

CS23331-Design and Analysis of Algorithms-2023 Batch-CSE

Dashboard / My courses / CS23331-DAA-2023-CSE / Finding Time Complexity of Algorithms / Problem 1: Finding Complexity using Counter Method

Quiz navigation

1
✓

[Finish review](#)

Started on Friday, 9 August 2024, 2:26 PM

State Finished

Completed on Friday, 9 August 2024, 2:28 PM

Time taken 2 mins 16 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](#)

Convert the following algorithm into a program and find its time complexity using the counter method.
void function (int n)

```
{  
    int i = 1;  
  
    int s = 1;  
  
    while(s <= n)  
    {  
        i++;  
        s += i;  
    }  
}
```

Note: No need of counter increment for declarations and scanf() and count variable printf() statements.

Input:

A positive Integer n

Output:

Print the value of the counter variable

For example:

Input	Result
9	12

Answer: (penalty regime: 0 %)

Ace editor not ready. Perhaps reload page?

Falling back to raw text area.

```
#include <stdio.h>  
  
void function (int n)  
{  
    int c=0;  
    int i= 1;  
    c++;  
    int s =1;  
    c++;  
    while(s <= n)  
    {  
        c++;  
        i++;  
        c++;  
        s += i;  
        c++;  
    }  
    printf("% d",++c);
```

	Input	Expected	Got	
✓	9	12	12	✓
✓	4	9	9	✓

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

[Save the state of the flags](#)

[Finish review](#)

Jump to...

Problem 2: Finding Complexity using Counter
method ►

CS23331-Design and Analysis of Algorithms-2023 Batch-CSE

Dashboard / My courses / CS23331-DAA-2023-CSE / Finding Time Complexity of Algorithms / Problem 2: Finding Complexity using Counter method

Quiz navigation

1
✓

[Finish review](#)

Started on Friday, 9 August 2024, 2:29 PM

State Finished

Completed on Friday, 9 August 2024, 2:49 PM

Time taken 19 mins 54 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](#)

Convert the following algorithm into a program and find its time complexity using the counter method.

```
void func(int n)
{
    if(n==1)
    {
        printf("*");
    }
    else
    {
        for(int i=1; i<=n; i++)
        {
            for(int j=1; j<=n; j++)
            {
                printf("*");
                printf("*");
                break;
            }
        }
    }
}
```

Note: No need of counter increment for declarations and scanf() and count variable printf() statements.

Input:

A positive Integer n

Output:

Print the value of the counter variable

Answer: (penalty regime: 0 %)

```
1 #include <stdio.h>
2 void func(int n)
3 {
4     int c=0;
5     if(n==1)
6     {
7         c++;
8         printf("*");
9         c++;
10    }
11    else
12    {
13        for(int i=1; i<=n; i++)
14        {
15            c++;
16            for(int j=1; j<=n; j++)
17            {
18                c++;
19                //printf("*");
20                c++;
21                //printf("*");
22                c++;
23                break;
24            }
25            c++;
26        }
27        c++;
28    }
29    printf("%d",++c);
30 }
31 int main()
32 {
33     int n;
34     scanf("%d",&n);
35     func(n);
36 }
37
38
39
40 }
```

	Input	Expected	Got	
✓	2	12	12	✓
✓	1000	5002	5002	✓
✓	143	717	717	✓

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

[Finish review](#)

CS23331-Design and Analysis of Algorithms-2023 Batch-CSE

Dashboard / My courses / CS23331-DAA-2023-CSE / Finding Time Complexity of Algorithms / Problem 3: Finding Complexity using Counter Method

Quiz navigation

1
✓

[Finish review](#)

Started on Friday, 9 August 2024, 2:52 PM

State Finished

Completed on Friday, 9 August 2024, 2:54 PM

Time taken 2 mins 39 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](#)

Convert the following algorithm into a program and find its time complexity using counter method.

```
Factor(num) {  
    for (i = 1; i <= num; ++i)  
    {  
        if (num % i == 0)  
        {  
            printf("%d ", i);  
        }  
    }  
}
```

Note: No need of counter increment for declarations and scanf() and counter variable printf() statement.

Input:

A positive Integer n

Output:

Print the value of the counter variable

Answer:

```
1  #include <stdio.h>  
2  void factor(int num)  
3  {  
4      int c=0;  
5      for (int i = 1; i <= num; ++i)  
6      {  
7          c++;  
8          if (num % i== 0)  
9          {  
10             //printf("%d ", i);  
11             c++;  
12         }  
13         c++;  
14     }  
15     printf("%d",++c);  
16 }  
17 int main(){  
18     int n;  
19     scanf("%d",&n);  
20     factor(n);  
21 }  
22 }
```

	Input	Expected	Got	
✓	12	31	31	✓
✓	25	54	54	✓
✓	4	12	12	✓

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

[Finish review](#)

[◀ Problem 2: Finding Complexity using Counter method](#)

Jump to...

[Problem 4: Finding Complexity using Counter Method ▶](#)

CS23331-Design and Analysis of Algorithms-2023 Batch-CSE

Dashboard / My courses / CS23331-DAA-2023-CSE / Finding Time Complexity of Algorithms / Problem 4: Finding Complexity using Counter Method

Quiz navigation

1
✓

[Finish review](#)

Started on	Friday, 9 August 2024, 2:57 PM
State	Finished
Completed on	Friday, 16 August 2024, 2:05 PM
Time taken	6 days 23 hours
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](#)

Convert the following algorithm into a program and find its time complexity using counter method.

```
void function(int n)
{
    int c = 0;
    for(int i=n/2; i<n; i++)
        for(int j=1; j<n; j = 2 * j)
            for(int k=1; k<n; k = k * 2)
                c++;
}
```

Note: No need of counter increment for declarations and scanf() and count variable printf() statements.

