

Problem 1: Finding Complexity

Not secure 118.185.187.137/moodle/mod/quiz/review.php?attempt=229104&cmid=2648

ASHIKA R 2024-CSE A2

RAJALAKSHMI ENGINEERING COLLEGE

Dashboard My courses

CS23331-DAA-2024-CSE / Problem 1: Finding Complexity using Counter Method

Problem 1: Finding Complexity using Counter Method

Started on	Wednesday, 13 August 2025, 6:19 PM
State	Finished
Completed on	Wednesday, 13 August 2025, 6:29 PM
Time taken	10 mins 19 secs
Marks	1.00/1.00
Grade	10.00 out of 10.00 (100%)

Question 1 | Correct Mark 1.00 out of 1.00

8 Search ENG IN 19:03 15-11-2025

Problem 1: Finding Complexity

Not secure 118.185.187.137/moodle/mod/quiz/review.php?attempt=229104&cmid=2648

General

BASIC C PROGRAMMING

BASIC C PROGRAMMING-PR...

Finding Time Complexity o...

Problem 1: Finding Complexi...

Problem 2: Finding Complexi...

Problem 3: Finding Complexi...

Problem 4: Finding Complexi...

Problem 5: Finding Complexi...

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 S...

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

19:04
ENG IN 15-11-2025

Problem 1: Finding Complexity

Not secure 118.185.187.137/moodle/mod/quiz/review.php?attempt=229104&cmid=2648

General

BASIC C PROGRAMMING

BASIC C PROGRAMMING-PR...

Finding Time Complexity o...

Problem 1: Finding Complexi...

Problem 2: Finding Complexi...

Problem 3: Finding Complexi...

Problem 4: Finding Complexi...

Problem 5: Finding Complexi...

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 S...

For example:

Input	Result
9	12

Answer: (penalty regime: 0 %)

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

19:05
15-11-2025

Problem 1: Finding Complexity

Not secure 118.185.187.137/moodle/mod/quiz/review.php?attempt=229104&cmid=2648

X General

▼ BASIC C PROGRAMMING

○ BASIC C PROGRAMMING-PR...

▼ Finding Time Complexity o...

○ Problem 1: Finding Complexi...

○ Problem 2: Finding Complexi...

○ Problem 3: Finding Complexi...

○ Problem 4: Finding Complexi...

○ Problem 5: Finding Complexi...

▼ 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 S...

20 i++;
21 count++;
22 s+=i;
23 count++;
24 }count++;
25 }
26

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

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

Finish review

Back to Course

7 19:05
ENG IN 15-11-2025

Problem 2: Finding Complexity

Not secure 118.185.187.137/moodle/mod/quiz/review.php?attempt=233730&cmid=2649

RAJALAKSHMI ENGINEERING COLLEGE

ASHIKA R 2024-CSE A2

Quiz navigation

Finish review

Dashboard My courses

CS23331-DAA-2024-CSE / Problem 2: Finding Complexity using Counter method

Problem 2: Finding Complexity using Counter method

Started on	Wednesday, 20 August 2025, 7:03 PM
State	Finished
Completed on	Wednesday, 20 August 2025, 7:24 PM
Time taken	21 mins 36 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

19:05 15-11-2025

Problem 2: Finding Complexity

Grade 10.00 out of 10.00 (100%)

Question 1 | Correct Mark 1.00 out of 1.00 [Flag question](#)

[Open course index](#)

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;
            }
        }
    }
}
```

Quiz navigation

1

Finish review

19:06
15-11-2025

Problem 2: Finding Complexity

Not secure 118.185.187.137/moodle/mod/quiz/review.php?attempt=233730&cmid=2649

}

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

Input:

A positive Integer n

Open course index

Print the value of the counter variable

Quiz navigation

1

Finish review

Answer: (penalty regime: 0 %)

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

7 19:06 ENG IN 15-11-2025

Problem 2: Finding Complexity

Not secure 118.185.187.137/moodle/mod/quiz/review.php?attempt=233730&cmid=2649

```
20         count++;
21         count++;
22         break;
23     }
24     count++;
25     count++;
26 }
27 count++;
28
29
30 printf("%d", count);
31
32 return 0;
33 }
34 }
```

Quiz navigation

1

Finish review

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

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

Snipping Tool

Screenshot copied to clipboard
Automatically saved to screenshots folder.

Mark-up and share

19:06
ENG IN
15-11-2025

Problem 3: Finding Complexity

Not secure 118.185.187.137/moodle/mod/quiz/review.php?attempt=233731&cmid=2650

RAJALAKSHMI ENGINEERING COLLEGE

ASHIKA R 2024-CSE A2

Quiz navigation

1

Finish review

Dashboard My courses

CS23331-DAA-2024-CSE / Problem 3: Finding Complexity using Counter Method

Problem 3: Finding Complexity using Counter Method

Started on	Wednesday, 20 August 2025, 7:05 PM
State	Finished
Completed on	Wednesday, 20 August 2025, 7:28 PM
Time taken	22 mins 59 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](#)

7 19:06 ENG IN 15-11-2025

Problem 3: Finding Complexity

Not secure 118.185.187.137/moodle/mod/quiz/review.php?attempt=233731&cmid=2650

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.

```
Open course index
{
    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:

Quiz navigation

1

Finish review

Snipping Tool

Screenshot copied to clipboard
Automatically saved to screenshots folder.

Mark-up and share

19:06
ENG IN 15-11-2025

Problem 3: Finding Complexity

Not secure 118.185.187.137/moodle/mod/quiz/review.php?attempt=233731&cmid=2650

Answer:

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

Quiz navigation

1

Finish review

	Input	Expected	Got	
✓	12	31	31	✓
✓	25	54	54	✓
✓	1	1	1	✓

Windows taskbar: File Explorer, Search, Task View, Edge, File Manager, HP Assistant, Microsoft Store, Microsoft Edge, Google Chrome, Paint, 19:07, ENG IN, 15-11-2025

Problem 4: Finding Complexity

Not secure 118.185.187.137/moodle/mod/quiz/review.php?attempt=233732&cmid=2651

RAJALAKSHMI ENGINEERING COLLEGE

ASHIKA R 2024-CSE A2

Quiz navigation

1

Finish review

Dashboard My courses

CS23331-DAA-2024-CSE / Problem 4: Finding Complexity using Counter Method

Problem 4: Finding Complexity using Counter Method

Started on	Wednesday, 20 August 2025, 7:11 PM
State	Finished
Completed on	Wednesday, 20 August 2025, 7:19 PM
Time taken	7 mins 59 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](#)

7 19:07 ENG IN 15-11-2025

Problem 4: Finding Complexity

Not secure 118.185.187.137/moodle/mod/quiz/review.php?attempt=233732&cmid=2651

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.

