

1-DP-Playing with Numbers

Started on Monday, 13 October 2025, 1:22 PM

State Finished

Completed on Monday, 13 October 2025, 1:42 PM

Time taken 20 mins 13 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 long long int countWays(int n){
3     long long int dp[n+1];
4     dp[0]=1;
5     for(int i=1;i<=n;i++){
6         dp[i]=0;
7         if(i>=1)
8             dp[i]+=dp[i-1];
9         if(i>=3)
10            dp[i]+=dp[i-3];
11    }
12    return dp[n];
13 }
14 int main(){
15     int n;
16     scanf("%d",&n);
17     printf("%lld\n",countWays(n));
18     return 0;
19 }
```

	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](#)



2-DP-Playing with chessboard

Started on Monday, 13 October 2025, 1:31 PM

State Finished

Completed on Monday, 13 October 2025, 1:39 PM

Time taken 8 mins 30 secs

Grade 10.00 out of 10.00 (100%)

Question 1 | Correct Mark 10.00 out of 10.00 [Flag question](#)

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

```

	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](#)



3-DP-Longest Common Subsequence

Started on Monday, 13 October 2025, 1:32 PM

State Finished

Completed on Monday, 13 October 2025, 1:46 PM

Time taken 14 mins 50 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

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

[Finish review](#)

[Back to Course](#)



4-DP-Longest non-decreasing Subsequence

Started on Monday, 13 October 2025, 1:32 PM

State Finished

Completed on Monday, 13 October 2025, 1:48 PM

Time taken 16 mins 20 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

Problem statement:

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

Eg:

Input:9

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

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

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++)
10         dp[i] = 1;
11
12    for (int i = 1; i < n; i++) {
13        for (int j = 0; j < i; j++) {
14            if (arr[j] <= arr[i]) {
15                dp[i] = max(dp[i], dp[j] + 1);
16            }
17        }
18    }
19
20    int res = 1;
21    for (int i = 0; i < n; i++)
22        if (dp[i] > res) res = dp[i];
23
24    return res;
25 }
26
27 int main() {
28     int n;
29     scanf("%d", &n);
30     int arr[n];
31     for (int i = 0; i < n; i++)
32         scanf("%d", &arr[i]);
33
34     printf("%d\n", longestNonDecreasingSubsequence(arr, n));
35     return 0;
36 }
37

```

	Input	Expected	Got	
✓	9 -1 3 4 5 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](#)