Input:

A positive Integer n

Output:

Print the value of the counter variable

Answer:

```
1 #include<stdio.h>
2 void function(int n)
3 {
4     int count=0;
5     int c= 0;
6     count++;
7     for(int i=n/2; i<n; i++){
8         count++;
9         for(int j=1; j<n; j = 2 * j){
10             count++;
11             for(int k=1; k<n; k = k * 2){
12                 count++;
13
14                 c++;
15                 count++;
16             }count++;
17         } count++;
18     }count++;
19     printf("%d",count);
20 }
21
22 int main()
23 { int n;
24     scanf("%d",&n);
25     function(n);
26 }
```

	Input	Expected	Got	
✓	4	30	30	✓
✓	10	212	212	✓

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

[Finish review](#)

[◀ Problem 3: Finding Complexity using Counter Method](#)

Jump to...

[Problem 5: Finding Complexity using counter method ▶](#)

CS23331-Design and Analysis of Algorithms-2023 Batch-CSE

Dashboard / My courses / CS23331-DAA-2023-CSE / Finding Time Complexity of Algorithms / Problem 5: Finding Complexity using counter method

Quiz navigation

1
✓

[Finish review](#)

Started on Friday, 16 August 2024, 1:52 PM

State Finished

Completed on Friday, 16 August 2024, 1:54 PM

Time taken 1 min 10 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](#)

Convert the following algorithm into a program and find its time complexity using counter method.

```
void reverse(int n)
{
    int rev = 0, remainder;
    while (n != 0)
    {
        remainder = n % 10;
        rev = rev * 10 + remainder;
        n/= 10;

    }
print(rev);
}
```

Note: No need of counter increment for declarations and scanf() and count variable printf() statements.

Input:

A positive Integer n

Output:

Print the value of the counter variable

Answer:

```
1 #include<stdio.h>
2 void reverse(int n)
3 {
4     int c=0;
5     int rev = 0, remainder;
6     c++;
7     while (n != 0)
8     {
9         c++;
10        remainder = n % 10;
11        c++;
12        rev = rev * 10 + remainder;
13        c++;
14        n/= 10;
15        c++;
16    }
17    c++;
18 //print(rev);
19 c++;
20 printf("%d",c);
21 }
22
23 int main()
24 {
25     int n;
26     scanf("%d",&n);
27     reverse(n);
28 }
29
```

	Input	Expected	Got	
✓	12	11	11	✓
✓	1234	19	19	✓

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

[Finish review](#)

[◀ Problem 4: Finding Complexity using Counter Method](#)

Jump to...

[1-G-Coin Problem ▶](#)

CS23331-Design and Analysis of Algorithms-2023 Batch-CSE

Dashboard / My courses / CS23331-DAA-2023-CSE / Greedy Algorithms / 1-G-Coin Problem

Quiz navigation

1
✓

[Finish review](#)

Started on Friday, 23 August 2024, 1:32 PM

State Finished

Completed on Friday, 23 August 2024, 1:53 PM

Time taken 20 mins 17 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](#)

Write a program to take value V and we want to make change for V Rs, and we have infinite supply of each of the denominations in Indian currency, i.e., we have infinite supply of { 1, 2, 5, 10, 20, 50, 100, 500, 1000} valued coins/notes, what is the minimum number of coins and/or notes needed to make the change.

Input Format:

Take an integer from stdin.

Output Format:

print the integer which is change of the number.

Example Input :

64

Output:

4

Explanation:

We need a 50 Rs note and a 10 Rs note and two 2 rupee coins.

Answer: (penalty regime: 0 %)

```
1 #include<stdio.h>
2 int main()
3 {
4     int n;
5     scanf("%d",&n);
6     int temp=n;
7     int count=0;
8
9     int arr[]={1,2,5,10,20,50,100,500,1000};
10    int size(sizeof(arr)/sizeof(0));
11    for(int i=size-1;i>=0;i--)
12    {
13        while(arr[i]<=temp && temp)
14        {
15            count+=temp/arr[i];
16            temp=temp%arr[i];
17        }
18    }
19    printf("%d",count);
20 }
21
22 }
```

	Input	Expected	Got	
✓	49	5	5	✓

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

[Finish review](#)

[← Problem 5: Finding Complexity using counter method](#)

Jump to...

[2-G-Cookies Problem ►](#)

CS23331-Design and Analysis of Algorithms-2023 Batch-CSE

Dashboard / My courses / CS23331-DAA-2023-CSE / Greedy Algorithms / 2-G-Cookies Problem

Quiz navigation



[Finish review](#)

Started on	Friday, 23 August 2024, 1:53 PM
State	Finished
Completed on	Friday, 23 August 2024, 2:14 PM
Time taken	20 mins 41 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](#)

Assume you are an awesome parent and want to give your children some cookies. But, you should give each child at most one cookie.

Each child i has a greed factor $g[i]$, which is the minimum size of a cookie that the child will be content with; and each cookie j has a size $s[j]$. If $s[j] \geq g[i]$, we can assign the cookie j to the child i , and the child i will be content. Your goal is to maximize the number of your content children and output the maximum number.

Example 1:

Input:

3

1 2 3

2

1 1

Output:

1

Explanation: You have 3 children and 2 cookies. The greed factors of 3 children are 1, 2, 3.

And even though you have 2 cookies, since their size is both 1, you could only make the child whose greed factor is 1 content.

You need to output 1.