Open course index

```
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:

Quiz navigation

1

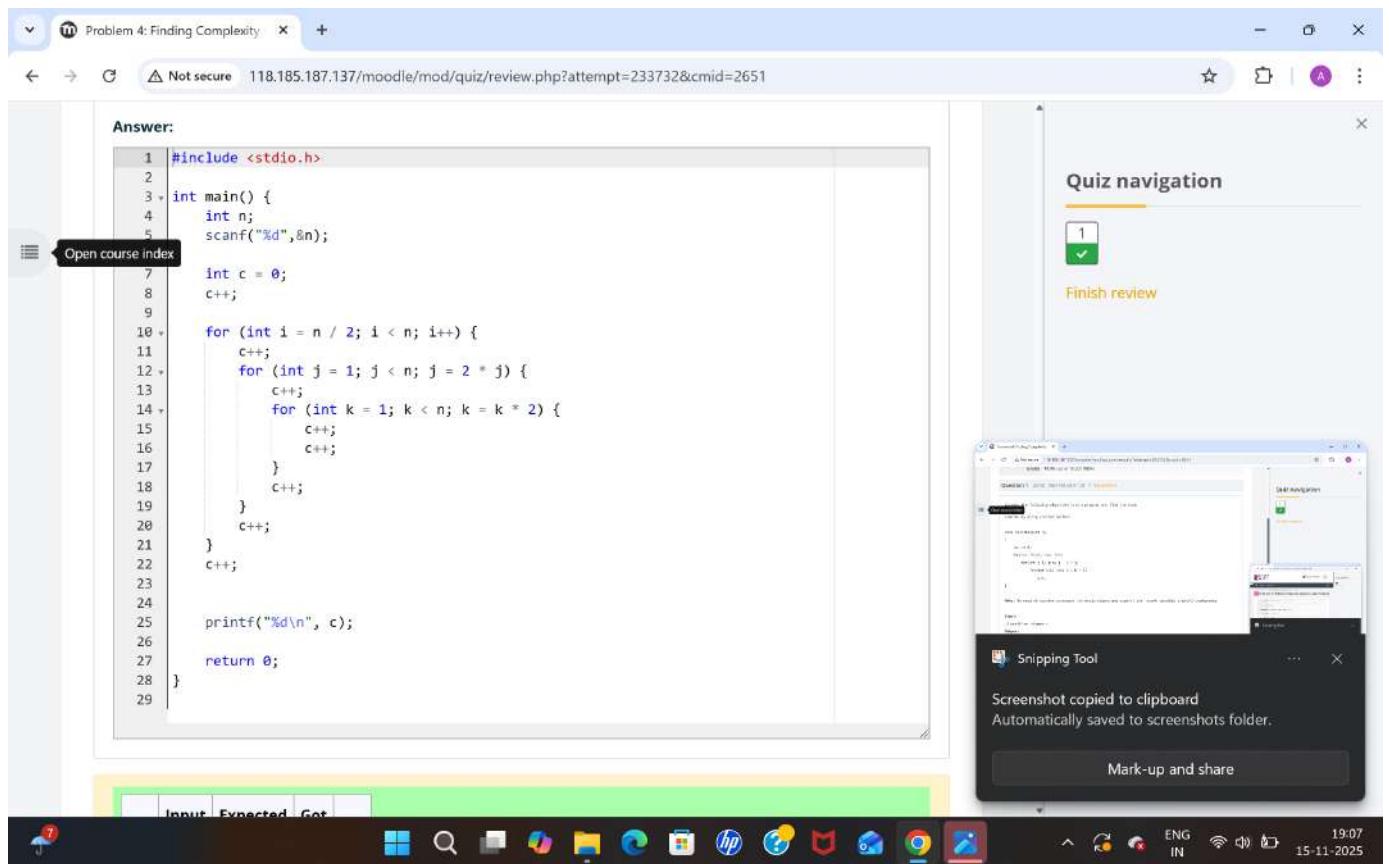
Finish review

Snipping Tool

Screenshot copied to clipboard
Automatically saved to screenshots folder.

Mark-up and share

19:07
ENG IN 15-11-2025



Problem 4: Finding Complexity

Not secure 118.185.187.137/moodle/mod/quiz/review.php?attempt=233732&cmid=2651

```
25     printf( "%d\n", c );
26
27     return 0;
28 }
29 }
```

Open course index

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

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

Finish review

Back to Course

Quiz navigation

1

Finish review

Snipping Tool

Screenshot copied to clipboard
Automatically saved to screenshots folder.

Mark-up and share

19:07
ENG IN 15-11-2025

Problem 5: Finding Complexity

Not secure 118.185.187.137/moodle/mod/quiz/review.php?attempt=233733&cmid=2652

RAJALAKSHMI ENGINEERING COLLEGE

ASHIKA R 2024-CSE A2

Quiz navigation

1

Finish review

Dashboard My courses

CS23331-DAA-2024-CSE / Problem 5: Finding Complexity using counter method

Problem 5: Finding Complexity using counter method

Started on	Wednesday, 20 August 2025, 7:29 PM
State	Finished
Completed on	Wednesday, 20 August 2025, 7:33 PM
Time taken	3 mins 48 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](#)

7 19:08 ENG IN 15-11-2025

Problem 5: Finding Complexity

Not secure 118.185.187.137/moodle/mod/quiz/review.php?attempt=233733&cmid=2652

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:

Quiz navigation

1

Finish review

Snipping Tool

Screenshot copied to clipboard
Automatically saved to screenshots folder.

Mark-up and share

19:08
ENG IN 15-11-2025

The screenshot shows a Moodle quiz review page for Problem 5: Finding Complexity. The URL is 118.185.187.137/moodle/mod/quiz/review.php?attempt=233733&cmid=2652. The page displays the following code:

```
#include <stdio.h>
int main() {
    int n;
    scanf("%d", &n);

    int rev = 0, remainder;
    int count = 0;
    count++;

    while (n != 0) {
        remainder = n % 10;
        count++;
        rev = rev * 10 + remainder;
        count++;
        n /= 10;
        count++;
        count++;
    }

    count++;
    count++;
    printf("%d\n", count);

    return 0;
}
```

The code editor has lines 1 through 27 numbered. Lines 1-6, 8-10, 12-14, 16-17, 19-20, 22-23, and 25-26 are visible. Line 27 is partially visible at the bottom. The line numbers 11 and 11+ are also present. The code is syntax-highlighted with colors for different language elements.

To the right of the code editor is a "Quiz navigation" panel with a progress bar showing step 1 completed. Below it is a "Finish review" button. At the bottom right, there is a "Snipping Tool" window showing a screenshot of the quiz interface, and a message "Screenshot copied to clipboard".

Problem 5: Finding Complexity

Not secure 118.185.187.137/moodle/mod/quiz/review.php?attempt=233733&cmid=2652

```
25 |     return 0;
26 |
27 }
```

	Open course index	put	Expected	Got	
1	✓	12	11	11	✓
2	✓	1234	19	19	✓

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

Quiz navigation

1

Finish review

Back to Course

Finish review

Data retention summary

Snipping Tool

Screenshot copied to clipboard
Automatically saved to screenshots folder.

Mark-up and share

19:08
ENG IN 15-11-2025

1-Number of Zeros in a Given Array

Not secure 118.185.187.137/moodle/mod/quiz/review.php?attempt=245007&cmid=2653

RAJALAKSHMI ENGINEERING COLLEGE

ASHIKA R 2024-CSE A2

Quiz navigation

Finish review

Started on Wednesday, 17 September 2025, 8:25 AM

State Finished

Completed on Wednesday, 17 September 2025, 8:29 AM

Time taken 3 mins 32 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

7 19:09 ENG IN 15-11-2025

1-Number of Zeros in a Given /

Not secure 118.185.187.137/moodle/mod/quiz/review.php?attempt=245007&cmid=2653

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 // Function to find the first occurrence of 0 using Divide and Conquer (Binary Search)
4 int findFirstZero(int arr[], int low, int high) {
5     int result = -1;
6     while (low <= high) {
7         int mid = low + (high - low) / 2;
8
9         if (arr[mid] == 0) {
10             result = mid;
11             high = mid - 1; // Look for 0s on the left side
12         } else {
13             low = mid + 1; // Look for 0s on the right side
14         }
15     }
16 }
```

Quiz navigation

1

Finish review

Snipping Tool

Screenshot copied to clipboard
Automatically saved to screenshots folder.

Mark-up and share

19:09
ENG IN 15-11-2025

1-Number of Zeros in a Given /

Not secure 118.185.187.137/moodle/mod/quiz/review.php?attempt=245007&cmid=2653

```
int mid = low + (high - low) / 2;

if (arr[mid] == 0) {
    result = mid;
    high = mid - 1; // Look for 0s on the left side
} else {
    low = mid + 1; // Look for 0s on the right side
}

return result;

int main() {
    int m;
    // Input size of array
    scanf("%d", &m);

    int arr[m];

    // Input array elements
    for (int i = 0; i < m; i++) {
        scanf("%d", &arr[i]);
    }

    // Use binary search to find the first occurrence of 0
    int firstZeroIndex = findFirstZero(arr, 0, m - 1);

    if (firstZeroIndex == -1) {
        // If no 0 found in the array
        printf("0\n");
    } else {
        // Count the number of 0s, which is the difference between size and the first index
        int countOfZeroes = m - firstZeroIndex;
        printf("%d\n", countOfZeroes);
    }

    return 0;
}
```

Quiz navigation

1

Finish review

Snipping Tool

Screenshot copied to clipboard
Automatically saved to screenshots folder.

Mark-up and share

19:09
ENG IN 15-11-2025

1-Number of Zeros in a Given /

Not secure 118.185.187.137/moodle/mod/quiz/review.php?attempt=245007&cmid=2653

```
42
43     return 0;
44 }
45
46
```

Open course index

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

Quiz navigation

1

Finish review

7 19:09
ENG IN 15-11-2025

1-Number of Zeros in a Given /

Not secure 118.185.187.137/moodle/mod/quiz/review.php?attempt=245007&cmid=2653

Open course index

	0			
	0			
	0			
✓	17	2	2	✓
1				
1				
1				
1				
1				
1				
1				
1				
1				
1				
0				
0				

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

Quiz navigation

1

Finish review

This screenshot shows a Moodle quiz review page. The main area displays a table of results for a single attempt. The table has five columns: the first column contains a green checkmark or a red X, the second column contains numerical values (17, 2, 2, and another green checkmark), and the third, fourth, and fifth columns are empty. Below the table, a message says "Passed all tests!" with a green checkmark icon. A "Correct" button and a message about marks are also present. To the right, a "Quiz navigation" sidebar shows a single item labeled "1" with a dropdown arrow, and a "Finish review" link. The browser's address bar shows the URL and indicates it is not secure. The taskbar at the bottom shows various application icons and the system clock.

