1-DP-Playing with Numbers

Started on	Friday, 10 October 2025, 2:24 PM
State	Finished
Completed on	Friday, 10 October 2025, 2:50 PM
Time taken	25 mins 38 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

Sample Input

6

Sample Output

6

Answer: (penalty regime: 0 %)

```
#include<stdio.h>
#include<stdint.h>
uint64_t countryWays(int n){
    uint64_t dp[n+1];
    
    dp[0]=1;
   for(int i=1;i<=n;i++){
       dp[i]=0;
        if(i>=1){
            dp[i]+=dp[i-1];
        }if(i>=3){
            dp[i]+=dp[i-3];
        }
    }
    return dp[n];
}
int main(){
    int n;
    scanf("%d",&n);
    uint64_t result=countryWays(n);
    printf("%lu\n",result);
}
```

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

Passed all tests! ✓

Correct

Marks for this submission: 10.00/10.00.

Finish review

Back to Course

Quiz navigation

1 ✓

Finish review

2-DP-Playing with chessboard

Started on	Friday, 10 October 2025, 2:50 PM
State	Finished
Completed on	Thursday, 16 October 2025, 11:31 PM
Time taken	6 days 8 hours
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

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],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]);
        }
    }
    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.

Finish review

Back to Course

Quiz navigation

1 ✓

Finish review

×
- Problem 5: Finding Complex...
- Divide and Conquer
- 1-Number of Zeros in a Give...
- 2-Majority Element
- 3-Finding Floor Value
- 4-Two Elements sum to x
- 5-Implementation of Quick ...
- Greedy Algorithms
- 1-G-Coin Problem
- 2-G-Cookies Problem
- 3-G-Burger Problem
- 4-G-Array Sum max problem
- 5-G-Product of Array eleme...
- Dynamic Programming
- 1-DP-Playing with Numbers
- 2-DP-Playing with chessboard
- 3-DP-Longest Common Sub...
- 4-DP-Longest non-decreasi...
- Competitive Programming
- 1-Finding Duplicates-O(n^2)...
- 2-Finding Duplicates-O(n) Ti...
- 3-Print Intersection of 2 sort...
- 4-Print Intersection of 2 sort...

RAJALAKSHMI ENGINEERING COLLEGE

JAI VARDHAN U L 2024-CSE J2

Dashboard My courses

Quiz navigation
1
Finish review

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

3-DP-Longest Common Subsequence

Started on	Thursday, 16 October 2025, 11:31 PM
State	Finished
Completed on	Thursday, 16 October 2025, 11:41 PM
Time taken	9 mins 57 secs
Marks	1.00/1.00
Grade	**10.00** out of 10.00 (**100**%)

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 100
int max(int a,int b){
    return (a>b)? a:b;
}
int lcs(char *s1,char *s2){
    int n=strlen(s1);
    int m=strlen(s2);
    int dp[MAX][MAX];

    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]=1+dp[i-1][j-1];
            else
             dp[i][j]=max(dp[i-1][j],dp[i][j-1]);
        }
    }
    return dp[n][n];
}
int main(){
    char s1[MAX],s2[MAX];
    scanf("%s", s1);
    scanf("%s", s2);
    printf("%d\n",lcs(s1,s2));
    return 0;
}
```

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

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

Finish review

Back to Course

Quiz navigation

1 ✓

Finish review

4-DP-Longest non-decreasing Subsequence

Started on	Thursday, 16 October 2025, 11:41 PM
State	Finished
Completed on	Thursday, 16 October 2025, 11:52 PM
Time taken	10 mins 51 secs
Marks	1.00/1.00
Grade	**10.00** out of 10.00 (**100**%)

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 %)

```c
#include<stdio.h>
#define MAX 100
int max(int a,int b){
    return (a>b)? a:b;
}
int longNonDecreasingSubsequence(int arr[],int n){
    int dp[MAX];
    int maxLen=1;
    for(int i=0;i<n;i++){
        dp[i]=1;
        for(int j=0;j<i;j++){
            if(arr[j]<=arr[i]){
                dp[i]=max(dp[i],dp[j]+1);
            }
        }
        if(dp[i]>maxLen){
            maxLen=dp[i];
        }
    }
    return maxLen;
}
int main(){
    int n;
    scanf("%d",&n);
    int arr[MAX];
    for(int i=0;i<n;i++){
        scanf("%d",&arr[i]);
    }
    int result=longNonDecreasingSubsequence(arr,n);
    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.

Finish review

Back to Course