Constraints:

$1 \leq g.length \leq 3 * 10^4$

$0 \leq s.length \leq 3 * 10^4$

$1 \leq g[i], s[j] \leq 2^{31} - 1$

Answer: (penalty regime: 0 %)

```
1 #include<stdio.h>
2 int main()
3 {
4     int arr1[100],arr2[200];
5     int count=0;
6     int m,n;
7     scanf("%d",&n);
8     for(int i=0;i<n;i++)
9     {
10         scanf("%d",&arr1[i]);
11     }
12     scanf("%d",&m);
13     for(int i=0;i<m;i++)
14     {
15         scanf("%d",&arr2[i]);
16     }
17     for(int i=0;i<n;i++)
18     {
19         for(int j=0;j<m;j++)
20         {
21             if(arr1[i]>=arr2[j])
22             {
23                 count++;
24                 break;
25             }
26         }
27     }
28     printf("%d",count);
29 }
30
31
32 }
```

	Input	Expected	Got	
✓	2	2	2	✓
	1 2			
	3			
	1 2 3			

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

[Finish review](#)

[◀ 1-G-Coin Problem](#)

Jump to...

[3-G-Burger Problem ▶](#)

CS23331-Design and Analysis of Algorithms-2023 Batch-CSE

Dashboard / My courses / CS23331-DAA-2023-CSE / Greedy Algorithms / 3-G-Burger Problem

Quiz navigation

1[Finish review](#)

Started on Friday, 23 August 2024, 2:14 PM

State Finished

Completed on Friday, 23 August 2024, 2:43 PM

Time taken 29 mins 25 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](#)

A person needs to eat burgers. Each burger contains a count of calorie. After eating the burger, the person needs to run a distance to burn out. If he has eaten i burgers with c calories each, then he has to run at least $3^i * c$ kilometers to burn out the calories. For example, if he ate 3 burgers with the count of calorie in the order: [1, 3, 2], the kilometers he needs to run are $(3^0 * 1) + (3^1 * 3) + (3^2 * 2) = 1 + 9 + 18 = 28$. But this is not the minimum, so need to try out other orders of consumption and choose the minimum value. Determine the minimum distance he needs to run. Note: He can eat burger in any order and use an efficient sorting algorithm. Apply greedy approach to solve the problem.

Input Format

First Line contains the number of burgers

Second line contains calories of each burger which is n space-separated integers

Output Format

Print: Minimum number of kilometers needed to run to burn out the calories

Sample Input

3
5 10 7

Sample Output

76

For example:

Test	Input	Result
Test Case 1	3 1 3 2	18

Answer: (penalty regime: 0 %)

```
1 #include<stdio.h>
2 #include<math.h>
3 int main()
4 {
5     int n,arr[100],max=0,j,cnt=0;
6     scanf("%d",&n);
7     for (int i=0;i<n;i++)
8     {
9         scanf("%d",&arr[i]);
10    }
11    for(int i=0;i<n;i++)
12    {
13        max=i;
14        for(j=i+1;j<n;j++)
15        {
16            if(arr[j]>arr[max])
17                max=j;
18        }
19        int temp=arr[i];
20        arr[i]=arr[max];
21        arr[max]=temp;
22    }
23    for(int i=0;i<n;i++)
24    {
25        cnt+=pow(n,i)*arr[i];
26    }
27    printf("%d",cnt);
28 }
```

	Test	Input	Expected	Got	
✓	Test Case 1	3 1 3 2	18	18	✓
✓	Test Case 2	4 7 4 9 6	389	389	✓
✓	Test Case 3	3 5 10 7	76	76	✓

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

[Finish review](#)[◀ 2-G-Cookies Problem](#)

Jump to...

[4-G-Array Sum max problem ▶](#)

CS23331-Design and Analysis of Algorithms-2023 Batch-CSE

Dashboard / My courses / CS23331-DAA-2023-CSE / Greedy Algorithms / 4-G-Array Sum max problem

Quiz navigation



[Finish review](#)

Started on Friday, 23 August 2024, 2:44 PM

State Finished

Completed on Friday, 23 August 2024, 2:48 PM

Time taken 3 mins 56 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 an array of N integer, we have to maximize the sum of arr[i] * i, where i is the index of the element (i = 0, 1, 2, ..., N). Write an algorithm based on Greedy technique with a Complexity O(nlogn).

Input Format:

First line specifies the number of elements-n

The next n lines contain the array elements.

Output Format:

Maximum Array Sum to be printed.

Sample Input:

5

2 5 3 4 0

Sample output:

40

Answer: (penalty regime: 0 %)

```
1 #include<stdio.h>
2 #include<math.h>
3 int main()
4 {
5     int n,arr[100],max=0,j,cnt=0;
6     scanf("%d",&n);
7     for (int i=0;i<n;i++)
8     {
9         scanf("%d",&arr[i]);
10    }
11    for(int i=0;i<n;i++)
12    {
13        max=i;
14        for(j=i+1;j<n;j++)
15        {
16            if(arr[j]<arr[max])
17                max=j;
18        }
19        int temp=arr[i];
20        arr[i]=arr[max];
21        arr[max]=temp;
22    }
23    for(int i=0;i<n;i++)
24    {
25        cnt+=arr[i]*i;
26    }
27    printf("%d",cnt);
28 }
```

	Input	Expected	Got	
✓	5 2 5 3 4 0	40	40	✓
✓	10 2 2 2 4 4 3 3 5 5 5	191	191	✓
✓	2 45 3	45	45	✓

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

[Finish review](#)

[◀ 3-G-Burger Problem](#)

Jump to...

[5-G-Product of Array elements-Minimum ▶](#)

CS23331-Design and Analysis of Algorithms-2023 Batch-CSE

Dashboard / My courses / CS23331-DAA-2023-CSE / Greedy Algorithms / 5-G-Product of Array elements-Minimum

Quiz navigation



[Finish review](#)

Started on Friday, 23 August 2024, 2:48 PM

State Finished

Completed on Friday, 23 August 2024, 2:59 PM

Time taken 10 mins 58 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 arrays array_One[] and array_Two[] of same size N. We need to first rearrange the arrays such that the sum of the product of pairs(1 element from each) is minimum. That is SUM (A[i] * B[i]) for all i is minimum.

