

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

Started on	Tuesday, 19 November 2024, 9:52 PM
State	Finished
Completed on	Tuesday, 19 November 2024, 9:57 PM
Time taken	4 mins 11 secs
Marks	1.00/1.00
Grade	10.00 out of 10.00 (100%)

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 MAX_LEN 100
5
6  int main() {
7      char s1[MAX_LEN], s2[MAX_LEN];
8      int dp[MAX_LEN + 1][MAX_LEN + 1];
9      scanf("%s", s1);
10     scanf("%s", s2);
11     int m = 0, n = 0;
12     while (s1[m] != '\0') m++;
13     while (s2[n] != '\0') n++;
14
15     // Initialize the DP table
16     for (int i = 0; i <= m; i++) {
17         for (int j = 0; j <= n; j++) {
18             if (i == 0 || j == 0) {
19                 dp[i][j] = 0;
20             } else if (s1[i - 1] == s2[j - 1]) {
21                 dp[i][j] = dp[i - 1][j - 1] + 1;
22             } else {
23                 dp[i][j] = (dp[i - 1][j] > dp[i][j - 1]) ? dp[i - 1][j] : dp[i][j - 1];
24             }
25         }
26     }
27     printf("%d\n", dp[m][n]);
28
29     return 0;
30 }
31

```

	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 ▶