

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

<b>Started on</b>	Tuesday, 29 October 2024, 2:18 PM
<b>State</b>	Finished
<b>Completed on</b>	Tuesday, 29 October 2024, 2:36 PM
<b>Time taken</b>	18 mins 7 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	2
azb	

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

```

1  #include <stdio.h>
2  #include <string.h>
3
4  #define M 100
5
6  int max(int a, int b) {
7      return (a > b) ? a : b;
8  }
9
10 int main() {
11     char s1[M], s2[M];
12     int dp[M+1][M+1];
13
14     scanf("%s", s1);
15     scanf("%s", s2);
16
17     int n = strlen(s1);
18     int m = strlen(s2);
19
20     for (int i = 0; i <= n; i++) {
21         for (int j = 0; j <= m; j++) {
22             if (i == 0 || j == 0)
23                 dp[i][j] = 0;
24             else if (s1[i-1] == s2[j-1])
25                 dp[i][j] = dp[i-1][j-1] + 1;
26             else
27                 dp[i][j] = max(dp[i-1][j], dp[i][j-1]);
28         }
29     }
30
31     printf("%d\n", dp[n][m]);
32
33     return 0;
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 ▶