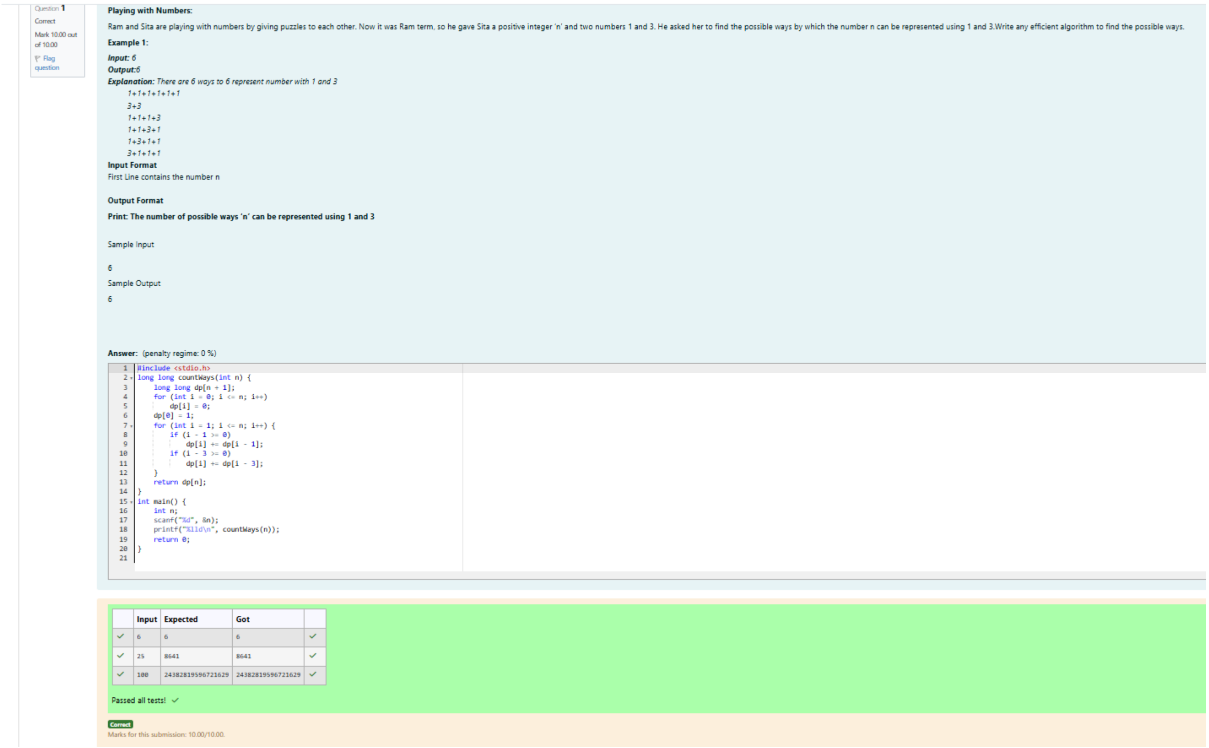
DYNAMIC PROGRAMMING



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 6 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