2-Majority Element: Attempt re... +

Not secure 118.185.187.137/moodle/mod/quiz/review.php?attempt=245066&cmid=2654

X

Problem 1: Finding Complexi...
Problem 2: Finding Complexi...
Problem 3: Finding Complexi...
Problem 4: Finding Complexi...
Problem 5: Finding Complexi...
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 S...
Greedy Algorithms
1-G-Coin Problem
2-G-Cookies Problem
3-G-Burger Problem

RAJALAKSHMI ENGINEERING COLLEGE

ASHIKA R 2024-CSE A2

Dashboard My courses

CS23331-DAA-2024-CSE / 2-Majority Element

2-Majority Element

Started on	Wednesday, 17 September 2025, 8:29 AM
State	Finished
Completed on	Wednesday, 17 September 2025, 8:30 AM
Time taken	1 min 4 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

7 19:10
ENG IN 15-11-2025

2-Majority Element: Attempt re

Not secure 118.185.187.137/moodle/mod/quiz/review.php?attempt=245066&cmid=2654

Question 1 | Correct: Mark 1.00 out of 1.00 | Flag question

Given an array `nums` of size `n`, return *the majority element*.
The majority element is the element that appears more than $\lceil \frac{n}{2} \rceil$ times. You may assume that the majority element always exists in the array.

Example 1:
`Input: nums = [3,2,3]`
`Output: 3`

Example 2:
`Input: nums = [2,2,1,1,1,2,2]`
`Output: 2`

Constraints:

- `n == nums.length`
- `1 <= n <= 5 * 104`
- `-231 <= nums[i] <= 231 - 1`

For example:

Input	Result
3	3

Snipping Tool
Screenshot copied to clipboard
Automatically saved to screenshots folder.
Mark-up and share

15-11-2025 19:10

2-Majority Element: Attempt review

Not secure 118.185.187.137/moodle/mod/quiz/review.php?attempt=245066&cmid=2654

Input **Result**

3	3
3 2 3	
7	2
2 2 1 1 1 2 2	

Answer: (penalty regime: 0 %)

```
#include <stdio.h>
int majorityElement(int nums[], int n) {
    int candidate = nums[0];
    int count = 1;

    // Boyer-Moore Voting Algorithm
    for (int i = 1; i < n; i++) {
        if (nums[i] == candidate) {
            count++;
        } else {
            count--;
            if (count == 0) {
                candidate = nums[i];
                count = 1;
            }
        }
    }
    return candidate; // The candidate will be the majority
}
int main() {
    int n;
    // Input size of array
}
```

Snipping Tool
Screenshot copied to clipboard
Automatically saved to screenshots folder.
Mark-up and share

19:10 15-11-2025

Problem 1: Finding Complexity...
Problem 2: Finding Complexity...
Problem 3: Finding Complexity...
Problem 4: Finding Complexity...
Problem 5: Finding Complexity...
Divide and Conquer
1-Number of Zeros in a Given Number
2-Majority Element
3-Finding Floor Value
4-Two Elements sum to x
5-Implementation of Quick Sort
Greedy Algorithms
1-G-Coin Problem
2-G-Cookies Problem
3-G-Burger Problem

2-Majority Element: Attempt re

Not secure 118.185.187.137/moodle/mod/quiz/review.php?attempt=245066&cmid=2654

x

- Problem 1: Finding Complexity...
- Problem 2: Finding Complexity...
- Problem 3: Finding Complexity...
- Problem 4: Finding Complexity...
- Problem 5: Finding Complexity...
- 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 S...
- Greedy Algorithms**
- 1-G-Coin Problem
- 2-G-Cookies Problem
- 3-G-Burger Problem

```
17 }  
18 }  
19  
20 return candidate; // The candidate will be the majority element  
21 }  
22  
23 int main() {  
24     int n;  
25     // Input size of array  
26     scanf("%d", &n);  
27  
28     int nums[n];  
29  
30     // Input array elements  
31     for (int i = 0; i < n; i++) {  
32         scanf("%d", &nums[i]);  
33     }  
34  
35     // Find and print the majority element  
36     printf("%d\n", majorityElement(nums, n));  
37  
38     return 0;  
39 }  
40 }
```

	Input	Expected	Got	
<input checked="" type="checkbox"/>	3	3	3	<input checked="" type="checkbox"/>
	3 2 3			

Passed all tests! ✓

Snipping Tool

Screenshot copied to clipboard
Automatically saved to screenshots folder.

Mark-up and share

19:10
ENG IN WiFi 15-11-2025

3-Finding Floor Value: Attempt

Not secure 118.185.187.137/moodle/mod/quiz/review.php?attempt=245081&cmid=2655

RAJALAKSHMI ENGINEERING COLLEGE ASHIKA R 2024-CSE A2

Dashboard My courses

CS23331-DAA-2024-CSE / 3-Finding Floor Value

3-Finding Floor Value

Started on	Wednesday, 17 September 2025, 8:30 AM
State	Finished
Completed on	Wednesday, 17 September 2025, 8:31 AM
Time taken	32 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

7 ENG IN 19:11 15-11-2025

3-Finding Floor Value: Attempt

Not secure 118.185.187.137/moodle/mod/quiz/review.php?attempt=245081&cmid=2655

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.

Open course index Import 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 // Function to find the floor of x using binary search
4 int findFloor(int arr[], int n, int x) {
5     int low = 0, high = n - 1;
6     int floorValue = -1; // Initialize the floor as -1 (indicating no floor found)
7
8     while (low <= high) {
9         int mid = low + (high - low) / 2;
10
11         // If we find an element equal to x, it's the floor
12         if (arr[mid] == x) {
13             return arr[mid];
14         }
15     }
}
```

Quiz navigation

1

Finish review

Snipping Tool

Screenshot copied to clipboard
Automatically saved to screenshots folder.

Mark-up and share

19:11 15-11-2025

3-Finding Floor Value: Attempt

Not secure 118.185.187.137/moodle/mod/quiz/review.php?attempt=245081&cmid=2655

```
10
11     // If we find an element equal to x, it's the floor
12     if (arr[mid] == x) {
13         return arr[mid];
14     }
15
16     // If the current element is less than or equal to x, it's a potential floor
17     if (arr[mid] < x) {
18         floorValue = arr[mid];
19         low = mid + 1; // Move to the right half to search for a larger floor
20     } else {
21         high = mid - 1; // Move to the left half to search for a smaller element
22     }
23
24
25     // Return the largest element that is smaller than or equal to x
26     return floorValue;
27 }
28
29 int main() {
30     int n;
31     // Input size of the array
32     scanf("%d", &n);
33
34     int arr[n];
35
36     // Input elements of the array
37     for (int i = 0; i < n; i++) {
38         scanf("%d", &arr[i]);
39     }
40
41     int x;
42     // Input the value of x
43     scanf("%d", &x);
44
45     // Find and print the floor value for x
46     printf("%d\n", findFloor(arr, n, x));

```

Quiz navigation

1

Finish review

Snipping Tool

Screenshot copied to clipboard
Automatically saved to screenshots folder.

Mark-up and share

ENG IN 19:11 15-11-2025

Screenshot of a Moodle quiz review page showing a code editor and a results table.

Code Editor:

```
41 int x;
42 // Input the value of x
43 scanf("%d", &x);
44
45 // Find and print the floor value for x
46 printf("%d\n", findFloor(arr, n, x));
47
48 return 0;
```

Quiz navigation:

1

[Finish review](#)

Results Table:

	Input	Expected	Got	
✓	6	2	2	✓
	1			
	2			
	8			
	10			
	12			
	19			
	5			
✓	5	85	85	✓
	18			
	22			
	85			
	108			

Snipping Tool Overlay:

Screenshot copied to clipboard
Automatically saved to screenshots folder.
[Mark-up and share](#)

Windows taskbar:



19:11 15-11-2025

3-Finding Floor Value: Attempt

Not secure 118.185.187.137/moodle/mod/quiz/review.php?attempt=245081&cmid=2655

Open course index

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

Quiz navigation

1

Finish review

Snipping Tool

Screenshot copied to clipboard
Automatically saved to screenshots folder.

Mark-up and share

19:11 15-11-2025

	5			
✓	5	85	85	✓
	10			
	22			
	108			
	129			
	100			

	7	9	9	✓
✓	3			
	5			
	7			
	9			
	11			
	13			
	15			
	10			

4-Two Elements sum to x: Attempt 2656

Not secure 118.185.187.137/moodle/mod/quiz/review.php?attempt=245084&cmid=2656

ASHIKA R 2024-CSE A2

Quiz navigation

1

Finish review

RAJALAKSHMI ENGINEERING COLLEGE

Dashboard My courses

CS23331-DAA-2024-CSE / 4-Two Elements sum to x

4-Two Elements sum to x

Started on	Wednesday, 17 September 2025, 8:31 AM
State	Finished
Completed on	Wednesday, 17 September 2025, 8:32 AM
Time taken	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

19:11 15-11-2025

4-Two Elements sum to x: Attempt 245084

Not secure 118.185.187.137/moodle/mod/quiz/review.php?attempt=245084&cmid=2656

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 "Yes", 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 // Recursive function to find two elements whose sum equals x
4 void findPair(int arr[], int low, int high, int x) {
5     // Base case: If low pointer exceeds high pointer, no such pair exists
6     if (low >= high) {
7         printf("No\n");
8         return;
9     }
10
11     // Calculate the sum of elements at low and high pointers
12     int sum = arr[low] + arr[high];
13
14     // If the sum is equal to x, print the pair
```

Quiz navigation

1

Finish review

Snipping Tool

Screenshot copied to clipboard
Automatically saved to screenshots folder.

