Reg.No : 24BCE0554
Name   : Partha Pratim Gogoi
Topic : Longest Common Subsequence

Algorithm:

```
1   longestCommonSubsequence(a,b):                          [T.C: O(n*m)]
    1.1   n = length(a)                                     [T.C: O(1)]
    1.2   m = length(b)                                     [T.C: O(1)]
    1.3   dp[n+1][m+1] = {0};                    [T.C: O((n+1)*(m+1))]
    1.4   for i 1→n:                                        [T.C: O(n)]
      1.4.1   for j 1→m:                                    [T.C: O(m)]
        1.4.1.1   if (a[i-1]==b[j-1]):                      [T.C: O(1)]
        1.4.1.2   dp[i][j] = 1+dp[i-1][j-1]                 [T.C: O(1)]
        1.4.1.3   else:                                     [T.C: O(1)]
        1.4.1.4   dp[i][j] = max(dp[i][j-1],dp[i-1][j])     [T.C: O(1)]
    1.5   return dp[n][m]
```

Time Complexity:

Total Time Complexity = $O((n+1)*(m+1)) + O(n)*O(m) + 2O(1)$
= $O(n*m)$

Source Code:

```cpp
#include <iostream>
#include <string>
#include <vector>
#include <algorithm>
using namespace std;


int longestCommonSubsequence(string a, string b) {
        int n = a.length();
        int m = b.length();

        // --- Create 0-initialized 2D array to build LCS --- //
        // --- vector<int> inner(m+1, 0); ------------------- //
        // --- vector<vector<int>> dp(n+1, inner); ---------- //
        vector<vector<int>> dp(n+1, vector<int>(m+1, 0));

        // --- LCS Algorithm --- //
        for (int i=1; i<n+1; i++) {
        for (int j=1; j<m+1; j++) {
                // ------------------------- //
                // --- Check element match --- //
                // ------------------------- //
                if (a[i-1] == b[j-1]) {
                        // --------------- DEBUG Statements ------------- //
                        // cout << endl << "Element match:" << endl; ------ //
                        // cout << dp[i][j] << endl; --------------------- //
                        // cout << dp[i-1][j-1] << endl; ----------------- //
                        // ---------------------------------------------- //
                        dp[i][j] = 1 + dp[i-1][j-1];
                } else {
                        // --------------- DEBUG Statements ------------- //
                        // cout << endl << "Element mismatch:" << endl; --- //
                        // cout << dp[i][j] << endl; --------------------- //
                        // cout << dp[i][j-1] << endl; ------------------- //
                        // cout << dp[i-1][j] << endl; ------------------- //
                        // ---------------------------------------------- //
                        dp[i][j] = max(dp[i-1][j], dp[i][j-1]);
                }
        }}

        return dp[n][m];
}

int main() {
        string a = "wrtvertubryvcwrybutruibvctyret";
        string b = "wqercfw6vtetubervycwetyebvc";

        cout << "Length of Longest Common Subqeuence: " <<
longestCommonSubsequence(a,b);
        cout << endl;

        return 0;
}
```

Sample Output:

```
24BCE0554

◄ 0s ⊚ ../'[1143] - Longest Common Subsequence'/a.out
Length of Longest Common Subqevence: 15
```