



Dashboard My courses



CS23331-DAA-2024-CSE / 1-DP-Playing with Numbers



1-DP-Playing with Numbers

| Started on | Sunday, 12 October 2025, 10:05 AM |
|--------------|---|
| State | Finished |
| Completed on | Sunday, 12 October 2025, 10:19 AM |
| Time taken | 13 mins 51 secs |
| Grade | 10.00 out of 10.00 (100 %) |

Question 1 | Correct Mark 10.00 out of 10.00 | Flag question

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

Consula laura

```
6
Sample Output
6
```

Answer: (penalty regime: 0 %)

```
1 #include<stdio.h>
2 - long long calculate_ways(int n) {
3 - if(n < 0) {
      if(n == 0) {
       long long dp3 = 1;
      long long dp2 = 1;
      long long dp1 = 1;
      long long current_dp = 0;
       return dp2;
       return dp1;
      for(int i = 3; i <= n; i++) {
       current_dp = dp1 + dp3;
          dp3 = dp2;
           dp1 = current_dp;
       return dp1;
25 - int main() {
      if(scanf("%d", &n) != 1) {
      long long result = calculate_ways(n);
      printf("%lld\n", result);
```

| | Input | Expected | Got | |
|---|-------|-------------------|-------------------|---|
| • | 6 | 6 | 6 | * |
| • | 25 | 8641 | 8641 | v |
| • | 100 | 24382819596721629 | 24382819596721629 | v |

Passed all tests! 🗸

Correct

Marks for this submission: 10.00/10.00.

Finish review

Back to Course

Data retention summary





Dashboard My courses



CS23331-DAA-2024-CSE / 2-DP-Playing with chessboard



2-DP-Playing with chessboard

| Started on | Saturday, 18 October 2025, 8:11 AM |
|--------------|------------------------------------|
| State | Finished |
| Completed on | Saturday, 18 October 2025, 8:19 AM |
| 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*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

124

234

871

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

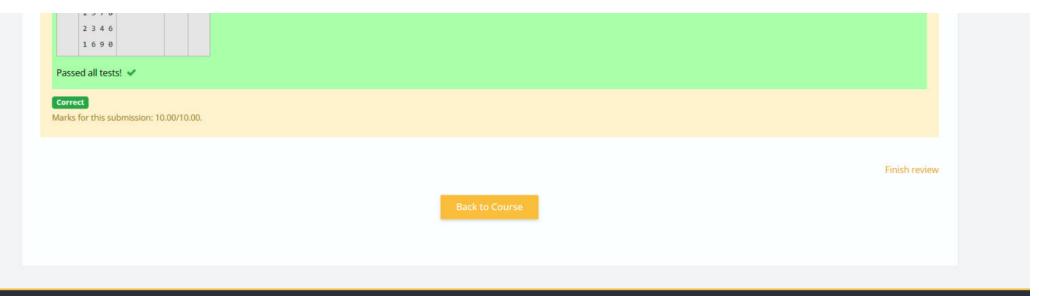
The next n lines contain the n*n chessboard values

Output Format

Print Maximum monetary value of the path

Answer: (penalty regime: 0 %)

| | Input | Expected | Got | |
|---|---------|----------|-----|---|
| ~ | | 19 | 19 | * |
| | 1 2 4 | | | |
| | 8 7 1 | | | |
| ~ | 3 | 12 | 12 | • |
| | 1 3 1 | | | |
| | 151 | | | |
| ~ | | 28 | 28 | ~ |
| | 1 1 3 4 | | | |



Data retention summary



Input Result



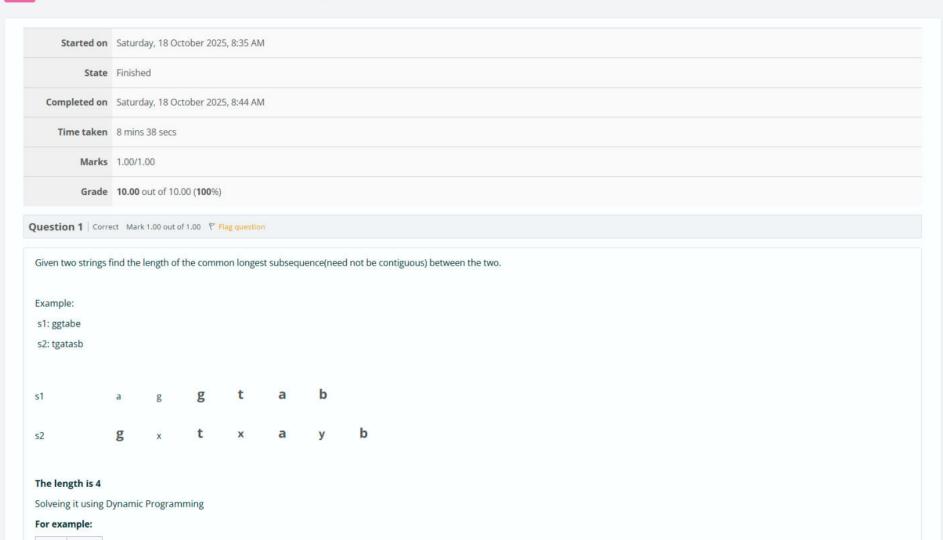
Dashboard My courses



CS23331-DAA-2024-CSE / 3-DP-Longest Common Subsequence



3-DP-Longest Common Subsequence



```
aab 2
azb
```

Answer: (penalty regime: 0 %)

```
1 #include<stdio.h>
2 #include<string.h>
3 + int max(int a, int b) {
       return (a > b) ? a : b;
       char s1[100], s2[100];
scanf("%s", s1);
scanf("%s", s2);
       int m = strlen(s2);
        int dp[n + 1][m + 1];
            dp[i][0] = 0;
        for(int j = 0; j \le m; j++) {
            dp[0][j] = 0;
        for(int i = 1; i <= n; i++) {
            for(int j = 1; j <= m; j++) {
    if(s1[i - 1] == s2[j - 1]) {
                     dp[i][j] = 1 + dp[i - 1][j - 1];
                else {
                     dp[i][j] = max(dp[i - 1][j], dp[i][j - 1]);
       printf("%d", dp[n][m]);
```

| | Input | Expected | Got | |
|---|--------------|----------|-----|---|
| * | aab azb | 2 | 2 | * |
| * | ABCD ABCD | 4 | 4 | * |

Passed all tests! 🗸

Correct

Marks for this submission: 1.00/1.00.

Back to Course

Data retention summar



n Dashboard My courses



CS23331-DAA-2024-CSE / 4-DP-Longest non-decreasing Subsequence



4-DP-Longest non-decreasing Subsequence

| Started on | Saturday, 18 October 2025, 8:47 AM | |
|--|--|--|
| State | Finished | |
| Completed on | Saturday, 18 October 2025, 8:52 AM | |
| Time taken | 5 mins 44 secs | |
| Marks | ss 1.00/1.00 | |
| Grade | 10.00 out of 10.00 (100%) | |
| Question 1 Corre | ect Mark 1.00 out of 1.00 P Flag question | |
| Eg: Input:9 Sequence:[-1,3,4,5] the subsequence Output:6 Answer: (penalty #include 2 int max(3 retu | the Longest Non-decreasing Subsequence in a given Sequence. 5,2,2,2,2,3] is [-1,2,2,2,2,3] regime: 0 %) | |
| 4 } 5 + int main 6 int : 7 | 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.

Finish review

Back to Course

Data retention summary