Mark-up and share

19:11 15-11-2025

4-Two Elements sum to x: Attempt 245084

```
if (low >= high) {
    printf("No\n");
    return;
}

// Calculate the sum of elements at low and high pointers
int sum = arr[low] + arr[high];

// If the sum is equal to x, print the pair
if (sum == x) {
    printf("%d\n", arr[low]);
    printf("%d\n", arr[high]);
    return;
}

// If the sum is less than x, move the low pointer to the right
if (sum < x) {
    findPair(arr, low + 1, high, x);
} else {
    // If the sum is greater than x, move the high pointer to the left
    findPair(arr, low, high - 1, x);
}

int main() {
    int n;
    // Input size of the array
    scanf("%d", &n);

    int arr[n];
    // Input array elements
    for (int i = 0; i < n; i++) {
        scanf("%d", &arr[i]);
    }
    int x;
```

Quiz navigation

1

Finish review

Snipping Tool

Screenshot copied to clipboard
Automatically saved to screenshots folder.

Mark-up and share

19:12 15-11-2025

4-Two Elements sum to x: Attempt 245084

Not secure 118.185.187.137/moodle/mod/quiz/review.php?attempt=245084&cmid=2656

```
36
37 // Input array elements
38 for (int i = 0; i < n; i++) {
39     scanf("%d", &arr[i]);
40 }
41
42 int x;
43 // Input the value of x
44 scanf("%d", &x);
45
46 // Call the recursive function to find the pair
47 findPair(arr, 0, n - 1, x);
48
49 return 0;
50 }
```

Quiz navigation

1

Finish review

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

Snipping Tool

Screenshot copied to clipboard
Automatically saved to screenshots folder.

Mark-up and share

19:12 15-11-2025

4-Two Elements sum to x: Attempt 1

Not secure 118.185.187.137/moodle/mod/quiz/review.php?attempt=245084&cmid=2656

Open course index

	10			
	14			
✓	5	No	No	✓
	2			
	8			
	18			
	100			

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

Quiz navigation

1

Finish review

Back to Course

Data retention summary

Snipping Tool

Screenshot copied to clipboard
Automatically saved to screenshots folder.

Mark-up and share

19:12 15-11-2025

CS23331-DAA-2024-CSE: 5-Implementation of Quick Sort

Not secure 118.185.187.137/moodle/mod/quiz/view.php?id=2657

ASHIKA R 2024-CSE A2

RAJALAKSHMI ENGINEERING COLLEGE

Dashboard My courses

CS23331-DAA-2024-CSE / 5-Implementation of Quick Sort

5-Implementation of Quick Sort

Mark as done

Opened: Thursday, 30 May 2024, 9:49 AM

Re-attempt quiz

Attempts allowed: 2

Grading method: Highest grade

Summary of your previous attempts

8 19:12 15-11-2025

5-Implementation of Quick Sort

Not secure 118.185.187.137/moodle/mod/quiz/review.php?attempt=245090&cmid=2657

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	12 34 67 78 98
67 34 12 98 78	

Answer:

```
1 #include <stdio.h>
2
3 // Function to swap two elements
4 void swap(int *a, int *b) {
5     int temp = *a;
6     *a = *b;
7     *b = temp;
8 }
```

Quiz navigation

1

Finish review

8 19:12 ENG IN 15-11-2025

5-Implementation of Quick Sort

Not secure 118.185.187.137/moodle/mod/quiz/review.php?attempt=245090&cmid=2657

```
4+ void swap(int *a, int *b) {
5     int temp = *a;
6     *a = *b;
7     *b = temp;
8 }
9
10 // Partition function for Quick Sort
11 int partition(int arr[], int low, int high) {
12     int pivot = arr[high]; // Taking last element as pivot
13     int i = low - 1; // Index of smaller element
14
15     for (int j = low; j <= high - 1; j++) {
16         if (arr[j] < pivot) {
17             i++;
18             swap(&arr[i], &arr[j]);
19         }
20     }
21     swap(&arr[i + 1], &arr[high]);
22     return (i + 1);
23 }
24
25 // Quick Sort recursive function
26 void quickSort(int arr[], int low, int high) {
27     if (low < high) {
28         int pi = partition(arr, low, high);
29
30         // Recursively sort elements before and after partition
31         quickSort(arr, low, pi - 1);
32         quickSort(arr, pi + 1, high);
33     }
34 }
35
36 int main() {
37     int n;
38     scanf("%d", &n);
39
40     int arr[n];
}

```

Quiz navigation

1

Finish review

Snipping Tool

Screenshot copied to clipboard
Automatically saved to screenshots folder.

Mark-up and share

19:13 15-11-2025

Screenshot of a Moodle quiz review page showing code execution results.

Code Snippet:

```
37 int n;
38 scanf("%d", &n);
39
40 int arr[n];
41
42 for (int i = 0; i < n; i++) {
43     scanf("%d", &arr[i]);
44 }
45
46 quickSort(arr, 0, n - 1);
47
48 for (int i = 0; i < n; i++) {
49     printf("%d ", arr[i]);
50 }
51 printf("\n");
52
53 return 0;
54 }
55 }
```

Quiz navigation: Quiz navigation panel showing step 1 completed.

Test Results Table:

	Input	Expected	Got	
✓	5 67 34 12 98 78	12 34 67 78 98	12 34 67 78 98	✓
✓	10 1 56 78 98 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	✓

Message: Passed all tests! ✓

Snipping Tool Overlay:

Screenshot copied to clipboard
Automatically saved to screenshots folder.
Mark-up and share

Windows Taskbar:

8 19:13 ENG IN 15-11-2025

1-G-Coin Problem: Attempt review

Not secure 118.185.187.137/moodle/mod/quiz/review.php?attempt=232422&cmid=2658

ASHIKA R 2024-CSE A2

Quiz navigation

1

Finish review

RAJALAKSHMI ENGINEERING COLLEGE

Dashboard My courses

CS23331-DAA-2024-CSE / 1-G-Coin Problem

1-G-Coin Problem

Started on	Wednesday, 20 August 2025, 8:26 AM
State	Finished
Completed on	Wednesday, 20 August 2025, 8:48 AM
Time taken	21 mins 50 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

8 19:13 ENG IN 15-11-2025

1-G-Coin Problem: Attempt review

Not secure 118.185.187.137/moodle/mod/quiz/review.php?attempt=232422&cmid=2658

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 notes, what is the minimum number of coins and/or notes needed to make the change.

Open course index

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
3 int minCoins(int V) {
4     int denominations[] = {1000, 500, 100, 50, 20, 10, 5, 2, 1};
```

Quiz navigation

1

Finish review

Snipping Tool

Screenshot copied to clipboard
Automatically saved to screenshots folder.

Mark-up and share

19:13 15-11-2025

1-G-Coin Problem: Attempt review

Not secure 118.185.187.137/moodle/mod/quiz/review.php?attempt=232422&cmid=2658

```
1 #include <stdio.h>
2
3 + int minCoins(int V) {
4     int denominations[] = {1000, 500, 100, 50, 20, 10, 5, 2, 1};
5     int n = sizeof(denominations) / sizeof(denominations[0]);
6     int count = 0;
7
8     Open course index
9     for (int i = 0; i < n; i++) {
10         if (V == 0)
11             break;
12         count += V / denominations[i]; // number of notes/coins of this denomination
13         V = V % denominations[i];      // reduce remaining amount
14     }
15     return count;
16 }
17 +
18 int main() {
19     int V;
20     scanf("%d", &V);
21     printf("%d\n", minCoins(V));
22     return 0;
23 }
```

Quiz navigation

Finish review

Snipping Tool

Screenshot copied to clipboard
Automatically saved to screenshots folder.

Mark-up and share

	Input	Expected	Got	
✓	49	5	5	✓

Passed all tests! ✓

19:13 15-11-2025

2-G-Cookies Problem: Attempt

Not secure 118.185.187.137/moodle/mod/quiz/review.php?attempt=232622&cmid=2659

RAJALAKSHMI ENGINEERING COLLEGE

ASHIKA R 2024-CSE A2

Quiz navigation

1

Finish review

Dashboard My courses

CS23331-DAA-2024-CSE / 2-G-Cookies Problem

2-G-Cookies Problem

Started on	Wednesday, 20 August 2025, 8:48 AM
State	Finished
Completed on	Wednesday, 20 August 2025, 9:16 AM
Time taken	27 mins 19 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

Snipping Tool

Screenshot copied to clipboard
Automatically saved to screenshots folder.

Mark-up and share

8 19:14 ENG IN 15-11-2025

2-G-Cookies Problem: Attempt

Not secure 118.185.187.137/moodle/mod/quiz/review.php?attempt=232622&cmid=2659

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 least 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 <= n <= 3 * 10^4
1 <= s.length <= 2 * 10^4
```

Quiz navigation

1

Finish review

Snipping Tool

Screenshot copied to clipboard
Automatically saved to screenshots folder.

