



## 3-DP-Longest Common Subsequence

Started on	Friday, 10 October 2025, 2:05 PM
State	Finished
Completed on	Friday, 10 October 2025, 2:14 PM
Time taken	8 mins 22 secs
Marks	1.00/1.00
Grade	<b>10.00</b> 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	<b>g</b>	<b>t</b>	a	<b>b</b>		
s2		<b>g</b>	x	<b>t</b>	x	a	y	<b>b</b>

**The length is 4**

Solveing it using Dynamic Programming

For example:

Input	Result
aab	2
azb	

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

```

1
2 #include <stdio.h>
3 #include <stdlib.h>
4 #include <string.h>
5
6 int main(void) {
7     char A[10005], B[10005];
8     if (scanf("%10004s %10004s", A, B) != 2) {
9         fprintf(stderr, "Expected two strings (whitespace separated) on input.\n");
10        return 1;
11    }
12
13    int n = (int)strlen(A);
14    int m = (int)strlen(B);
15
16    if (n == 0 || m == 0) {
17        printf("0\n");
18        return 0;
19    }
20
21    int *prev = (int*) calloc(m + 1, sizeof(int));
22    int *curr = (int*) calloc(m + 1, sizeof(int));
23    if (!prev || !curr) {
24        fprintf(stderr, "Memory allocation failed\n");
25        free(prev);
26        free(curr);
27        return 1;
28    }
29
30    for (int i = 1; i <= m; i++) {
31        curr[i] = prev[i - 1];
32    }
33
34    for (int i = 1; i <= n; i++) {
35        for (int j = 1; j <= m; j++) {
36            if (A[i - 1] == B[j - 1]) {
37                curr[j] = prev[j - 1] + 1;
38            } else {
39                curr[j] = prev[j];
40            }
41        }
42    }
43
44    printf("%d\n", curr[m]);
45
46    free(prev);
47    free(curr);
48
49    return 0;
50}
```

```

25     free(prev),
26     free(curr);
27     return 1;
28 }
29
30 for (int i = 1; i <= n; ++i) {
31
32     curr[0] = 0;
33     for (int j = 1; j <= m; ++j) {
34         if (A[i - 1] == B[j - 1])
35             curr[j] = prev[j - 1] + 1;
36         else
37             curr[j] = (prev[j] > curr[j - 1]) ? prev[j] : curr[j - 1];
38     }
39
40     int *tmp = prev;
41     prev = curr;
42     curr = tmp;
43 }
44
45 printf("%d\n", prev[m]);
46
47 free(prev);
48 free(curr);
49 return 0;
50 }
51
52

```

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

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

Finish review

Back to Course

Data retention summary