

```
1 //
2 // You are given strings s and t.
3 // Find one longest string that is a subsequence (non-continuous) of both s and t.
4 //
5 // Time Complexity: O(NM)
6 //
7
8 #include <bits/stdc++.h>
9
10 using namespace std;
11
12 int main() {
13     string s, t;
14     cin >> s >> t;
15
16     int n = (int)s.size(), m = (int)t.size();
17
18     vector<vector<int>> dp(n, vector<int>(m, 0));
19
20     dp[0][0] = (s[0] == t[0]);
21     for (int i = 1; i < n; i++) dp[i][0] = max(dp[i-1][0], (int)(s[i] == t[0]));
22     for (int i = 1; i < m; i++) dp[0][i] = max(dp[0][i-1], (int)(s[0] == t[i]));
23
24     for (int i = 1; i < n; i++) {
25         for (int j = 1; j < m; j++) {
26             dp[i][j] = max({
27                 dp[i-1][j-1] + (s[i] == t[j]),
28                 dp[i-1][j],
29                 dp[i][j-1]
30             });
31         }
32     }
33
34     // this is the length of the LCS
35     // cout << dp[n-1][m-1] << endl;
36
37     if (dp[n-1][m-1] == 0) {
38         cout << "" << endl;
39         return 0;
40     }
41
42     int i = n-1;
43     int j = m-1;
44
45     string ans = "";
46     while (i >= 0 && j >= 0) {
47         if (s[i] == t[j]) {
48             ans += s[i];
49             i--; j--;
50             continue;
51         }
52
53         if (i == 0) j--;
54         else if (j == 0) i--;
55         else {
56             if (dp[i-1][j] > dp[i][j-1]) i--;
57             else j--;
58         }
59     }
60
61     reverse(ans.begin(), ans.end());
62     cout << ans << endl;
63     return 0;
64 }
```