

## 2-DP-Playing with chessboard

Started on	Wednesday, 8 October 2025, 9:11 AM
State	Finished
Completed on	Wednesday, 8 October 2025, 9:19 AM
Time taken	7 mins 50 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\*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 void solve_chessboard_path() {
8     int n;
9     scanf("%d", &n);
10    int M[n][n];
11    long long DP[n][n];
12    for (int i = 0; i < n; i++) {
13        for (int j = 0; j < n; j++) {
14            scanf("%d", &M[i][j]);
15        }
16    }
17    DP[0][0] = M[0][0];
18    for (int j = 1; j < n; j++) {
19        DP[0][j] = M[0][j] + DP[0][j-1];
20    }
21    for (int i = 1; i < n; i++) {
22        DP[i][0] = M[i][0] + DP[i-1][0];
23    }
24    for (int i = 1; i < n; i++) {
25        for (int j = 1; j < n; j++) {
26            DP[i][j] = M[i][j] + max(DP[i-1][j], DP[i][j-1]);
27        }
28    }
29    printf("%lld\n", DP[n-1][n-1]);
30 }
31
32 int main() {
33     solve_chessboard_path();
34     return 0;
35 }
```

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](#)