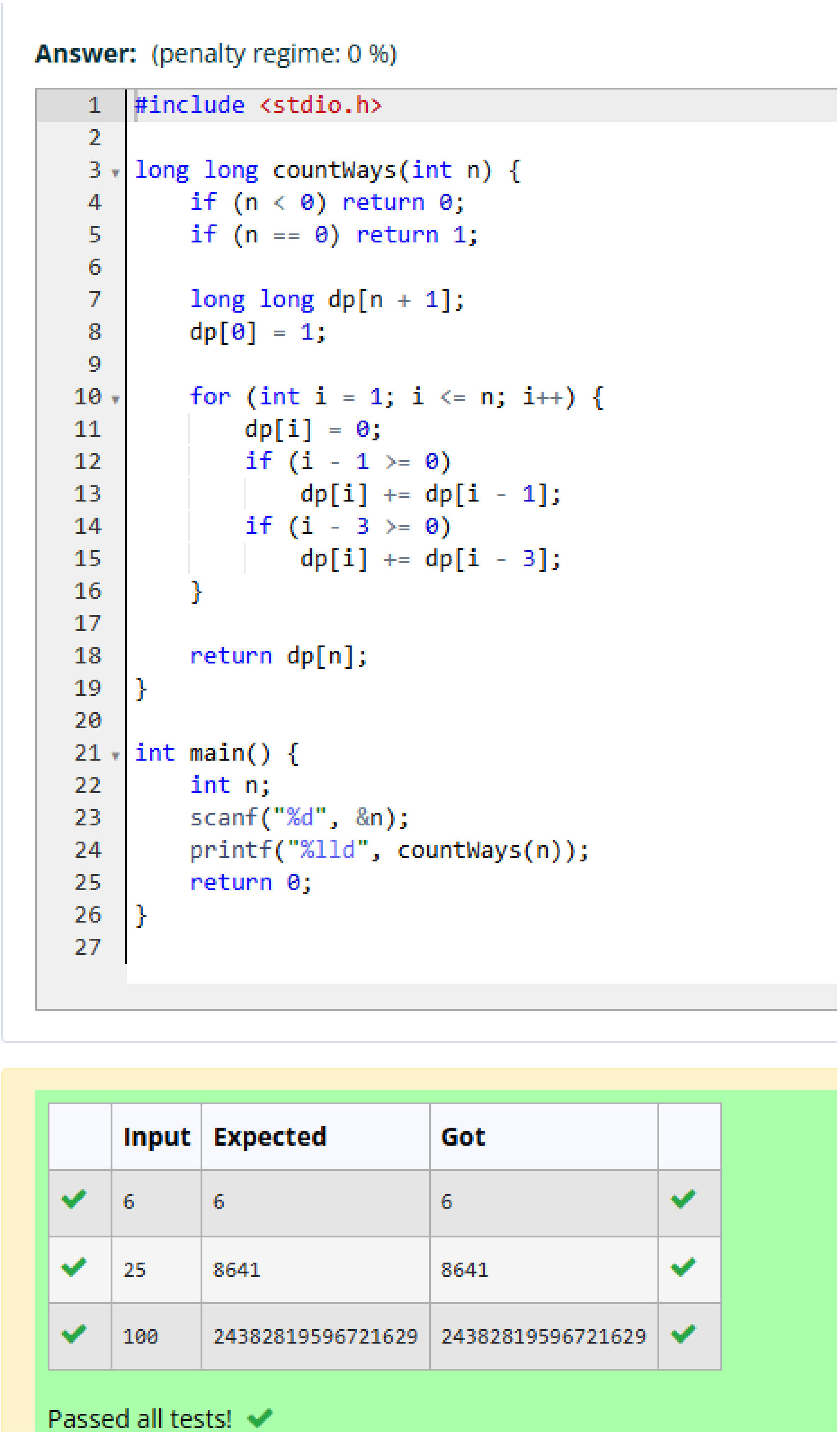
NAME: ROSHAN.A

REG NO: 241801232

DYNAMIC PROGRAMMING

PROGRAM 1: 1



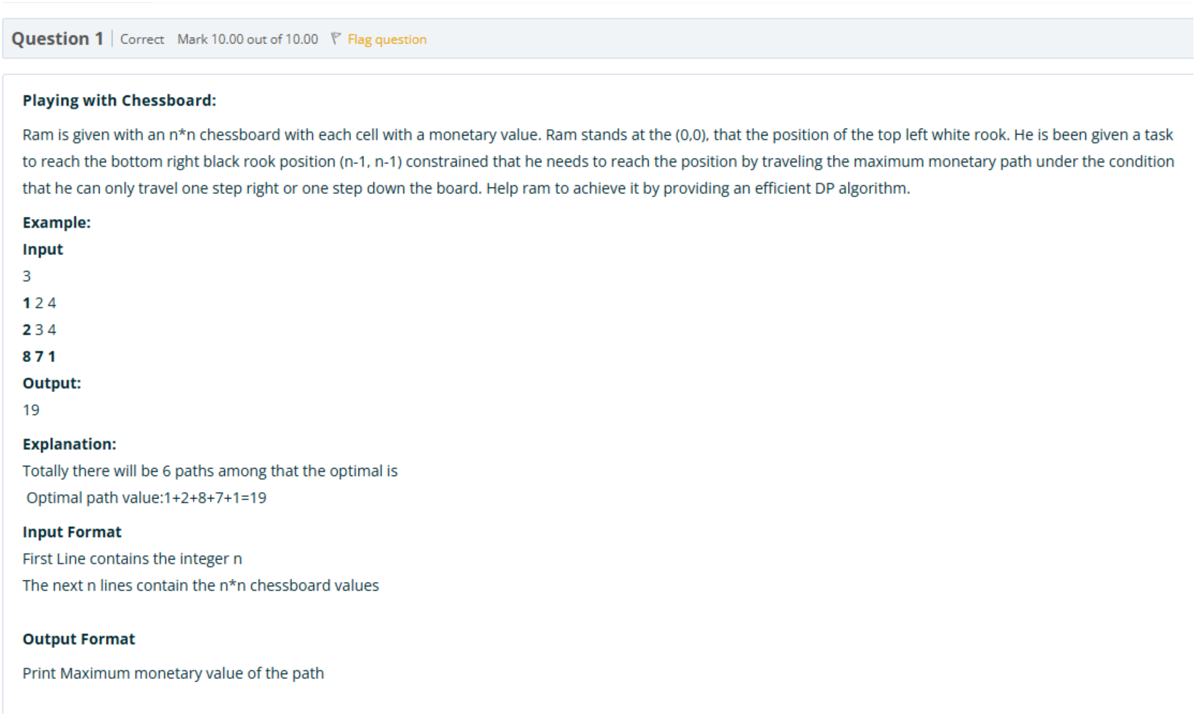
Answer: (penalty regime: 0 %)

```c
#include <stdio.h>

long long countWays(int n) {
    if (n < 0) return 0;
    if (n == 0) return 1;

    long long dp[n + 1];
    dp[0] = 1;

    for (int i = 1; i <= n; i++) {
        dp[i] = 0;
        if (i - 1 >= 0)
            dp[i] += dp[i - 1];
        if (i - 3 >= 0)
            dp[i] += dp[i - 3];
    }

    return dp[n];
}

int main() {
    int n;
    scanf("%d", &n);
    printf("%lld", countWays(n));
    return 0;
}
```

	Input	Expected	Got	
✓	6	6	6	✓
✓	25	8641	8641	✓
✓	100	24382819596721629	24382819596721629	✓

Passed all tests! ✓

PROGRAM 2:



Question 1 | Correct Mark 10.00 out of 10.00 Flag question

Playing with Chessboard:

Ram is given with an n*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

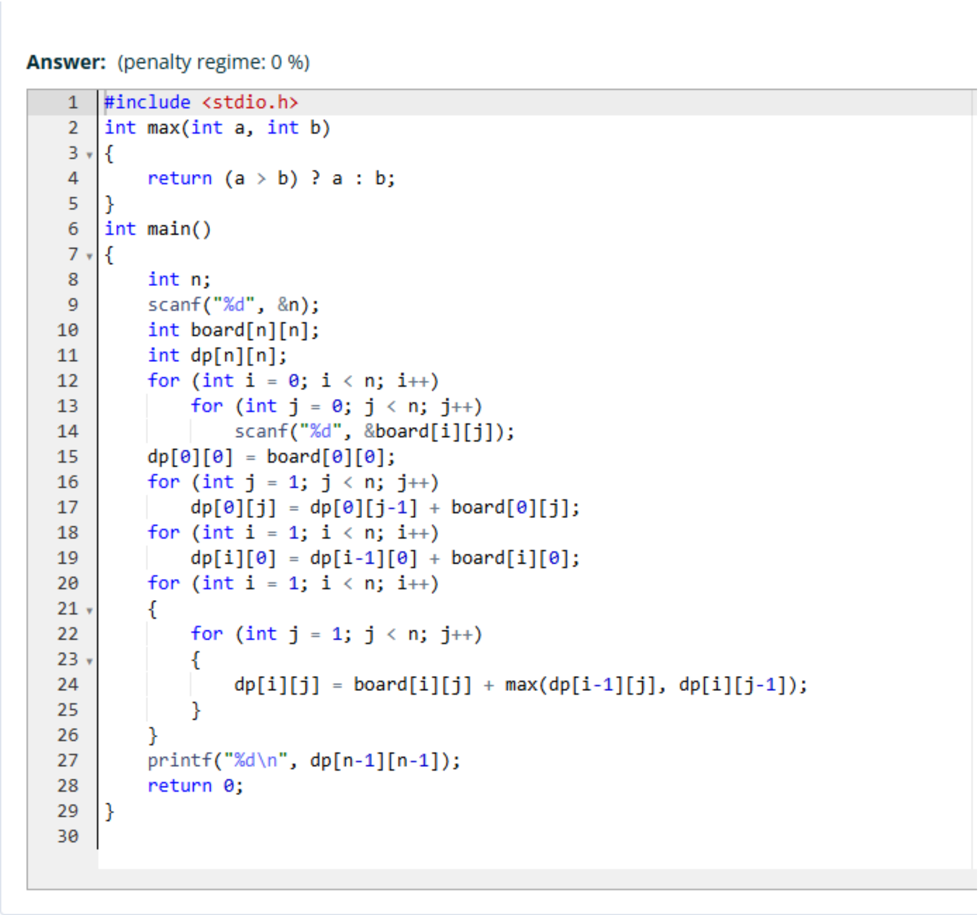
Input Format

First Line contains the integer n

The next n lines contain the n*n chessboard values

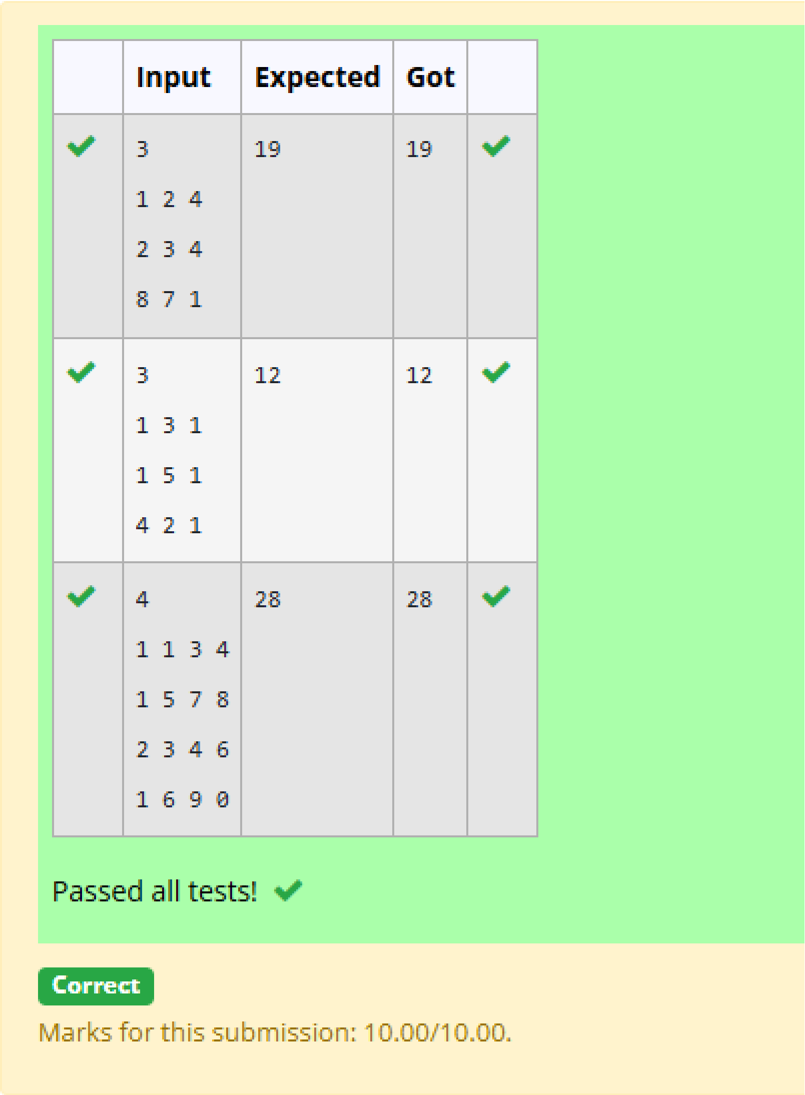
Output Format

Print Maximum monetary value of the path



Answer: (penalty regime: 0 %)

```c
#include <stdio.h>
int max(int a, int b)
{
    return (a > b) ? a : b;
}
int main()
{
    int n;
    scanf("%d", &n);
    int board[n][n];
    int dp[n][n];
    for (int i = 0; i < n; i++)
        for (int j = 0; j < n; j++)