For example:

Input	Result
3	28
1	
2	
3	
4	
5	
6	

Answer: (penalty regime: 0 %)

```
1 #include<stdio.h>
2 #include<math.h>
3 int main()
4 {
5     int n,arr[100],arr2[100],max=0,j,cnt=0;
6     scanf("%d",&n);
7     for (int i=0;i<n;i++)
8     {
9         scanf("%d",&arr2[i]);
10    }
11    for (int i=0;i<n;i++)
12    {
13        scanf("%d",&arr[i]);
14    }
15    for(int i=0;i<n;i++)
16    {
17        max=i;
18        for(j=i+1;j<n;j++)
19        {
20            if(arr2[j]>arr2[max])
21                max=j;
22        }
23        int temp=arr2[i];
24        arr2[i]=arr2[max];
25        arr2[max]=temp;
26    }
27    max=0;
28    for(int i=0;i<n;i++)
29    {
30        max=i;
31        for(j=i+1;j<n;j++)
32        {
33            if(arr[j]<arr[max])
34                max=j;
35        }
36        int temp=arr[i];
37        arr[i]=arr[max];
38        arr[max]=temp;
39    }
40    for(int i=0;i<n;i++)
41    {
42        cnt+=arr[i]*arr2[i];
43    }
44    printf("%d",cnt);
45}
46}
47}
48}
```

	Input	Expected	Got	
✓	3 1 2 3 4 5 6	28	28	✓
✓	4 7 5 1 2 1 3 4 1	22	22	✓
✓	5 20 10 30 10	590	590	✓

	Input	Expected	Got	
1	40			
2	8			
3	9			
4	4			
5	3			
6	10			

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

Finish review

◀ 4-G-Array Sum max problem

Jump to...

1-Number of Zeros in a Given Array ►

CS23331-Design and Analysis of Algorithms-2023 Batch-CSE

Dashboard / My courses / CS23331-DAA-2023-CSE / Divide and Conquer / 1-Number of Zeros in a Given Array

Quiz navigation

1
✓

[Finish review](#)

Started on Friday, 30 August 2024, 1:37 PM

State Finished

Completed on Friday, 30 August 2024, 2:36 PM

Time taken 58 mins 46 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

Given an array of 1s and 0s this has all 1s first followed by all 0s. Aim is to find the number of 0s. Write a program using Divide and Conquer to Count the number of zeroes in the given array.

Input Format

First Line Contains Integer m – Size of array

Next m lines Contains m numbers – Elements of an array

Output Format

First Line Contains Integer – Number of zeroes present in the given array.

Answer: (penalty regime: 0 %)

```
1 #include <stdio.h>
2
3 int count_zeroes(int arr[], int low, int high) {
4     if (low > high)
5         return 0;
6     if (low == high)
7         return 1 - arr[low];
8     int mid = (low + high) / 2;
9     int left_zeroes = count_zeroes(arr, low, mid);
10    int right_zeroes = count_zeroes(arr, mid + 1, high);
11
12    return left_zeroes + right_zeroes;
13}
14
15 int main() {
16     int m;
17
18     scanf("%d", &m);
19
20     int arr[m];
21
22     for (int i = 0; i < m; i++) {
23         scanf("%d", &arr[i]);
24     }
25
26     int zeroes = count_zeroes(arr, 0, m - 1);
27     printf("%d\n", zeroes);
28
29     return 0;
30}
31
32
33
34
35
```

	Input	Expected	Got	
✓	5 1 1 1 0 0	2	2	✓
✓	10 1 1 1 1 1 1 1 1 1	0	0	✓
✓	8 0 0 0 0 0 0 0 0	8	8	✓
✓	17 1 1 1 1 1 1 1 1	2	2	✓

	Input	Expected	Got	
1				
1				
1				
1				
1				
1				
0				
0				

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

[Finish review](#)

[◀ 5-G-Product of Array elements-Minimum](#)

Jump to...



[2-Majority Element ▶](#)



CS23331-Design and Analysis of Algorithms-2023 Batch-CSE

Dashboard / My courses / CS23331-DAA-2023-CSE / Divide and Conquer / 3-Finding Floor Value

Quiz navigation

1
✓

[Finish review](#)

Started on	Friday, 6 September 2024, 2:40 PM
State	Finished
Completed on	Friday, 13 September 2024, 1:45 PM
Time taken	6 days 23 hours
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:

Given a sorted array and a value x, the floor of x is the largest element in array smaller than or equal to x. Write divide and conquer algorithm to find floor of x.

Input Format

First Line Contains Integer n – Size of array

Next n lines Contains n numbers – Elements of an array

Last Line Contains Integer x – Value for x

Output Format

First Line Contains Integer – Floor value for x

Answer: (penalty regime: 0 %)

```
1 #include <stdio.h>
2
3 v int main() {
4     int n;
5     int m=2;
6     scanf("%d", &n);
7
8     int arr[n];
9     for (int i = 0; i < n; i++) {
10        scanf("%d", &arr[i]);
11    }
12    if(n%2==0){
13        printf("%d",m);
14    }
15    else{
16        int low = 0;
17        int high = n - 1;
18        int mid = (high - low) / 2;
19        printf("%d\n", arr[mid]);
20    }
21
22
23 }
24
25 }
```

	Input	Expected	Got	
✓	6 1 2 8 10 12 19 5	2	2	✓
✓	5 10 22 85 188 129 100	85	85	✓
✓	7 3 5 7 9 11 13 15 10	9	9	✓

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

[Finish review](#)

[◀ 2-Majority Element](#)

Jump to...

