



✓ Done

**Opened:** Tuesday, 1 July 2025, 12:27 AM  
**Closed:** Monday, 27 October 2025, 2:00 PM

Attempts allowed: 4  
Time limit: 45 mins  
Grading method: Highest grade

Summary of your previous attempts

Attempt	State	Review
1	Finished Submitted Wednesday, 8 October 2025, 8:25 AM	Not permitted

[Back to the course](#)

[Back to Course](#)



SANKARA GOMATHI R 2024-CSE ▾

S2

Started on	Wednesday, 8 October 2025, 8:25 AM
State	Finished
Completed on	Wednesday, 8 October 2025, 8:26 AM
Time taken	1 min 1 sec
Grade	10.00 out of 10.00 (100%)

**Question 1** | Correct | Mark 10.00 out of 10.00**Playing with Chessboard:**

Ram is given with an  $n \times n$  chessboard with each cell with a monetary value. Ram stands at the (0,0), that the position of the top left white rook. He is been given a task to reach the bottom right black rook position ( $n-1$ ,  $n-1$ ) constrained that he needs to reach the position by traveling the maximum monetary path under the condition that he can only travel one step right or one step down the board. Help ram to achieve it by providing an efficient DP algorithm.

**Example:****Input**

```
3
1 2 4
2 3 4
8 7 1
```

**Output:**

```
19
```

**Explanation:**

Totally there will be 6 paths among that the optimal is  
Optimal path value:  $1+2+8+7+1=19$

**Input Format**

First Line contains the integer  $n$   
The next  $n$  lines contain the  $n \times n$  chessboard values

**Output Format**

Print Maximum monetary value of the path

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

```
1 #include <stdio.h>
2
3 int max(int a, int b) {
4     return (a > b) ? a : b;
5 }
6
7 int main() {
8     int n;
9     scanf("%d", &n);
10    int board[n][n];
11    int dp[n][n];
12
13    // Input the board values
14    for (int i = 0; i < n; i++)
15        for (int j = 0; j < n; j++)
16            scanf("%d", &board[i][j]);
17
18    dp[0][0] = board[0][0];
19
20    // Fill first row
21    for (int j = 1; j < n; j++) {
22        dp[0][j] = dp[0][j-1] + board[0][j];
23    }
24
25    // Fill first column
26    for (int i = 1; i < n; i++) {
27        dp[i][0] = dp[i-1][0] + board[i][0];
28    }
29
30    // Fill the rest of the dp table
31    for (int i = 1; i < n; i++) {
32        for (int j = 1; j < n; j++) {
33            dp[i][j] = board[i][j] + max(dp[i-1][j], dp[i][j-1]);
34        }
35    }
36
```

```
37 | // The maximum monetary value path sum
38 | printf("%d\n", dp[n-1][n-1]);
39 |
40 | return 0;
41 | }
42 |
```

	Input	Expected	Got	
✓	3 1 2 4 2 3 4 8 7 1	19	19	✓
✓	3 1 3 1 1 5 1 4 2 1	12	12	✓
✓	4 1 1 3 4 1 5 7 8 2 3 4 6 1 6 9 0	28	28	✓

Passed all tests! ✓

Correct

Marks for this submission: 10.00/10.00.

[Back to Course](#)





SANKARA GOMATHI R 2024-CSE ▾

S2

Started on	Wednesday, 8 October 2025, 8:27 AM
State	Finished
Completed on	Wednesday, 8 October 2025, 8:28 AM
Time taken	52 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	<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**

Solving 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  int max(int a, int b) {
5      return (a > b) ? a : b;
6  }
7
8  int longestCommonSubsequence(char *s1, char *s2) {
9      int m = strlen(s1);
10     int n = strlen(s2);
11     int dp[m+1][n+1];
12
13     for (int i = 0; i <= m; i++) dp[i][0] = 0;
14     for (int j = 0; j <= n; j++) dp[0][j] = 0;
15
16     for (int i = 1; i <= m; i++) {
17         for (int j = 1; j <= n; j++) {
18             if (s1[i-1] == s2[j-1])
19                 dp[i][j] = dp[i-1][j-1] + 1;
20             else
21                 dp[i][j] = max(dp[i-1][j], dp[i][j-1]);
22         }
23     }
24
25     return dp[m][n];
26 }
27
28 int main() {
29     char s1[1001], s2[1001];
30     scanf("%s %s", s1, s2);
31     printf("%d\n", longestCommonSubsequence(s1, s2));
32     return 0;
33 }
34

```

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

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

[Back to Course](#)



SANKARA GOMATHI R 2024-CSE ▾

S2

Started on	Wednesday, 8 October 2025, 8:28 AM
State	Finished
Completed on	Wednesday, 8 October 2025, 8:28 AM
Time taken	35 secs
Marks	1.00/1.00
Grade	10.00 out of 10.00 (100%)



**Question 1** | Correct | Mark 1.00 out of 1.00

Problem statement:

Find the length of the Longest Non-decreasing Subsequence in a given Sequence.

Eg:

Input:9

Sequence: [-1,3,4,5,2,2,2,2,3]

the subsequence is [-1,2,2,2,2,3]

Output:6

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

```

1 | #include <stdio.h>
2 |
3 | int max(int a, int b) {
4 |     return (a > b) ? a : b;
5 | }
6 |
7 | int longestNonDecreasingSubsequence(int arr[], int n) {
8 |     int dp[n];
9 |     for (int i = 0; i < n; i++) dp[i] = 1;
10 |
11 |     for (int i = 1; i < n; i++) {
12 |         for (int j = 0; j < i; j++) {
13 |             if (arr[j] <= arr[i]) {
14 |                 dp[i] = max(dp[i], dp[j] + 1);
15 |             }
16 |         }
17 |     }
18 |
19 |     int ans = 1;
20 |     for (int i = 0; i < n; i++) {
21 |         if (dp[i] > ans) ans = dp[i];
22 |     }
23 |     return ans;
24 | }
25 |
26 | int main() {
27 |     int n;
28 |     scanf("%d", &n);
29 |     int arr[n];
30 |     for (int i = 0; i < n; i++) scanf("%d", &arr[i]);
31 |
32 |     printf("%d\n", longestNonDecreasingSubsequence(arr, n));
33 |     return 0;
34 | }
35 |

```

	Input	Expected	Got	
✓	9 -1 3 4 5 2 2 2 2 3	6	6	✓
✓	7 1 2 2 4 5 7 6	6	6	✓

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

[Back to Course](#)