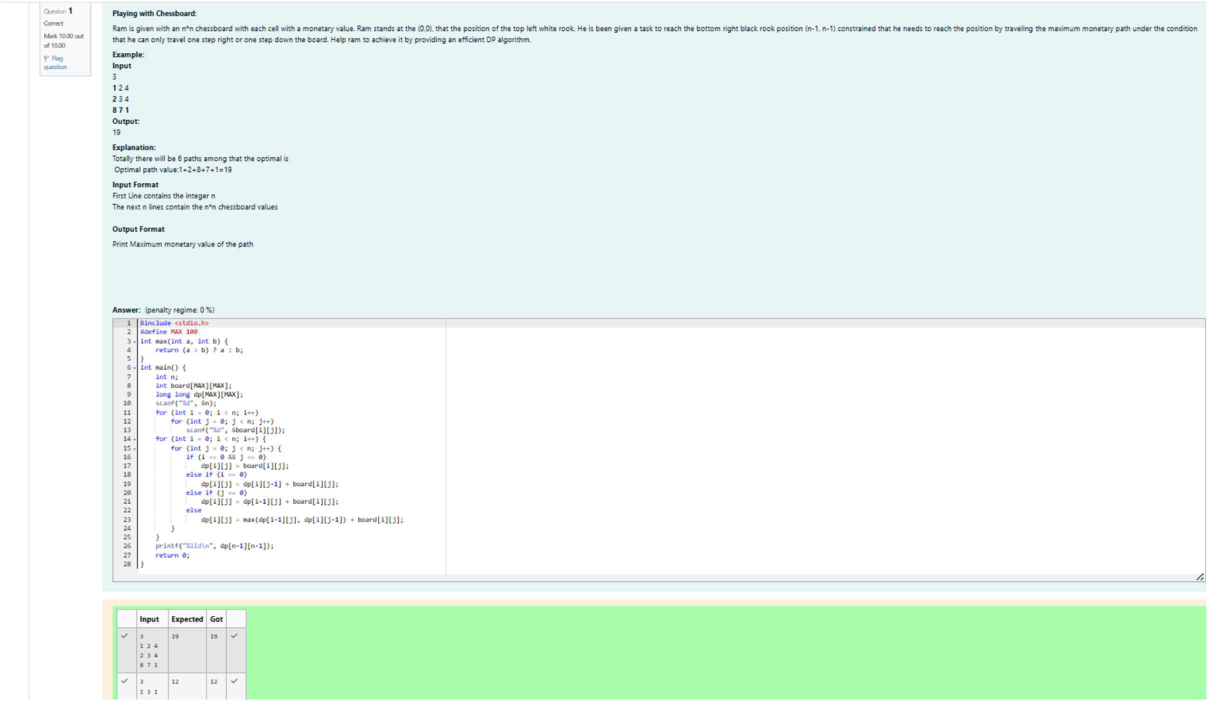
Answer: (penalty regime: 0 %)

```
#include <stdio.h>
long long countWays(int n) {
    long long dp[n + 1];
    for (int i = 0; i <= n; i++)
        dp[i] = 0;
    dp[0] = 1;
    for (int i = 1; i <= n; i++) {
        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\n", countWays(n));
    return 0;
}
```

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

Passed all tests! ✓

Correct
Marks for this submission: 10.00/10.00.



