **ITERATIVE** or **MEMOIZED** (choose one)

**PART II:** The time bound of my pseudo-code is:

**PART III:** Give pseudo-code.