            scanf("%d", &board[i][j]);
    dp[0][0] = board[0][0];
    for (int j = 1; j < n; j++)
        dp[0][j] = dp[0][j-1] + board[0][j];
    for (int i = 1; i < n; i++)
        dp[i][0] = dp[i-1][0] + board[i][0];
    for (int i = 1; i < n; i++)
    {
        for (int j = 1; j < n; j++)
        {
            dp[i][j] = board[i][j] + max(dp[i-1][j], dp[i][j-1]);
        }
    }
    printf("%d\n", dp[n-1][n-1]);
    return 0;
}
```



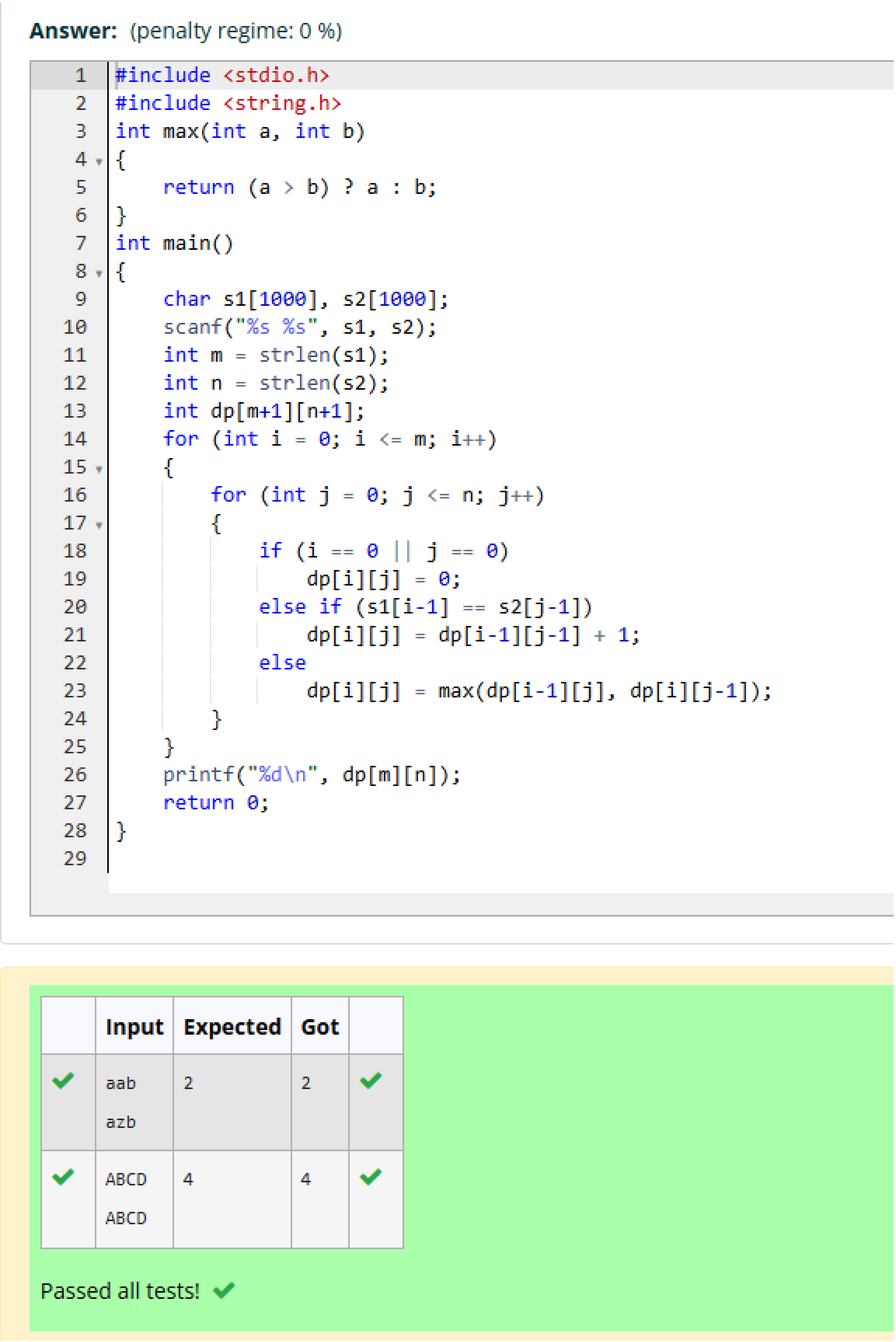
	Input	Expected	Got	
✓	3 1 2 4 2 3 4 8 7 1	19	19	✓
✓	3 1 3 1 1 5 1 4 2 1	12	12	✓
✓	4 1 1 3 4 1 5 7 8 2 3 4 6 1 6 9 0	28	28	✓

