

[Dashboard](#) / [My courses](#) / [CS23331-DAA-2023-CSE](#) / [Dynamic Programming](#) / [3-DP-Longest Common Subsequence](#)

<b>Started on</b>	Tuesday, 29 October 2024, 1:50 PM
<b>State</b>	Finished
<b>Completed on</b>	Tuesday, 29 October 2024, 2:04 PM
<b>Time taken</b>	14 mins 22 secs
<b>Marks</b>	1.00/1.00
<b>Grade</b>	<b>10.00</b> out of 10.00 ( <b>100%</b> )

## 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 azb	2

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

```

1  #include <stdio.h>
2  #include <string.h>
3
4  int longestCommonSubsequence(char *s1, char *s2) {
5      int m = strlen(s1);
6      int n = strlen(s2);
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             } else if (s1[i - 1] == s2[j - 1]) {
14                 dp[i][j] = dp[i - 1][j - 1] + 1;
15             } else {
16                 dp[i][j] = (dp[i - 1][j] > dp[i][j - 1]) ? dp[i - 1][j] : dp[i][j - 1];
17             }
18         }
19     }
20
21     return dp[m][n];
22 }
23
24 int main() {
25     char s1[100], s2[100];
26     scanf("%s", s1);
27     scanf("%s", s2);
28
29     printf("%d\n", longestCommonSubsequence(s1, s2));
30     return 0;
31 }
32

```

	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 ▶