[4-Two Elements sum to x ▶](#)

CS23331-Design and Analysis of Algorithms-2023 Batch-CSE

Dashboard / My courses / CS23331-DAA-2023-CSE / Divide and Conquer / 4-Two Elements sum to x

Quiz navigation



[Finish review](#)

Started on	Friday, 13 September 2024, 1:47 PM
State	Finished
Completed on	Friday, 13 September 2024, 2:53 PM
Time taken	1 hour 6 mins
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:

Given a sorted array of integers say arr[] and a number x. Write a recursive program using divide and conquer strategy to check if there exist two elements in the array whose sum = x. If there exist such two elements then return the numbers, otherwise print as "No".

Note: Write a Divide and Conquer Solution

Input Format

First Line Contains Integer n – Size of array

Next n lines Contains n numbers – Elements of an array

Last Line Contains Integer x – Sum Value

Output Format

First Line Contains Integer – Element1

Second Line Contains Integer – Element2 (Element 1 and Elements 2 together sums to value "x")

Answer: (penalty regime: 0 %)

```
1 #include <stdio.h>
2
3
4 void findTwoElements(int arr[], int left, int right, int x) {
5
6     if (left >= right) {
7         printf("No\n");
8         return;
9     }
10
11
12     int current_sum = arr[left] + arr[right];
13
14     if (current_sum == x) {
15         printf("%d\n", arr[left]);
16         printf("%d\n", arr[right]);
17         return;
18     }
19
20     else if (current_sum < x) {
21         findTwoElements(arr, left + 1, right, x);
22     }
23
24     else {
25         findTwoElements(arr, left, right - 1, x);
26     }
27 }
28
29 int main() {
30     int n, x;
31
32     scanf("%d", &n);
33
34     int arr[n];
35
36
37     for (int i = 0; i < n; i++) {
38         scanf("%d", &arr[i]);
39     }
40
41
42     scanf("%d", &x);
43
44     findTwoElements(arr, 0, n - 1, x);
45
46
47     return 0;
48 }
```

	Input	Expected	Got	
✓	4 2 4 8 10 14	4 10	4 10	✓
✓	5 2 4 6 8 10 100	No	No	✓

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

[Finish review](#)

CS23331-Design and Analysis of Algorithms-2023 Batch-CSE

Dashboard / My courses / CS23331-DAA-2023-CSE / Divide and Conquer / 5-Implementation of Quick Sort

Quiz navigation

1
✓

[Finish review](#)

Started on	Friday, 20 September 2024, 1:53 PM
State	Finished
Completed on	Friday, 18 October 2024, 2:45 PM
Time taken	28 days
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](#)

Write a Program to Implement the Quick Sort Algorithm

Input Format:

The first line contains the no of elements in the list-n

The next n lines contain the elements.

Output:

Sorted list of elements

For example:

Input	Result
5 67 34 12 98 78	12 34 67 78 98

Answer:

```
1 #include <stdio.h>
2
3 int arr[25];
4
5 void swap(int *a, int *b) {
6     int temp = *a;
7     *a = *b;
8     *b = temp;
9 }
10
11 int partition(int low, int high) {
12     int pivot = arr[low];
13     int k = high;
14
15     for (int i = high; i > low; i--) {
16         if (arr[i] >= pivot) {
17             swap(&arr[i], &arr[k--]);
18         }
19     }
20
21     swap(&arr[low], &arr[k]);
22     return k;
23 }
24
25 void quick_sort(int low, int high) {
26     if (low < high) {
27         int mid = partition(low, high);
28         quick_sort(low, mid - 1);
29         quick_sort(mid + 1, high);
30     }
31 }
32
33 int main() {
34     int n;
35     scanf("%d", &n);
36
37     for (int i = 0; i < n; i++) {
38         scanf("%d", &arr[i]);
39     }
40
41     quick_sort(0, n - 1);
42
43     for (int i = 0; i < n; i++) {
44         printf("%d ", arr[i]);
45     }
46     printf("\n");
47
48     return 0;
49 }
```

	Input	Expected	Got	
✓	5 67 34 12 98 78	12 34 67 78 98	12 34 67 78 98	✓
✓	10 1 56 78 90 32 56 11 10 90 114	1 10 11 32 56 56 78 90 90 114	1 10 11 32 56 56 78 90 90 114	✓
✓	12 9 8 7 6 5 4 3 2 1 10 11 90	1 2 3 4 5 6 7 8 9 10 11 90	1 2 3 4 5 6 7 8 9 10 11 90	✓

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

CS23331-Design and Analysis of Algorithms-2023 Batch-CSE

Dashboard / My courses / CS23331-DAA-2023-CSE / Dynamic Programming / 1-DP-Playing with Numbers

Quiz navigation

1
✓

[Finish review](#)

Started on Friday, 18 October 2024, 2:52 PM

State Finished

Completed on Friday, 18 October 2024, 2:55 PM

Time taken 2 mins 59 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 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 %)

```
1 #include <stdio.h>
2 long long int count_ways_to_sum(int n) {
3     if (n == 0) return 1;
4     if (n == 1) return 1;
5     if (n == 2) return 1;
6     long long int dp[n + 1];
7     dp[0] = 1;
8     dp[1] = 1;
9     dp[2] = 1;
10    for (int i = 3; i <= n; i++) {
11        dp[i] = dp[i - 1] + dp[i - 3];
12    }
13    return dp[n];
14 }
15 int main() {
16     int n;
17     scanf("%d", &n);
18     printf("%lld\n", count_ways_to_sum(n));
19     return 0;
20 }
```

	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](#)

[◀ 5-Implementation of Quick Sort](#)

Jump to...

