

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

Started on	Monday, 18 November 2024, 8:36 PM
State	Finished
Completed on	Monday, 18 November 2024, 8:36 PM
Time taken	27 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 azb	2

Answer: (penalty regime: 0 %)

```

1  #include <stdio.h>
2  #include <string.h>
3  #define MAX_LEN 100
4
5  int lcsLength(const char *s1, const char *s2) {
6      int len1 = strlen(s1);
7      int len2 = strlen(s2);
8      int dp[MAX_LEN + 1][MAX_LEN + 1];
9      for (int i = 0; i <= len1; i++) {
10         for (int j = 0; j <= len2; 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     return dp[len1][len2];
21 }
22 int main() {
23     char s1[MAX_LEN + 1], s2[MAX_LEN + 1];
24     scanf("%s", s1);
25     scanf("%s", s2);
26     int result = lcsLength(s1, s2);
27     printf("%d\n", result);
28     return 0;
29 }
```

	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 ▶