Mark-up and share

19:14
ENG IN 15-11-2025

2-G-Cookies Problem: Attempt

Not secure 118.185.187.137/moodle/mod/quiz/review.php?attempt=232622&cmid=2659

Open course index

Answer: (penalty regime: 0 %)

```
1 #include <stdio.h>
2 #include <stdlib.h>
3
4 int cmp(const void *a, const void *b) {
5     return *(int*)a - *(int*)b;
6 }
7
8 int main() {
9     int n, m;
10    scanf("%d", &n);
11    int g[n];
12    for (int i = 0; i < n; i++) scanf("%d", &g[i]);
13
14    scanf("%d", &m);
15    int s[m];
16    for (int i = 0; i < m; i++) scanf("%d", &s[i]);
17
18    qsort(g, n, sizeof(int), cmp);
19    qsort(s, m, sizeof(int), cmp);
20
21    int i = 0, j = 0, count = 0;
22    while (i < n && j < m) {
23        if (s[j] >= g[i]) {
24            count++; i++; j++;
25        } else {
26            j++;
27        }
28    }
29 }
```

Quiz navigation

Finish review

Snipping Tool

Screenshot copied to clipboard
Automatically saved to screenshots folder.

Mark-up and share

19:14
ENG IN 15-11-2025

2-G-Cookies Problem: Attempt

Not secure 118.185.187.137/moodle/mod/quiz/review.php?attempt=232622&cmid=2659

```
23 *
24     if (s[i] >= g[i]) {
25         count++; i++; j++;
26     } else {
27         j++;
28     }
29     printf("%d\n", count);
30
31 }
```

Open course index

Quiz navigation

1

Finish review

	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

Snipping Tool

Screenshot copied to clipboard
Automatically saved to screenshots folder.

Mark-up and share

19:14
ENG IN 15-11-2025

3-G-Burger Problem: Attempt

Not secure 118.185.187.137/moodle/mod/quiz/review.php?attempt=284205&cmid=2660

X A2

RAJALAKSHMI ENGINEERING COLLEGE

ASHIKA R 2024-CSE

Dashboard My courses

CS23331-DAA-2024-CSE / 3-G-Burger Problem

3-G-Burger Problem

Started on	Saturday, 15 November 2025, 7:20 PM
State	Finished
Completed on	Saturday, 15 November 2025, 7:48 PM
Time taken	28 mins 18 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

9 Search

ENG IN 19:48 15-11-2025

3-G-Burger Problem: Attempt

Not secure 118.185.187.137/moodle/mod/quiz/review.php?attempt=284205&cmid=2660

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 If he has eaten i burgers with c calories each, then he has to run at least $3^i \cdot c$ kilometers to burn out the calor: burgers with the count of calorie in the order: [1, 3, 2], the kilometers he needs to run are $(3^0 * 1) + (3^1 * 3) + (3^2 * 2)$. But this is not the minimum, so need to try out other orders of consumption and choose the minimum value. Determine he needs to run. Note: He can eat burger in any order and use an efficient sorting algorithm. Apply greedy approach t

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:

Snipping Tool

Screenshot copied to clipboard
Automatically saved to screenshots folder.

Mark-up and share

19:49
ENG IN 15-11-2025

- Problem 3: Finding Complexity
- Problem 4: Finding Complexity
- Problem 5: Finding Complexity
- Divide and Conquer**
 - 1-Number of Zeros in a Given Number
 - 2-Majority Element
 - 3-Finding Floor Value
 - 4-Two Elements sum to x
 - 5-Implementation of Quick Sort
- 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 elements

3-G-Burger Problem: Attempt

Not secure 118.185.187.137/moodle/mod/quiz/review.php?attempt=284205&cmid=2660

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;
6     int dist;
7     scanf("%d",&n);
8     int arr[100];
9     for(int i=0;i<n;i++)
10    scanf("%d",&arr[i]);
11    for(int i=0;i<n;i++)
12    {
13        for(int j=0;j<n;j++)
14        {
15            if(arr[i]>arr[j])
16            {
17                int temp=arr[i];
18                arr[i]=arr[j];
19                arr[j]=temp;
20            }
21        }
22    }
23    for(int i=0;i<n;i++)
24    {
25        dist+=pow(n,i)*arr[i];
26    }
}
```

Snipping Tool

Screenshot copied to clipboard
Automatically saved to screenshots folder.

Mark-up and share

9 Search

19:49
ENG IN 15-11-2025

3-G-Burger Problem: Attempt

Not secure 118.185.187.137/moodle/mod/quiz/review.php?attempt=284205&cmid=2660

X

- Problem 3: Finding Complexity
- Problem 4: Finding Complexity
- Problem 5: Finding Complexity

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 S...

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 elemen...

```
23      for(int i=0;i<n;i++)
24  {
25      dist+=pow(n,i)*arr[i];
26  }
27  printf("%d",dist);
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.

Back to Course

Snipping Tool

Screenshot copied to clipboard
Automatically saved to screenshots folder.

Mark-up and share

9 Search Back to Course 19:49 ENG IN 15-11-2025

4-G-Array Sum max problem: A

Not secure 118.185.187.137/moodle/mod/quiz/review.php?attempt=233724&cmid=2661

ASHIKA R 2024-CSE A2

Quiz navigation

1

Finish review

RAJALAKSHMI
ENGINEERING
COLLEGE

Dashboard My courses

CS23331-DAA-2024-CSE / 4-G-Array Sum max problem

4-G-Array Sum max problem

Started on	Wednesday, 20 August 2025, 6:51 PM
State	Finished
Completed on	Wednesday, 20 August 2025, 6:53 PM
Time taken	2 mins 23 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

9 19:50
Search ENG IN 15-11-2025

4-G-Array Sum max problem: A

Not secure 118.185.187.137/moodle/mod/quiz/review.php?attempt=233724&cmid=2661

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 $\text{arr}[i] * i$, where i is the index of the element ($i = 0, 1, 2, \dots, N$). Write an algorithm based on Greedy technique with a Complexity $O(n\log n)$.