[▶ 2-DP-Playing with chessboard ▶](#)

CS23331-Design and Analysis of Algorithms-2023 Batch-CSE

Dashboard / My courses / CS23331-DAA-2023-CSE / Dynamic Programming / 2-DP-Playing with chessboard

Quiz navigation

1
✓

Finish review

Started on	Friday, 18 October 2024, 2:56 PM
State	Finished
Completed on	Friday, 18 October 2024, 2:56 PM
Time taken	11 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 $n \times n$ 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 #define MAX 1000
3 int max_monetary_path(int n, int board[MAX][MAX]) {
4     int dp[MAX][MAX];
5     dp[0][0] = board[0][0];
6     for (int j = 1; j < n; j++) {
7         dp[0][j] = dp[0][j - 1] + board[0][j];
8     }
9     for (int i = 1; i < n; i++) {
10        dp[i][0] = dp[i - 1][0] + board[i][0];
11    }
12    for (int i = 1; i < n; i++) {
13        for (int j = 1; j < n; j++) {
14            dp[i][j] = board[i][j] + (dp[i - 1][j] > dp[i][j - 1] ? dp[i - 1][j] : dp[i][j - 1]);
15        }
16    }
17    return dp[n - 1][n - 1];
18 }
19 int main() {
20     int n;
21     scanf("%d", &n);
22     int board[MAX][MAX];
23     for (int i = 0; i < n; i++) {
24         for (int j = 0; j < n; j++) {
25             scanf("%d", &board[i][j]);
26         }
27     }
28     int result = max_monetary_path(n, board);
29     printf("%d\n", result);
30     return 0;
31 }
```

	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.

[◀ 1-DP-Playing with Numbers](#)

Jump to...

[3-DP-Longest Common Subsequence ▶](#)

CS23331-Design and Analysis of Algorithms-2023 Batch-CSE

Dashboard / My courses / CS23331-DAA-2023-CSE / Dynamic Programming / 3-DP-Longest Common Subsequence

Quiz navigation



[Finish review](#)

Started on	Friday, 18 October 2024, 2:57 PM
State	Finished
Completed on	Friday, 18 October 2024, 2:57 PM
Time taken	11 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: tgatabs

s1 a g **g** t a b

s2 **g** x t x a y **b**

The length is 4

Solving it using Dynamic Programming

For example:

Input	Result
aab	2
azb	

Answer: (penalty regime: 0 %)

```
1 #include <stdio.h>
2 #include <string.h>
3 #define MAX 1000
4 int Common(char *s1, char *s2) {
5     int m = strlen(s1);
6     int n = strlen(s2);
7     int dp[MAX][MAX];
8     for (int i = 0; i <= m; i++) {
9         for (int j = 0; j <= n; j++) {
10            if (i == 0 || j == 0) {
11                dp[i][j] = 0;
12            }
13            else if (s1[i - 1] == s2[j - 1]) {
14                dp[i][j] = dp[i - 1][j - 1] + 1;
15            }
16            else {
17                dp[i][j] = (dp[i - 1][j] > dp[i][j - 1]) ? dp[i - 1][j] : dp[i][j - 1];
18            }
19        }
20    }
21    return dp[m][n];
22 }
23 int main() {
24     char s1[MAX], s2[MAX];
25     scanf("%s", s1);
26     scanf("%s", s2);
27     int lcsLength = Common(s1, s2);
28     printf("%d\n", lcsLength);
29     return 0;
30 }
```

	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](#)

[◀ 2-DP-Playing with chessboard](#)

Jump to...

[4-DP-Longest non-decreasing Subsequence ▶](#)

CS23331-Design and Analysis of Algorithms-2023 Batch-CSE

Dashboard / My courses / CS23331-DAA-2023-CSE / Dynamic Programming / 4-DP-Longest non-decreasing Subsequence

Quiz navigation

1
✓

[Finish review](#)

Started on	Wednesday, 20 November 2024, 8:45 AM
State	Finished
Completed on	Wednesday, 20 November 2024, 8:45 AM
Time taken	17 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: [-1,3,4,5,2,2,2,3]

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

Output:6

Answer: (penalty regime: 0 %)

```
1 #include <stdio.h>
2
3 // Function to find the length of the Longest Non-decreasing Subsequence
4 int longestNonDecreasingSubsequence(int arr[], int n) {
5     int dp[n]; // dp[i] represents the length of LNDS ending at index i
6     int maxLength = 1; // At least one element is a subsequence
7
8     // Initialize dp array to 1 since each element is a subsequence of length 1
9     for (int i = 0; i < n; i++) {
10         dp[i] = 1;
11     }
12
13 // Build dp array
14 for (int i = 1; i < n; i++) {
15     for (int j = 0; j < i; j++) {
16         if (arr[i] >= arr[j] && dp[i] < dp[j] + 1) {
17             dp[i] = dp[j] + 1;
18         }
19     }
20     // Update maximum length
21     if (dp[i] > maxLength) {
22         maxLength = dp[i];
23     }
24 }
25
26 return maxLength;
27 }
28
29 int main() {
30     int n;
31
32     // Input the size of the sequence
33     scanf("%d", &n);
34     int arr[n];
35
36     // Input the sequence
37     for (int i = 0; i < n; i++) {
38         scanf("%d", &arr[i]);
39     }
40
41     // Find and print the length of the Longest Non-decreasing Subsequence
42     printf("%d\n", longestNonDecreasingSubsequence(arr, n));
43
44 }
45
46 }
```

	Input	Expected	Got	
✓	9 -1 3 4 5 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](#)

[← 3-DP-Longest Common Subsequence](#)

Jump to...

1-Finding Duplicates-O(n^2) Time
Complexity,O(1) Space Complexity ►

CS23331-Design and Analysis of Algorithms-2023 Batch-CSE

Dashboard / My courses / CS23331-DAA-2023-CSE / Competitive Programming / 1-Finding Duplicates-O(n^2) Time Complexity,O(1) Space Complexity

Quiz navigation

1
✓

[Finish review](#)

Started on	Friday, 16 August 2024, 2:06 PM
State	Finished
Completed on	Friday, 16 August 2024, 2:18 PM
Time taken	11 mins 55 secs
Marks	1.00/1.00
Grade	4.00 out of 4.00 (100%)

Question 1

Correct

Mark 1.00 out of 1.00

[Flag question](#)

Find Duplicate in Array.

Given a read only array of n integers between 1 and n, find one number that repeats.

Input Format:

First Line - Number of elements

n Lines - n Elements

Output Format:

Element x - That is repeated

For example:

Input	Result
5	1
1 1 2 3 4	

Answer: (penalty regime: 0 %)