Passed all tests! ✓

Correct

Marks for this submission: 10.00/10.00.

PROGRAM 3: 1



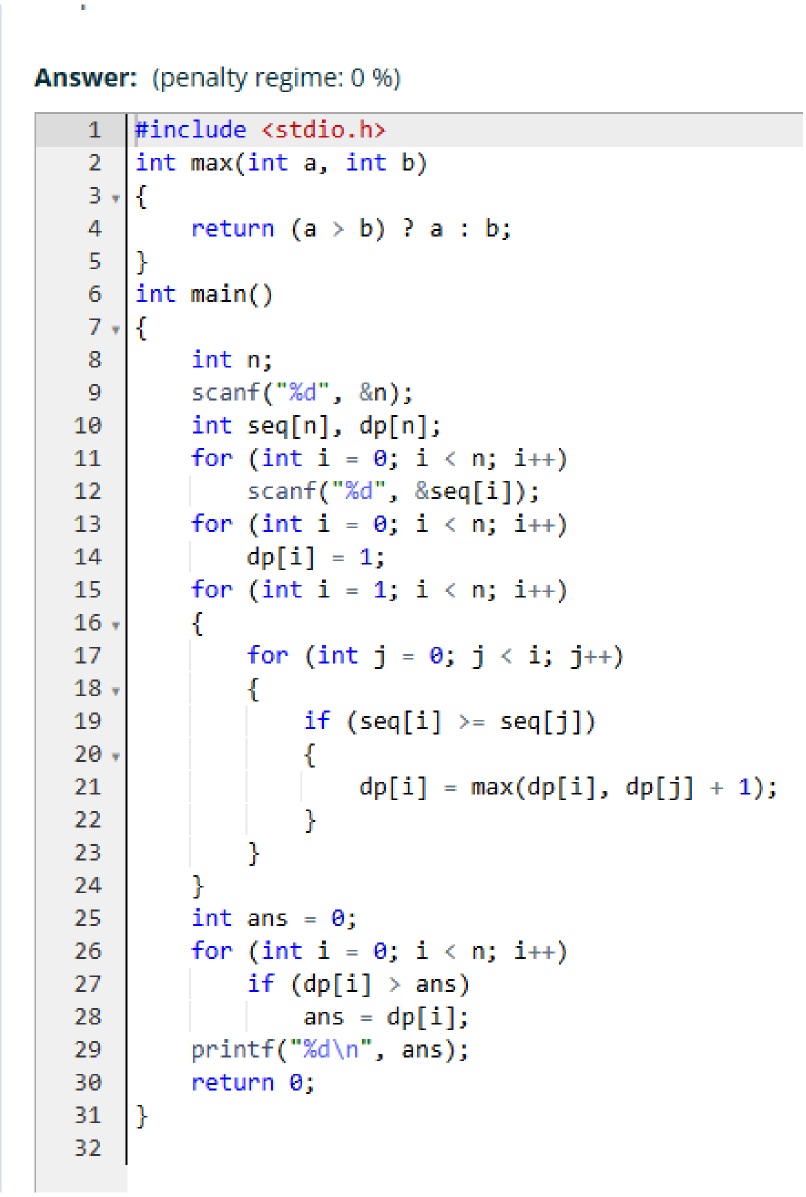
Answer: (penalty regime: 0 %)

```
#include <stdio.h>
#include <string.h>
int max(int a, int b)
{
    return (a > b) ? a : b;
}
int main()
{
    char s1[1000], s2[1000];
    scanf("%s %s", s1, s2);
    int m = strlen(s1);
    int n = strlen(s2);
    int dp[m+1][n+1];
    for (int i = 0; i <= m; i++)
    {
        for (int j = 0; j <= n; j++)
        {
            if (i == 0 || j == 0)
                dp[i][j] = 0;
            else if (s1[i-1] == s2[j-1])
                dp[i][j] = dp[i-1][j-1] + 1;
            else
                dp[i][j] = max(dp[i-1][j], dp[i][j-1]);
        }
    }
    printf("%d\n", dp[m][n]);
    return 0;
}