Open course index

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 <stdlib.h>
3
4
5 int compare(const void *a, const void *b) {
6     return *(int *)a - *(int *)b;
7 }
```

Quiz navigation

1

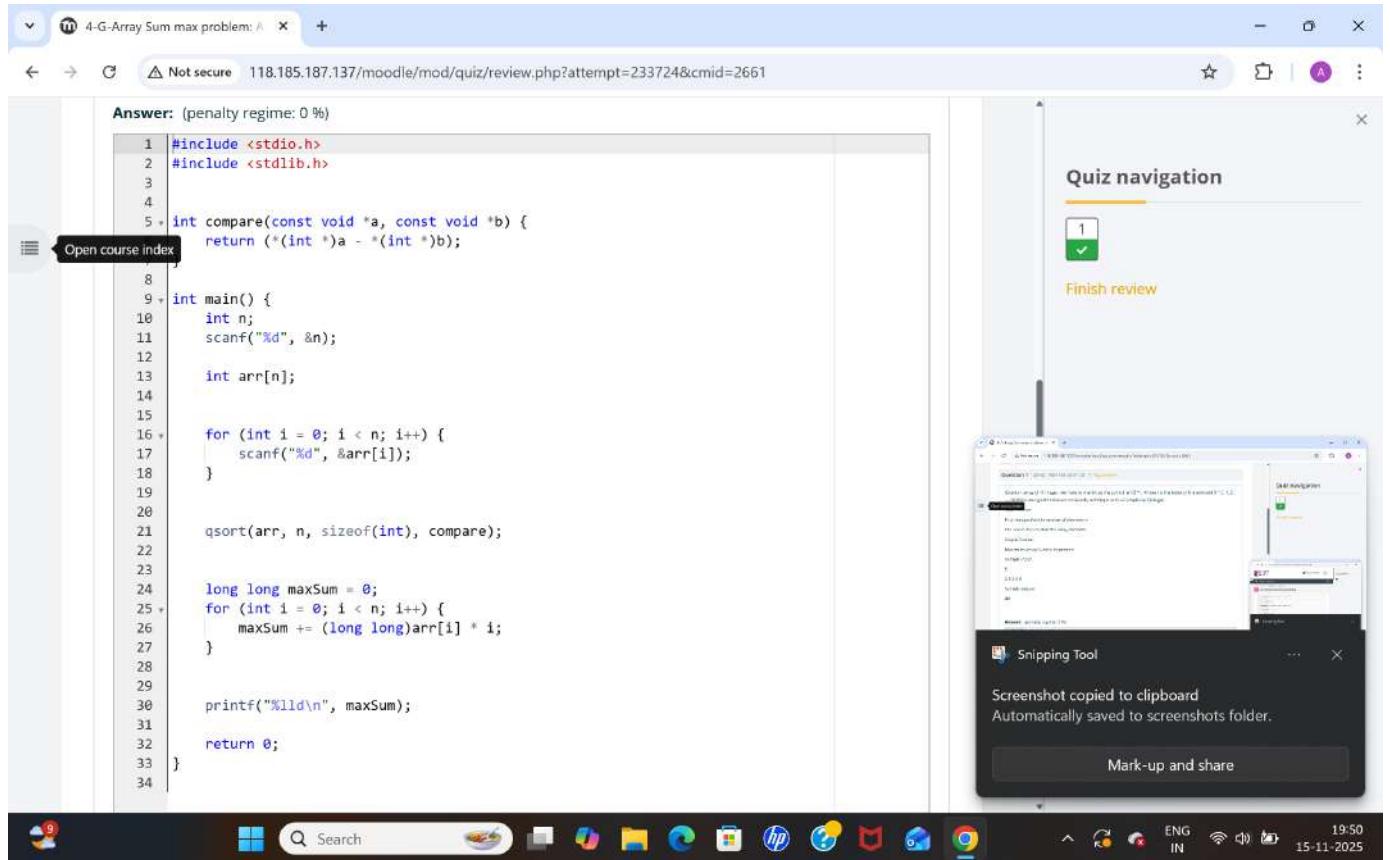
Finish review

Snipping Tool

Screenshot copied to clipboard
Automatically saved to screenshots folder.

Mark-up and share

19:50
ENG IN 15-11-2025



4-G-Array Sum max problem: A

Not secure 118.185.187.137/moodle/mod/quiz/review.php?attempt=233724&cmid=2661

Open course index

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

Passed all tests! ✓

Quiz navigation

1

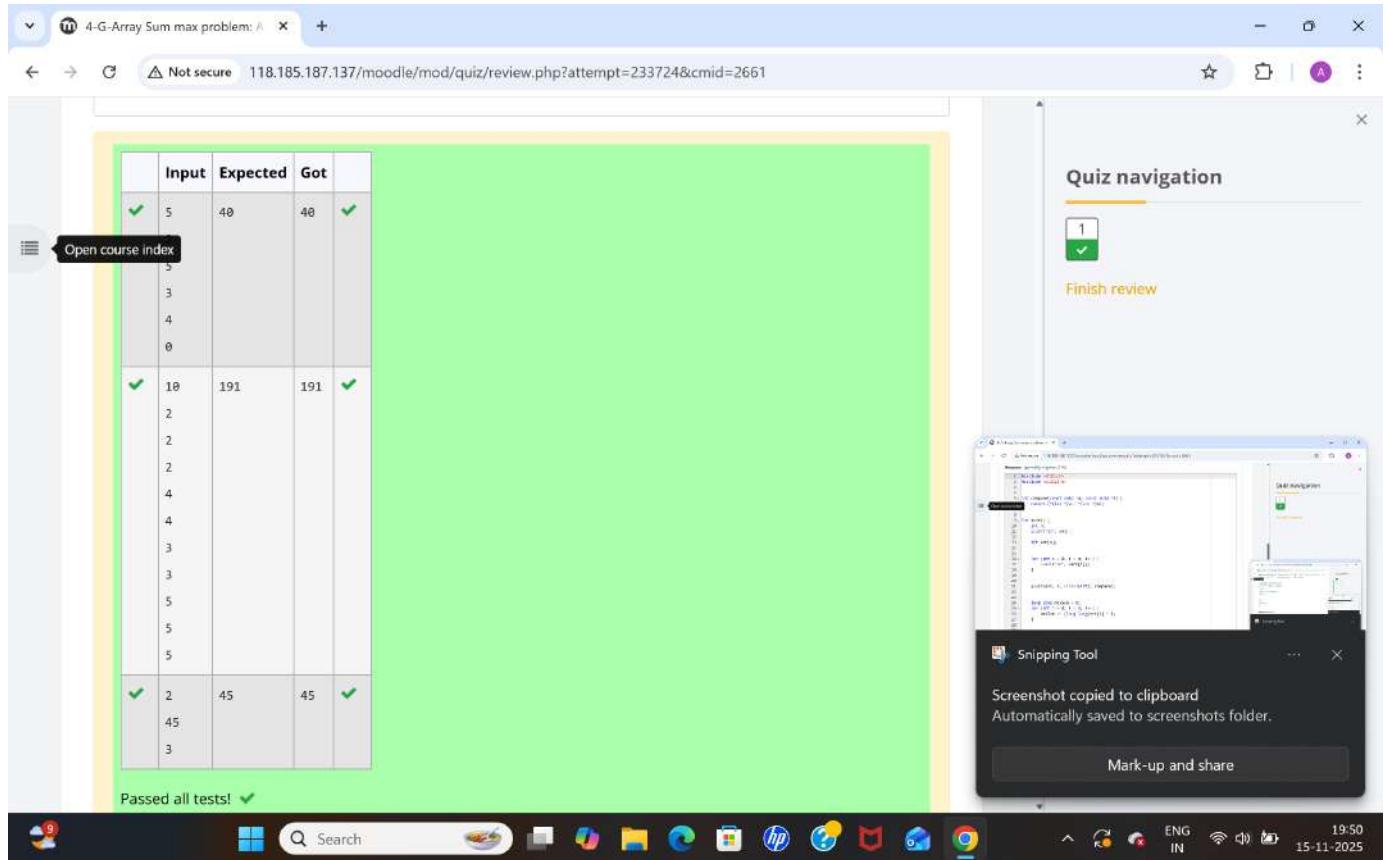
Finish review

Snipping Tool

Screenshot copied to clipboard
Automatically saved to screenshots folder.

Mark-up and share

19:50 15-11-2025



5-G-Product of Array elements

Not secure 118.185.187.137/moodle/mod/quiz/review.php?attempt=233726&cmid=2662

RAJALAKSHMI ENGINEERING COLLEGE

ASHIKA R 2024-CSE A2

Quiz navigation

1

Finish review

Dashboard My courses

CS23331-DAA-2024-CSE / 5-G-Product of Array elements-Minimum

5-G-Product of Array elements-Minimum

Started on	Wednesday, 20 August 2025, 6:53 PM
State	Finished
Completed on	Wednesday, 20 August 2025, 6:56 PM
Time taken	2 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

9 19:50
Search ENG IN 15-11-2025

Screenshot of a Moodle quiz review page titled "5-G-Product of Array elements".

The page shows a question with a mark of 1.00 out of 1.00. The question text is:

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.

A table titled "Input" and "Result" is shown:

Input	Result
3	28
1	
2	
3	
4	
5	
6	

The answer provided is:

```
1 #include <stdio.h>
2 #include <stdlib.h>
3
4
5 int compareAsc(const void *a, const void *b) {
6     return *(int *)a - *(int *)b;
7 }
8
9
10 int compareDesc(const void *a, const void *b) {
11     return *(int *)b - *(int *)a;
12 }
```

The quiz navigation sidebar shows a single item labeled "1". A "Finish review" button is also present.

A screenshot of the Snipping Tool application is overlaid on the bottom right, showing the copied screenshot and sharing options.

Screenshot of a Microsoft Edge browser window showing a C program for calculating the product of array elements. The code uses qsort and custom comparison functions for ascending and descending sorts. A Snipping Tool window is overlaid, indicating a screenshot was taken.