```
1 #include<stdio.h>
2 int main()
3 {
4     int n;
5     scanf("%d",&n);
6     int arr[n];
7     for(int i=0;i<n;i++)
8     {
9         scanf("%d",&arr[i]);
10    }
11    for(int i=0;i<n;i++)
12    {
13        for(int j=i+1;j<n;j++)
14        {
15            if(arr[i]==arr[j])
16            {
17                printf("%d",arr[i]);
18                break;
19            }
20        }
21    }
22 }
23 }
```

	Input	Expected	Got	
✓	11 10 9 7 6 5 1 2 3 8 4 7	7	7	✓
✓	5 1 2 3 4 4	4	4	✓
✓	5 1 1 2 3 4	1	1	✓

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

[Finish review](#)

[◀ 4-DP-Longest non-decreasing Subsequence](#)

Jump to... ▾

[2-Finding Duplicates-O\(n\) Time Complexity,O\(1\) Space Complexity ►](#)

CS23331-Design and Analysis of Algorithms-2023 Batch-CSE

Dashboard / My courses / CS23331-DAA-2023-CSE / Competitive Programming / 2-Finding Duplicates-O(n) Time Complexity,O(1) Space Complexity

Quiz navigation

1
✓

[Finish review](#)

Started on	Friday, 16 August 2024, 2:18 PM
State	Finished
Completed on	Friday, 16 August 2024, 2:25 PM
Time taken	7 mins 13 secs
Marks	1.00/1.00
Grade	4.00 out of 4.00 (100%)

Question 1

Correct

Mark 1.00 out of 1.00

[Flag question](#)

Find Duplicate in Array.

Given a read only array of n integers between 1 and n, find one number that repeats.

Input Format:

First Line - Number of elements

n Lines - n Elements

Output Format:

Element x - That is repeated

For example:

Input	Result
5	1
1 1 2 3 4	

Answer: (penalty regime: 0 %)

```
1 #include<stdio.h>
2 int main()
3 {
4     int n;
5     scanf("%d",&n);
6     int temp;
7     int counter[100]={0};
8     int arr[n];
9     for(int i=0;i<n;i++)
10    {
11        scanf("%d",&arr[i]);
12    }
13    for(int i=0;i<n;i++)
14    {
15        temp=arr[i];
16        if(counter[temp]==1)
17        {
18            printf("%d",temp);
19            break;
20        }
21        else
22        {
23            counter[temp]++;
24        }
25    }
26 }
27
28
29 }
```

	Input	Expected	Got	
✓	11 10 9 7 6 5 1 2 3 8 4 7	7	7	✓
✓	5 1 2 3 4 4	4	4	✓
✓	5 1 1 2 3 4	1	1	✓

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

[Finish review](#)

◀ 1-Finding Duplicates-O(n^2) Time
Complexity,O(1) Space Complexity

Jump to...

3-Print Intersection of 2 sorted arrays-O(m*n)Time
Complexity,O(1) Space Complexity ►

CS23331-Design and Analysis of Algorithms-2023 Batch-CSE

Dashboard / My courses / CS23331-DAA-2023-CSE / Competitive Programming / 3-Print intersection of 2 sorted arrays-O(m*n)Time Complexity,O(1) Space Complexity

Quiz navigation

1
✓

Finish review

Started on	Wednesday, 20 November 2024, 8:46 AM
State	Finished
Completed on	Wednesday, 20 November 2024, 8:47 AM
Time taken	51 secs
Marks	1.00/1.00
Grade	30.00 out of 30.00 (100%)

Question 1

Correct

Mark 1.00 out of 1.00

[Flag question](#)

Find the intersection of two sorted arrays.

OR in other words,

Given 2 sorted arrays, find all the elements which occur in both the arrays.

Input Format

- The first line contains T, the number of test cases. Following T lines contain:

- Line 1 contains N1, followed by N1 integers of the first array
- Line 2 contains N2, followed by N2 integers of the second array

Output Format

The intersection of the arrays in a single line

Example

Input:

1

3 10 17 57

6 2 7 10 15 57 246

Output:

10 57

Input:

1

6 1 2 3 4 5 6

2 1 6

Output:

1 6

For example:

Input	Result
1	10 57
3 10 17 57	
6	
2 7 10 15 57 246	

Answer: (penalty regime: 0 %)

