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	**10.00** out of 10.00 (**100**%)

Question 1 Correct Mark 1.00 out of 1.00 Flag question

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

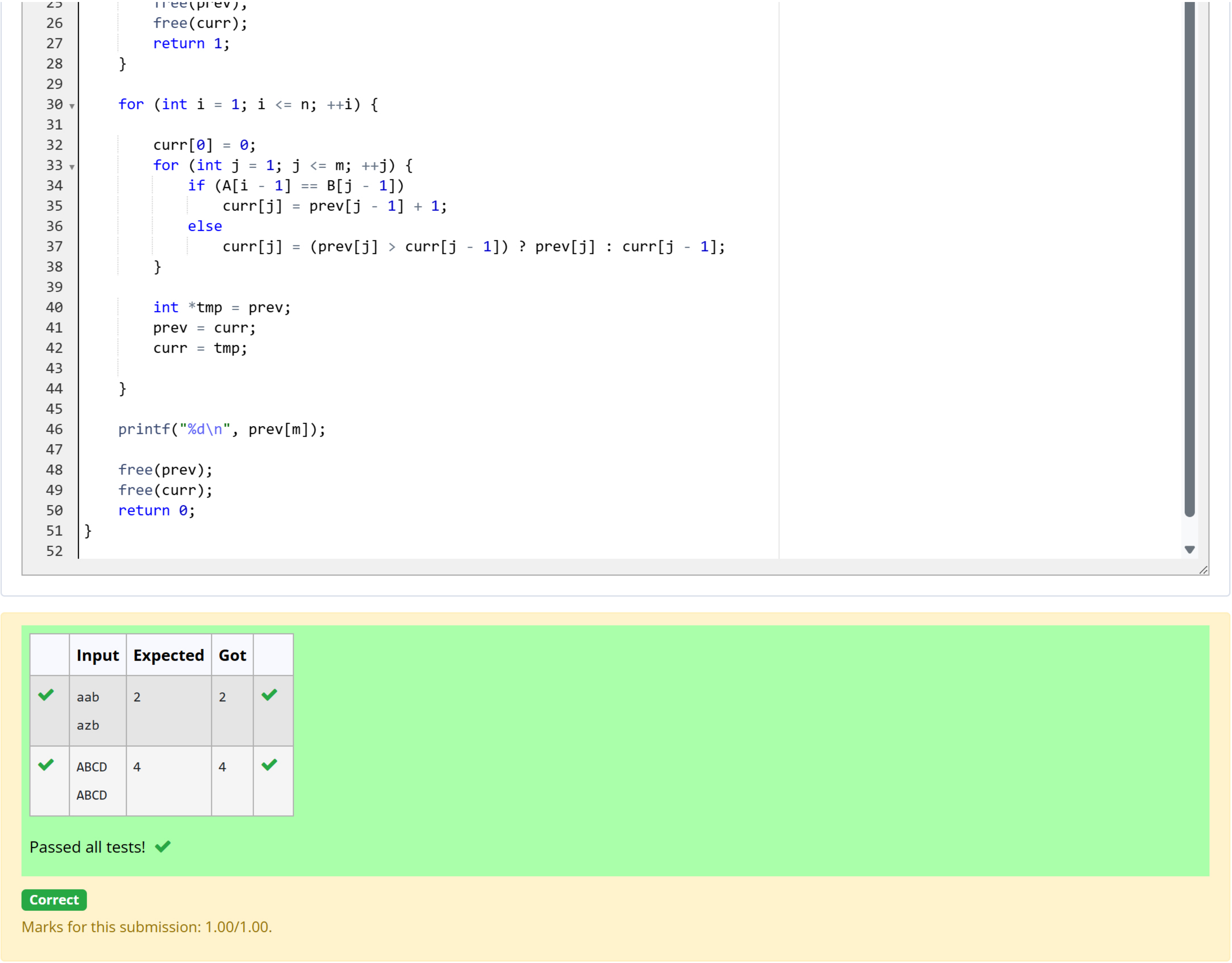
Solveing it using Dynamic Programming

For example:

Input	Result
aab azb	2

Answer: (penalty regime: 0 %)

```

#include <stdio.h>
#include <stdlib.h>
#include <string.h>

int main(void) {
    char A[10005], B[10005];
    if (scanf("%10004s %10004s", A, B) != 2) {
        fprintf(stderr, "Expected two strings (whitespace separated) on input.\n");
        return 1;
    }

    int n = (int)strlen(A);
    int m = (int)strlen(B);

    if (n == 0 || m == 0) {
        printf("0\n");
        return 0;
    }

    int *prev = (int*) calloc(m + 1, sizeof(int));
    int *curr = (int*) calloc(m + 1, sizeof(int));
    if (!prev || !curr) {
        fprintf(stderr, "Memory allocation failed\n");
        free(prev);
```



```
        free(prev);
        free(curr);
        return 1;
    }

    for (int i = 1; i <= n; ++i) {

        curr[0] = 0;
        for (int j = 1; j <= m; ++j) {
            if (A[i - 1] == B[j - 1])
                curr[j] = prev[j - 1] + 1;
            else
                curr[j] = (prev[j] > curr[j - 1]) ? prev[j] : curr[j - 1];
        }

        int *tmp = prev;
        prev = curr;
        curr = tmp;

    }

    printf("%d\n", prev[m]);

    free(prev);
    free(curr);
    return 0;
}
```

	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