```
5+ int compareAsc(const void *a, const void *b) {  
6    return (*(int *)a - *(int *)b);  
7}  
8  
10+ int compareDesc(const void *a, const void *b) {  
11    return (*(int *)b - *(int *)a);  
12}  
13  
14+ int main() {  
15    int n;  
16    scanf("%d", &n);  
17  
18    int array_One[n], array_Two[n];  
19  
20  
21+     for (int i = 0; i < n; i++) {  
22        scanf("%d", &array_One[i]);  
23    }  
24  
25    // Input array_Two  
26+     for (int i = 0; i < n; i++) {  
27        scanf("%d", &array_Two[i]);  
28    }  
29  
30  
31    qsort(array_One, n, sizeof(int), compareAsc);  
32  
33  
34    qsort(array_Two, n, sizeof(int), compareDesc);  
35  
36  
37    long long minSum = 0;  
38+     for (int i = 0; i < n; i++) {  
39        minSum += (long long)array_One[i] * array_Two[i];  
40    }  
41}
```

Quiz navigation sidebar shows step 1 completed.

Snipping Tool message: Screenshot copied to clipboard. Automatically saved to screenshots folder.

The screenshot shows a Moodle quiz review page for a programming assignment. The page displays the following code:

```
40 }  
41  
42  
43 printf("%lld\n", minSum);  
44  
45 return 0;  
46 }
```

The code is part of a C program that prints the value of `minSum`. The results of the test cases are shown in a table:

	Input	Expected	Got
✓	3	28	28 ✓
	1		
	2		
	3		
	4		
	5		
	6		
✓	4	22	22 ✓
	7		
	5		
	1		
	2		
	1		
	3		

A "Quiz navigation" sidebar indicates the user is on question 1 of the quiz. A "Snipping Tool" window is open at the bottom right, showing the copied screenshot and sharing options.

Screenshot of a Moodle quiz review page titled "5-G-Product of Array elements". The page shows a table of results for a submitted quiz attempt.

The table has columns for Question, Answer, Grade, and Feedback. The first row shows the answer "3" is correct. The second row shows the answer "4" is correct. The third row shows the answer "1" is correct. The fourth row shows the answer "28" is correct. The fifth row shows the answer "5" is correct. The sixth row shows the answer "590" is correct. The seventh row shows the answer "590" is correct. The eighth row shows the answer "✓" is correct.

Below the table, a message says "Passed all tests! ✓".

A "Correct" button is visible, along with the message "Marks for this submission: 1.00/1.00."

On the right, a "Quiz navigation" panel shows a single step labeled "1". A "Finish review" button is also present.

A Snipping Tool window is overlaid on the bottom right, showing the screenshot has been copied to the clipboard and saved to the screenshots folder. It includes a "Mark-up and share" button.

The taskbar at the bottom shows various icons and the date/time: 15-11-2025, 19:51.

1-DP-Playing with Numbers: At: +

Not secure 118.185.187.137/moodle/mod/quiz/review.php?attempt=274671&cmid=2663

ASHIKA R 2024-CSE A2

Quiz navigation

1

Finish review

RAJALAKSHMI
ENGINEERING
COLLEGE

Dashboard My courses

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

1-DP-Playing with Numbers

Started on	Thursday, 30 October 2025, 8:13 PM
State	Finished
Completed on	Thursday, 30 October 2025, 8:18 PM
Time taken	4 mins 50 secs
Grade	10.00 out of 10.00 (100%)

Question 1 | Correct Mark 10.00 out of 10.00 Flag question

Playing with Numbers:

9 Search hp 19:51 ENG IN 15-11-2025

1-DP-Playing with Numbers: At: +

Not secure 118.185.187.137/moodle/mod/quiz/review.php?attempt=274671&cmid=2663

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's turn, 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 the 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

Quiz navigation

1

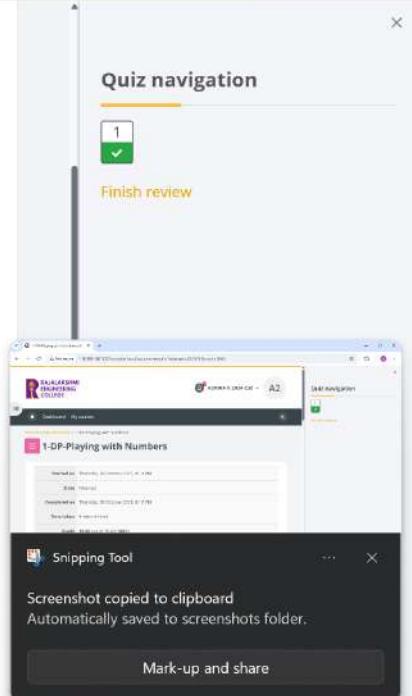
Finish review

Snipping Tool

Screenshot copied to clipboard
Automatically saved to screenshots folder.

Mark-up and share

19:52
ENG IN 15-11-2025



1-DP-Playing with Numbers: At: +

Not secure 118.185.187.137/moodle/mod/quiz/review.php?attempt=274671&cmid=2663

Sample Output
6

Open course index

Answer: (penalty regime: 0 %)

```
1 #include <stdio.h>
2
3 long long countWays(int n) {
4     long long dp[n + 1];
5
6     dp[0] = 1; // One way to make 0 (no numbers)
7
8     for (int i = 1; i <= n; i++) {
9         dp[i] = 0;
10        if (i - 1 >= 0)
11            dp[i] += dp[i - 1];
12        if (i - 3 >= 0)
13            dp[i] += dp[i - 3];
14    }
15
16    return dp[n];
17 }
18
19 int main() {
20     int n;
21     scanf("%d", &n);
22
23     printf("%lld\n", countWays(n));
24
25     return 0;
26 }
27
```

Quiz navigation
1

Finish review

Snipping Tool
Screenshot copied to clipboard
Automatically saved to screenshots folder.
Mark-up and share

19:52
ENG IN 15-11-2025

1-DP-Playing with Numbers: At: +

Not secure 118.185.187.137/moodle/mod/quiz/review.php?attempt=274671&cmid=2663

```
18
19 + int main() {
20     int n;
21     scanf("%d", &n);
22
23     printf("%lld\n", countWays(n));
24
25     return 0;
26
27 }
```

[Open course index](#)

Quiz navigation

1

[Finish review](#)

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

Passed all tests! ✓

Correct

Marks for this submission: 10.00/10.00.

[Back to Course](#)

Snipping Tool

Screenshot copied to clipboard
Automatically saved to screenshots folder.

Mark-up and share

19:52
ENG IN 15-11-2025

2-DP-Playing with chessboard

Not secure 118.185.187.137/moodle/mod/quiz/review.php?attempt=266152&cmid=2664

ASHIKA R 2024-CSE A2

Quiz navigation

1

Finish review

RAJALAKSHMI ENGINEERING COLLEGE

Dashboard My courses

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

2-DP-Playing with chessboard

Started on	Saturday, 18 October 2025, 11:27 AM
State	Finished
Completed on	Saturday, 18 October 2025, 11:28 AM
Time taken	1 min 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:

9 Search hp 19:52 ENG IN 15-11-2025

2-DP-Playing with chessboard

Not secure 118.185.187.137/moodle/mod/quiz/review.php?attempt=266152&cmid=2664

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 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.

[Open course index](#)

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

Quiz navigation

1

Finish review

Snipping Tool

Screenshot copied to clipboard
Automatically saved to screenshots folder.

Mark-up and share

19:52
ENG IN 15-11-2025

2-DP-Playing with chessboard

Not secure 118.185.187.137/moodle/mod/quiz/review.php?attempt=266152&cmid=2664

Output format
Print Maximum monetary value of the path

Open course index

Answer: (penalty regime: 0 %)

```
1 #include <stdio.h>
2 #include <stdlib.h>
3
4 int max(int a, int b) {
5     return (a > b ? a : b);
6 }
7
8 int main() {
9     int n;
10    if (scanf("%d", &n) != 1) {
11        return 0;
12    }
13    // allocate grid
14    int **grid = (int **)malloc(n * sizeof(int *));
15    for (int i = 0; i < n; i++) {
16        grid[i] = (int *)malloc(n * sizeof(int));
17    }
18    for (int i = 0; i < n; i++) {
19        for (int j = 0; j < n; j++) {
20            scanf("%d", &grid[i][j]);
21        }
22    }
23
24    // dp table
25    long long **dp = (long long **)malloc(n * sizeof(long long *));
26    for (int i = 0; i < n; i++) {
```

Quiz navigation
1
Finish review

Snipping Tool
Screenshot copied to clipboard
Automatically saved to screenshots folder.
Mark-up and share

19:52
ENG IN 15-11-2025

2-DP-Playing with chessboard

Not secure 118.185.187.137/moodle/mod/quiz/review.php?attempt=266152&cmid=2664