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

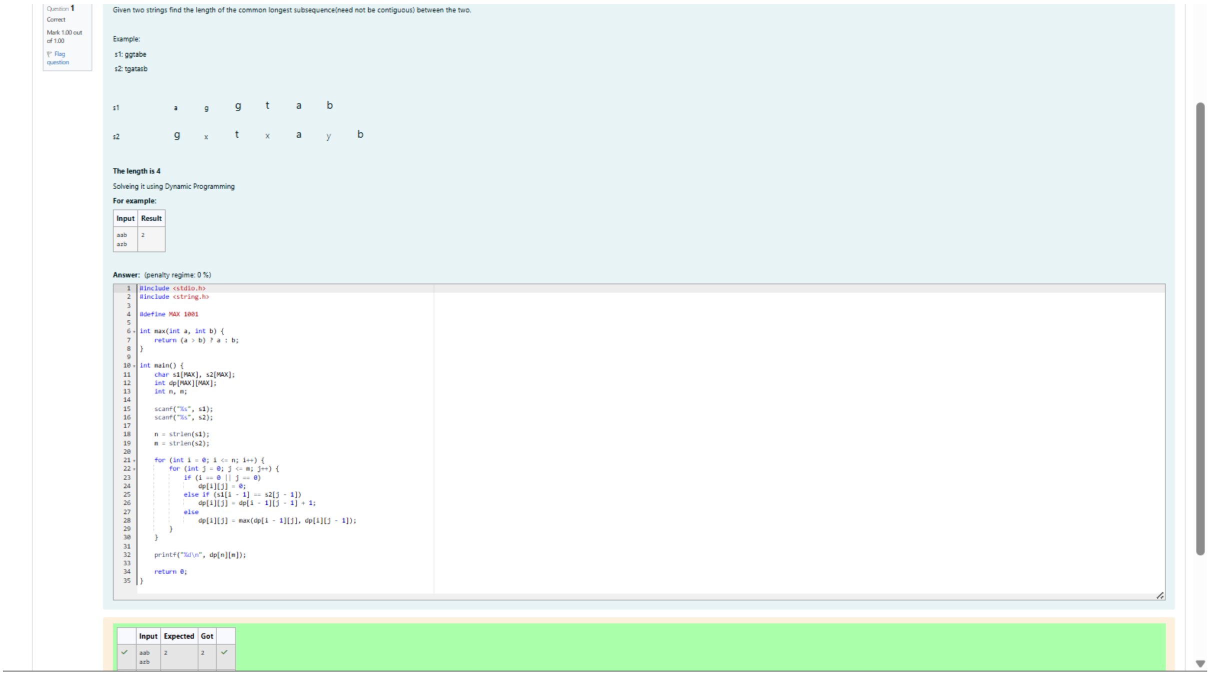
Output Format

Print Maximum monetary value of the path

Answer: (penalty regime: 0 %)

```
#include <stdio.h>
#define MAX 100
int max(int a, int b) {
    return (a > b) ? a : b;
}
int main() {
    int n;
    int board[MAX][MAX];
    long long dp[MAX][MAX];
    scanf("%d", &n);
    for (int i = 0; i < n; i++)
        for (int j = 0; j < n; j++)
            scanf("%d", &board[i][j]);
    for (int i = 0; i < n; i++) {
        for (int j = 0; j < n; j++) {
            if (i == 0 && j == 0)
                dp[i][j] = board[i][j];
            else if (i == 0)
                dp[i][j] = dp[i][j-1] + board[i][j];
            else if (j == 0)
                dp[i][j] = dp[i-1][j] + board[i][j];
            else
                dp[i][j] = max(dp[i-1][j], dp[i][j-1]) + board[i][j];
        }
    }
    printf("%lld\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	12	12	✓



Question **1**

Correct

Mark 1.00 out of 1.00

Flag question

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

Solveing it using Dynamic Programming

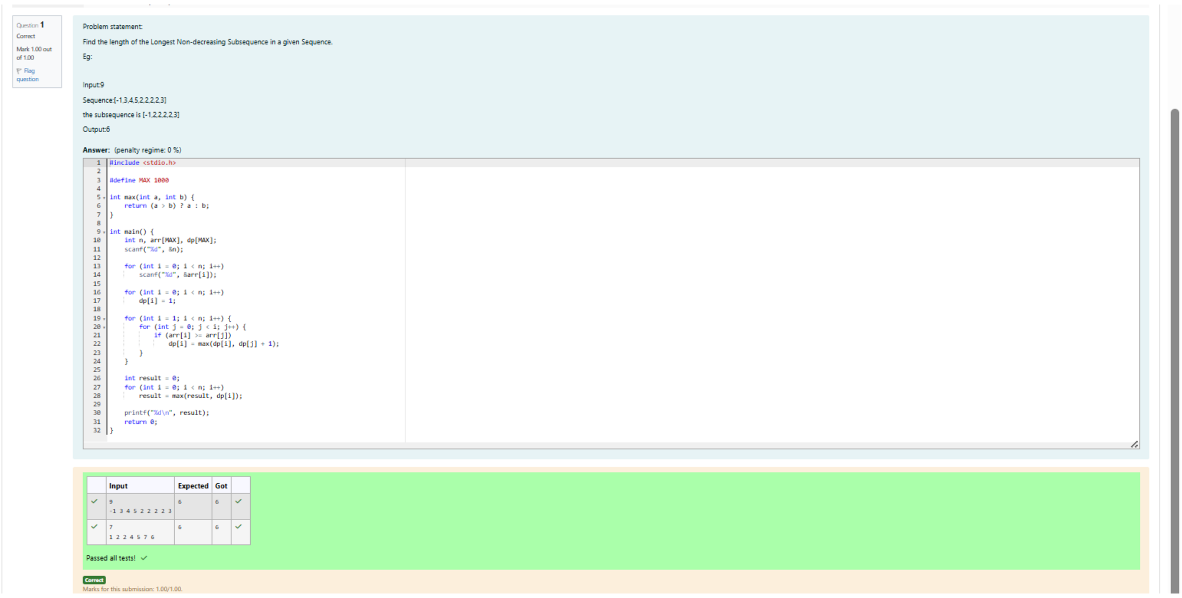
For example:

Input	Result
aab azb	2

Answer: (penalty regime: 0 %)

```
#include <stdio.h>
#include <string.h>

#define MAX 1001

int max(int a, int b) {
    return (a > b) ? a : b;
}

int main() {
    char s1[MAX], s2[MAX];
    int dp[MAX][MAX];
    int n, m;

    scanf("%s", s1);
    scanf("%s", s2);

    n = strlen(s1);
    m = strlen(s2);

    for (int i = 0; i <= n; i++) {
        for (int j = 0; j <= m; 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[n][m]);

    return 0;
}
```

	Input	Expected	Got	
✓	aab azb	2	2	✓



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

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

Output:6

Answer: (penalty regime: 0 %)

```
#include <stdio.h>

#define MAX 1000

int max(int a, int b) {
    return (a > b) ? a : b;
}

int main() {
    int n, arr[MAX], dp[MAX];
    scanf("%d", &n);

    for (int i = 0; i < n; i++)
        scanf("%d", &arr[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 (arr[i] >= arr[j])
                dp[i] = max(dp[i], dp[j] + 1);
        }
    }

    int result = 0;
    for (int i = 0; i < n; i++)
        result = max(result, dp[i]);

    printf("%d\n", result);
    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.