



SIVA S 2024-CSE

S2

Started on Wednesday, 8 October 2025, 4:01 PM**State** Finished**Completed on** Wednesday, 8 October 2025, 4:02 PM**Time taken** 38 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 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
12    // Read the board values
13    for (int i = 0; i < n; i++)
14        for (int j = 0; j < n; j++)
15            scanf("%d", &board[i][j]);
16
17    // DP table to store max path sums
18    int dp[n][n];
19    dp[0][0] = board[0][0];
20
21    // Fill first row
22    for (int j = 1; j < n; j++)
23        dp[0][j] = dp[0][j - 1] + board[0][j];
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    // Fill rest of the table
30    for (int i = 1: i < n: i++) {
```

--	--	--	--	--

	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](#)