```
// dp table
long long **dp = (long long **)malloc(n * sizeof(long long *));
for (int i = 0; i < n; i++) {
    dp[i] = (long long *)malloc(n * sizeof(long long));
}
// base
dp[0][0] = grid[0][0];
// first row
for (int j = 1; j < n; j++) {
    dp[0][j] = dp[0][j-1] + grid[0][j];
}
// first column
for (int i = 1; i < n; i++) {
    dp[i][0] = dp[i-1][0] + grid[i][0];
}
// fill rest
for (int i = 1; i < n; i++) {
    for (int j = 1; j < n; j++) {
        dp[i][j] = max(dp[i-1][j], dp[i][j-1]) + grid[i][j];
    }
}
long long result = dp[n-1][n-1];
printf("%lld\n", result);
// free memory
for (int i = 0; i < n; i++) {
    free(grid[i]);
    free(dp[i]);
}
free(grid);
free(dp);
return 0;
```

Quiz navigation

1

Finish review

Snipping Tool

Screenshot copied to clipboard
Automatically saved to screenshots folder.

Mark-up and share

19:52
ENG IN 15-11-2025

2-DP-Playing with chessboard

Not secure 118.185.187.137/moodle/mod/quiz/review.php?attempt=266152&cmid=2664

```
55
56     return 0;
57 }
58 }
```

	put	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.

Quiz navigation

1

Finish review

Snipping Tool

Screenshot copied to clipboard
Automatically saved to screenshots folder.

Mark-up and share

The screenshot shows a Microsoft Windows desktop environment. A browser window is open to a Moodle quiz review page. The page displays a code editor with a snippet of C-like code and a table of test cases with 'Expected' and 'Got' results. Below the table, a green bar indicates 'Passed all tests!' with a checkmark. A message at the bottom says 'Marks for this submission: 10.00/10.00.' To the right of the browser, a 'Quiz navigation' sidebar shows a single item labeled '1'. A 'Snipping Tool' window is overlaid on the desktop, containing a screenshot of the browser and the 'Mark-up and share' button. The taskbar at the bottom shows various pinned icons and the date/time '15-11-2025 19:53'.

3-DP-Longest Common Subsequence

Not secure 118.185.187.137/moodle/mod/quiz/review.php?attempt=266154&cmid=2665

RAJALAKSHMI ENGINEERING COLLEGE

ASHIKA R 2024-CSE A2

Quiz navigation

1

Finish review

Dashboard My courses

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

3-DP-Longest Common Subsequence

Started on	Saturday, 18 October 2025, 11:29 AM
State	Finished
Completed on	Saturday, 18 October 2025, 11:29 AM
Time taken	40 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

9 19:53 ENG IN 15-11-2025

3-DP-Longest Common Subsequence

Not secure 118.185.187.137/moodle/mod/quiz/review.php?attempt=266154&cmid=2665

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.

Open course index

s1: ggtabe
s2: tgatasb

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 %)

Quiz navigation

1

Finish review

Rajalakshmi Engineering College

3-DP-Longest Common Subsequence

Snipping Tool

Screenshot copied to clipboard
Automatically saved to screenshots folder.

Mark-up and share

9 19:53 ENG IN 15-11-2025

Answer: (penalty regime: 0 %)

```
1 #include <stdio.h>
2 #include <stdlib.h>
3 #include <string.h>
4
5 int max(int a, int b) {
6     return (a > b ? a : b);
7 }
8
9 int main() {
10     char s1[1001], s2[1001];
11     // Read two strings (space separated or newline separated)
12     if (scanf("%100s %100s", s1, s2) != 2) {
13         return 0;
14     }
15     int m = strlen(s1);
16     int n = strlen(s2);
17     // Create DP table of size (m+1) x (n+1)
18     int **dp = (int **)malloc((m+1) * sizeof(int *));
19     for (int i = 0; i <= m; i++) {
20         dp[i] = (int *)malloc((n+1) * sizeof(int));
21     }
22     // Initialize first row/column to 0
23     for (int i = 0; i <= m; i++) {
24         dp[i][0] = 0;
25     }
26     for (int j = 0; j <= n; j++) {
27         dp[0][j] = 0;
28     }
29     // Fill dp table
30     for (int i = 1; i <= m; i++) {
31         for (int j = 1; j <= n; j++) {
32             if (s1[i-1] == s2[j-1]) {
33                 dp[i][j] = 1 + dp[i-1][j-1];
34             } else {
```

Quiz navigation

Finish review

Snipping Tool

Screenshot copied to clipboard
Automatically saved to screenshots folder.

Mark-up and share

The screenshot shows a Windows desktop environment. In the center, a Microsoft Edge browser window is open, displaying a Moodle quiz page titled "3-DP-Longest Common Subsequence". The page contains a code editor with C code for finding the longest common subsequence between two strings. To the right of the browser, a "Quiz navigation" sidebar is visible, showing a green checkmark next to the first item. A "Snipping Tool" window is overlaid on the desktop, containing a screenshot of the browser and the message "Screenshot copied to clipboard" and "Automatically saved to screenshots folder". At the bottom of the screen, the Windows taskbar is visible with various pinned icons and system status indicators.

3-DP-Longest Common Subsequence

Not secure 118.185.187.137/moodle/mod/quiz/review.php?attempt=266154&cmid=2665

```
32+
33+    for (int j = s2.length - 1, j >= 0, j--) {
34+        if (s1[i-1] == s2[j-1]) {
35+            dp[i][j] = 1 + dp[i-1][j-1];
36+        } else {
37+            dp[i][j] = max(dp[i-1][j], dp[i][j-1]);
38+        }
39+    }
40+    // Answer is dp[m][n]
41+    printf("%d\n", dp[m][n]);
42+    // Free memory
43+    for (int i = 0; i <= m; i++) {
44+        free(dp[i]);
45+    }
46+    free(dp);
47+    return 0;
48+}
```

Quiz navigation

1

Finish review

Passed all tests! ✓

Correct

Snipping Tool

Screenshot copied to clipboard
Automatically saved to screenshots folder.

Mark-up and share

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

19:53 15-11-2025

4-DP-Longest non-decreasing Subsequence

Not secure 118.185.187.137/moodle/mod/quiz/review.php?attempt=266155&cmid=2666

ASHIKA R 2024-CSE A2

Quiz navigation

1

Finish review

RAJALAKSHMI ENGINEERING COLLEGE

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, 11:29 AM
State	Finished
Completed on	Saturday, 18 October 2025, 11:30 AM
Time taken	55 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

9 19:53 ENG IN 15-11-2025

4-DP-Longest non-decreasing Subsequence

Not secure 118.185.187.137/moodle/mod/quiz/review.php?attempt=266155&cmid=2666

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.

Open course index

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 %)

```
1 #include <stdio.h>
2 #include <stdlib.h>
3
4 int max(int a, int b) {
5     return (a > b ? a : b);
6 }
7
8 int main() {
9     int n;
10    if (scanf("%d", &n) != 1) {
11        return 0;
12    }
13    int *arr = (int *)malloc(n * sizeof(int));
14    for (int i = 0; i < n; i++) {
15        scanf("%d", &arr[i]);
16    }
17    if (n <= 0) {
```

Quiz navigation

1

Finish review

Snipping Tool

Screenshot copied to clipboard
Automatically saved to screenshots folder.

Mark-up and share

19:54
ENG IN 15-11-2025

4-DP-Longest non-decreasing

Not secure 118.185.187.137/moodle/mod/quiz/review.php?attempt=266155&cmid=2666

```
14+ for (int i = 0; i < n; i++) {
15    scanf("%d", &arr[i]);
16}
17+ if (n <= 0) {
18    printf("0\n");
19    free(arr);
20    return 0;
}
22
23 int *dp = (int *)malloc(n * sizeof(int));
24 for (int i = 0; i < n; i++) {
25     dp[i] = 1;
26 }
27
28 int result = 1;
29 for (int i = 1; i < n; i++) {
30     for (int j = 0; j < i; j++) {
31         if (arr[j] <= arr[i]) {
32             dp[i] = max(dp[i], dp[j] + 1);
33         }
34     }
35     if (dp[i] > result) {
36         result = dp[i];
37     }
38 }
39
40 printf("%d\n", result);
41
42 free(arr);
43 free(dp);
44
45 }
```

Quiz navigation

1

Finish review

Snipping Tool

Screenshot copied to clipboard
Automatically saved to screenshots folder.

Mark-up and share

19:54 15-11-2025

4-DP-Longest non-decreasing

Not secure 118.185.187.137/moodle/mod/quiz/review.php?attempt=266155&cmid=2666

```
44     return 0;
45 }
46
```

Open course index

	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

Quiz navigation

1

Finish review

Snipping Tool

Screenshot copied to clipboard
Automatically saved to screenshots folder.

Mark-up and share

19:54
ENG IN 15-11-2025

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

Not secure 118.185.187.137/moodle/mod/quiz/review.php?attempt=266156&cmid=2667

RAJALAKSHMI ENGINEERING COLLEGE

ASHIKA R 2024-CSE A2

Quiz navigation

1

Finish review

Started on Saturday, 18 October 2025, 11:31 AM

State Finished

Completed on Saturday, 18 October 2025, 11:32 AM

Time taken 1 min 50 secs

Marks 1.00/1.00

Grade 4.00 out of 4.00 (100%)

9 Search hp 19:54 ENG IN 15-11-2025