```

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

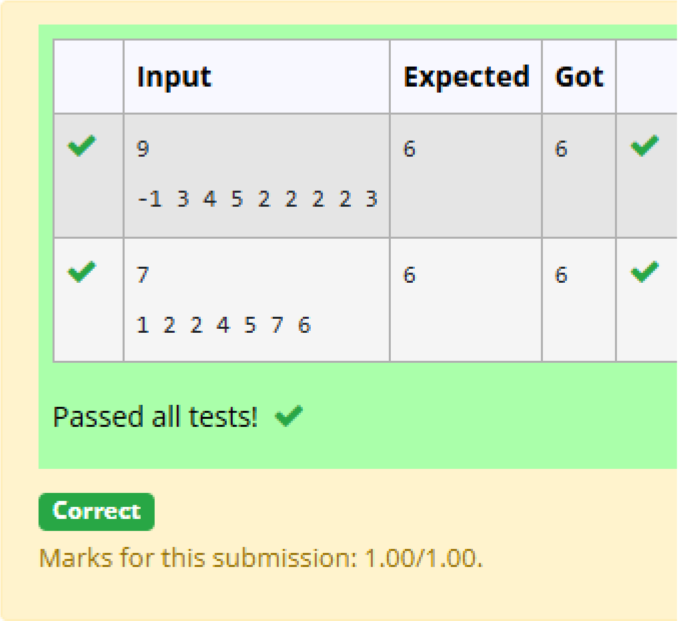
Passed all tests! ✓

PROGRAM 4: 1



Answer: (penalty regime: 0 %)

```c
#include <stdio.h>
int max(int a, int b)
{
    return (a > b) ? a : b;
}
int main()
{
    int n;
    scanf("%d", &n);
    int seq[n], dp[n];
    for (int i = 0; i < n; i++)
        scanf("%d", &seq[i]);
    for (int i = 0; i < n; i++)
        dp[i] = 1;
    for (int i = 1; i < n; i++)
    {
        for (int j = 0; j < i; j++)
        {
            if (seq[i] >= seq[j])
            {
                dp[i] = max(dp[i], dp[j] + 1);
            }
        }
    }
    int ans = 0;
    for (int i = 0; i < n; i++)
        if (dp[i] > ans)
            ans = dp[i];
    printf("%d\n", ans);
    return 0;
}

```



	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.