

## Question 1

Correct

Mark 1.00 out of 1.00

Given two strings find the length of the common longest subsequence(need not be contiguous) between the two.

Example:

s1: ggtabe

s2: tgatasb

s1	a	g	g	t	a	b	
s2	g	x	t	x	a	y	b

**The length is 4**

Solveing it using Dynamic Programming

**For example:**

Input	Result
aab	2
azb	

**Answer:** (penalty regime: 0 %)

```

1  #include <stdio.h>
2  #include <string.h>
3
4  int longestCommonSubsequence(char* X, char* Y) {
5      int m = strlen(X);
6      int n = strlen(Y);
7      int dp[m + 1][n + 1];
8
9      for (int i = 0; i <= m; i++) {
10         for (int j = 0; j <= n; j++) {
11             if (i == 0 || j == 0) {
12                 dp[i][j] = 0;
13             }
14             else if (X[i - 1] == Y[j - 1]) {
15                 dp[i][j] = dp[i - 1][j - 1] + 1;
16             }
17             else {
18                 dp[i][j] = (dp[i - 1][j] > dp[i][j - 1]) ? dp[i - 1][j] : dp[i][j - 1];
19             }
20         }
21     }
22
23     return dp[m][n];
24 }
25
26 int main() {
27     char X[50];
28     char Y[50];
29     scanf("%s %s", X, Y);
30     int length = longestCommonSubsequence(X, Y);
31     printf("%d", length);
32     return 0;
33 }
34
35 }
```

	Input	Expected	Got	
✓	aab azb	2	2	✓
✓	ABCD ABCD	4	4	✓

Passed all tests! ✓

**Correct**

Marks for this submission: 1.00/1.00.

[◀ 2-DP-Playing with chessboard](#)

Jump to...



[4-DP-Longest non-decreasing Subsequence ▶](#)