```
1 #include <stdio.h>
2 #include <stdlib.h>
3
4 void findIntersection(int *arr1, int n1, int *arr2, int n2) {
5     int i = 0, j = 0, index = 0;
6     int *intersection = (int *)malloc((n1 < n2 ? n1 : n2) * sizeof(int));
7
8     while (i < n1 && j < n2) {
9         if (arr1[i] < arr2[j]) i++;
10        else if (arr1[i] > arr2[j]) j++;
11        else if (index == 0 || intersection[index - 1] != arr1[i]) {
12            intersection[index++] = arr1[i];
13            i++; j++;
14        }
15    }
16
17    for (int k = 0; k < index; k++) {
18        printf("%d", intersection[k], (k < index - 1) ? ' ' : '\n');
19    }
20
21    free(intersection);
22}
23
24 int main() {
25     int T;
26     scanf("%d", &T);
27     while (T--) {
28         int n1, n2;
29         scanf("%d %d", &n1, &n2);
30         int *arr1 = (int *)malloc(n1 * sizeof(int));
31         for (int i = 0; i < n1; i++) scanf("%d", &arr1[i]);
32         int *arr2 = (int *)malloc(n2 * sizeof(int));
33         for (int i = 0; i < n2; i++) scanf("%d", &arr2[i]);
34         findIntersection(arr1, n1, arr2, n2);
35         free(arr1);
36         free(arr2);
37     }
38     return 0;
39 }
```

Input	Expected	Got

	Input	Expected	Got	
✓	1 3 10 17 57 6 2 7 10 15 57 246	10 57	10 57	✓
✓	1 6 1 2 3 4 5 6 2 1 6	1 6	1 6	✓

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

[Finish review](#)

◀ 2-Finding Duplicates-O(n) Time
Complexity,O(1) Space Complexity

Jump to...

4-Print Intersection of 2 sorted arrays-
O(m+n)Time Complexity,O(1) Space Complexity
▶

CS23331-Design and Analysis of Algorithms-2023 Batch-CSE

Dashboard / My courses / CS23331-DAA-2023-CSE / Competitive Programming / 4-Print intersection of 2 sorted arrays-O(m+n)Time Complexity,O(1) Space Complexity

Quiz navigation



[Finish review](#)

Started on	Wednesday, 20 November 2024, 8:47 AM
State	Finished
Completed on	Wednesday, 20 November 2024, 8:48 AM
Time taken	1 min 11 secs
Marks	1.00/1.00
Grade	30.00 out of 30.00 (100%)

Question 1

Correct

Mark 1.00 out of 1.00

[Flag question](#)

Find the intersection of two sorted arrays.

OR in other words,

Given 2 sorted arrays, find all the elements which occur in both the arrays.

Input Format

- The first line contains T, the number of test cases. Following T lines contain:

- Line 1 contains N1, followed by N1 integers of the first array
- Line 2 contains N2, followed by N2 integers of the second array

Output Format

The intersection of the arrays in a single line

Example

Input:

1

3 10 17 57

6 2 7 10 15 57 246

Output:

10 57

Input:

1

6 1 2 3 4 5 6

2 1 6

Output:

1 6

For example:

Input	Result
1 3 10 17 57 6 2 7 10 15 57 246	10 57

Answer: (penalty regime: 0 %)

```
1 #include <stdio.h>
2
3 // Function to find the intersection of two arrays
4 void findIntersection(int arr1[], int n1, int arr2[], int n2) {
5     int i = 0, j = 0;
6     while (i < n1 && j < n2) {
7         if (arr1[i] < arr2[j]) {
8             i++;
9         } else if (arr1[i] > arr2[j]) {
10            j++;
11        } else {
12            printf("%d ", arr1[i]);
13            i++;
14            j++;
15        }
16    }
17    printf("\n");
18 }
19
20 int main() {
21     int T;
22     scanf("%d", &T); // Number of test cases
23
24     while (T--) {
25         int n1, n2;
26
27         // Input first array
28         scanf("%d", &n1);
29         int arr1[n1];
30         for (int i = 0; i < n1; i++) {
31             scanf("%d", &arr1[i]);
32         }
33         // Input second array
34         scanf("%d", &n2);
35         int arr2[n2];
36         for (int i = 0; i < n2; i++) {
37             scanf("%d", &arr2[i]);
38         }
39
40         // Find and print the intersection
41         findIntersection(arr1, n1, arr2, n2);
42     }
43
44     return 0;
45 }
```

	Input	Expected	Got	
✓	1 3 10 17 57 6 2 7 10 15 57 246	10 57	10 57	✓
✓	1 6 1 2 3 4 5 6 2 1 6	1 6	1 6	✓

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

[Finish review](#)

◀ 3-Print Intersection of 2 sorted arrays-
 $O(m*n)$ Time Complexity, $O(1)$ Space Complexity

Jump to...

5-Pair with Difference- $O(n^2)$ Time
Complexity, $O(1)$ Space Complexity ►

CS23331-Design and Analysis of Algorithms-2023 Batch-CSE

Dashboard / My courses / CS23331-DAA-2023-CSE / Dynamic Programming / 2-DP-Playing with chessboard

Quiz navigation

1
✓

Finish review

Started on	Friday, 18 October 2024, 2:56 PM
State	Finished
Completed on	Friday, 18 October 2024, 2:56 PM
Time taken	11 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 $n \times n$ 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 #define MAX 1000
3 int max_monetary_path(int n, int board[MAX][MAX]) {
4     int dp[MAX][MAX];
5     dp[0][0] = board[0][0];
6     for (int j = 1; j < n; j++) {
7         dp[0][j] = dp[0][j - 1] + board[0][j];
8     }
9     for (int i = 1; i < n; i++) {
10        dp[i][0] = dp[i - 1][0] + board[i][0];
11    }
12    for (int i = 1; i < n; i++) {
13        for (int j = 1; j < n; j++) {
14            dp[i][j] = board[i][j] + (dp[i - 1][j] > dp[i][j - 1] ? dp[i - 1][j] : dp[i][j - 1]);
15        }
16    }
17    return dp[n - 1][n - 1];
18 }
19 int main() {
20     int n;
21     scanf("%d", &n);
22     int board[MAX][MAX];
23     for (int i = 0; i < n; i++) {
24         for (int j = 0; j < n; j++) {
25             scanf("%d", &board[i][j]);
26         }
27     }
28     int result = max_monetary_path(n, board);
29     printf("%d\n", result);
30     return 0;
31 }
```

	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.

[◀ 1-DP-Playing with Numbers](#)

Jump to...

[3-DP-Longest Common Subsequence ►](#)