



Dashboard My courses



CS23331-DAA-2024-CSE / 1-DP-Playing with Numbers

1-DP-Playing with Numbers

Started on	Sunday, 2 November 2025, 10:23 PM
State	Finished
Completed on	Sunday, 2 November 2025, 10:24 PM
Time taken	1 min 22 secs
Grade	10.00 out of 10.00 (100%)

Question 1 | Correct Mark 10.00 out of 10.00 

Playing with Numbers:

Ram and Sita are playing with numbers by giving puzzles to each other. Now it was Ram term, so he gave Sita a positive integer 'n' and two numbers 1 and 3. He asked her to find the possible ways by which the number n can be represented using 1 and 3. Write any efficient algorithm to find the possible ways.

Example 1:

Input: 6

Output: 6

Explanation: There are 6 ways to represent number with 1 and 3

1+1+1+1+1+1

3+3

1+1+1+3

1+1+3+1

1+3+1+1

3+1+1+1

Input Format

First Line contains the number n

Output Format

Print: The number of possible ways 'n' can be represented using 1 and 3

Sample Input

6

Sample Output

6

Answer: (penalty regime: 0 %)

```
1 #include <stdio.h>
2
3 long long countWays(int n) {
4     if (n < 0)
5         return 0;
6     if (n == 0)
7         return 1;
8
9     long long dp[n + 1];
10
11    dp[0] = 1;
12    for (int i = 1; i <= n; i++) {
13        dp[i] = dp[i - 1]; // taking +1
14        if (i >= 3)
15            dp[i] += dp[i - 3]; // taking +3
16    }
17
18    return dp[n];
19}
20
21 int main() {
22     int n;
23     scanf("%d", &n);
24     printf("%lld\n", countWays(n));
25     return 0;
26}
```

	Input	Expected	Got	
✓	6	6	6	✓
✓	25	8641	8641	✓

✓	100	24382819596721629	24382819596721629	✓
---	-----	-------------------	-------------------	---

Passed all tests! ✓

Correct

Marks for this submission: 10.00/10.00.

[Finish review](#)

[Back to Course](#)

[Data retention summary](#)



 Dashboard My courses 

CS23331-DAA-2024-CSE / 2-DP-Playing with chessboard

2-DP-Playing with chessboard

Started on Monday, 3 November 2025, 6:04 AM

State Finished

Completed on Monday, 3 November 2025, 6:05 AM

Time taken 47 secs

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 is the position of the top left white rook. He is 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*n chessboard values

Output Format

Print Maximum monetary value of the path

Answer: (penalty regime: 0 %)

```
1 #include <stdio.h>
2
3 #define MAX 100
4
5 int max(int a, int b) {
6     return (a > b) ? a : b;
7 }
8
9 int main() {
10    int n;
11    int board[MAX][MAX], dp[MAX][MAX];
12
13    scanf("%d", &n);
14
15    // Read the chessboard
16    for (int i = 0; i < n; i++) {
17        for (int j = 0; j < n; j++) {
18            scanf("%d", &board[i][j]);
19        }
20    }
21
22    // Initialize DP table
23    dp[0][0] = board[0][0];
24
25    // First row
26    for (int j = 1; j < n; j++)
27        dp[0][j] = dp[0][j - 1] + board[0][j];
28
29    // First column
30    for (int i = 1; i < n; i++)
31        dp[i][0] = dp[i - 1][0] + board[i][0];
32
33    // Fill the rest of the DP table
34    for (int i = 1; i < n; i++) {
35        for (int j = 1; j < n; j++) {
36            dp[i][j] = board[i][j] + max(dp[i - 1][j], dp[i][j - 1]);
37        }
38    }
39
40    printf("%d\n", dp[n - 1][n - 1]);
41}
```

```
42 }     return 0;  
43 }  
44 }
```

	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.

[Finish review](#)

[Back to Course](#)

Data retention summary

[Dashboard](#) [My courses](#)[CS23331-DAA-2024-CSE](#) / 3-DP-Longest Common Subsequence

3-DP-Longest Common Subsequence

Started on	Monday, 3 November 2025, 6:05 AM
State	Finished
Completed on	Monday, 3 November 2025, 6:05 AM
Time taken	37 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

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 1000
5
6 int max(int a, int b) {
7     return (a > b) ? a : b;
8 }
9
10 int main() {
11     char s1[MAX], s2[MAX];
12     scanf("%s", s1);
13     scanf("%s", s2);
14
15     int m = strlen(s1);
16     int n = strlen(s2);
17     int dp[m + 1][n + 1];
18
19     // Initialize
20     for (int i = 0; i <= m; i++)
21         dp[i][0] = 0;
22     for (int j = 0; j <= n; j++)
23         dp[0][j] = 0;
24
25     // Fill dp table
26     for (int i = 1; i <= m; i++) {
27         for (int j = 1; j <= n; j++) {
28             if (s1[i - 1] == s2[j - 1])
29                 dp[i][j] = 1 + dp[i - 1][j - 1];
30             else
31                 dp[i][j] = max(dp[i - 1][j], dp[i][j - 1]);
32         }
33     }
34
35     printf("%d\n", dp[m][n]);
36
37     return 0;
38 }
```

	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

[Dashboard](#) [My courses](#)[CS23331-DAA-2024-CSE](#) / 4-DP-Longest non-decreasing Subsequence

4-DP-Longest non-decreasing Subsequence

Started on	Monday, 3 November 2025, 6:06 AM
State	Finished
Completed on	Monday, 3 November 2025, 6:07 AM
Time taken	1 min 17 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 #define MAX 1000
```

```

4
5 int max(int a, int b) {
6     return (a > b) ? a : b;
7 }
8
9 int main() {
10    int n;
11    scanf("%d", &n);
12
13    int arr[MAX];
14    for (int i = 0; i < n; i++) {
15        scanf("%d", &arr[i]);
16    }
17
18    int dp[MAX];
19    int result = 1;
20
21    // Initialize dp
22    for (int i = 0; i < n; i++)
23        dp[i] = 1;
24
25    // Fill dp array
26    for (int i = 1; i < n; i++) {
27        for (int j = 0; j < i; j++) {
28            if (arr[j] <= arr[i]) // non-decreasing condition
29                dp[i] = max(dp[i], dp[j] + 1);
30        }
31        result = max(result, dp[i]);
32    }
33
34    printf("%d\n", result);
35
36    return 0;
37 }
38

```

	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.

[Finish review](#)

[Back to Course](#)

Data retention summary