×

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

Problem 1: Finding Complexity using Counter Method

Started on	Sunday, 17 August 2025, 10:58 PM
State	Finished
Completed on	Sunday, 17 August 2025, 11:26 PM
Time taken	27 mins 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

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 %)

```
#include<stdio.h>
int main(){
    int n;
    scanf("%d",&n);
    int i=1;
    int s=1;
    int counter=0;
    counter+=2;
    while(s<=n){
        counter++;
        i++;
        counter++;
        s+=i;
        counter++;
    }
    counter++;
    printf("%d",counter);
    return 0;
}
```

	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

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

Problem 1: Finding Complexity using Counter Method

Started on	Sunday, 17 August 2025, 10:58 PM
State	Finished
Completed on	Sunday, 17 August 2025, 11:26 PM
Time taken	27 mins 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

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 %)

```
#include<stdio.h>
int main(){
    int n;
    scanf("%d",&n);
    int i=1;
    int s=1;
    int counter=0;
    counter+=2;
    while(s<=n){
        counter++;
        i++;
        counter++;
        s+=i;
        counter++;
    }
    counter++;
    printf("%d",counter);
    return 0;
}
```

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

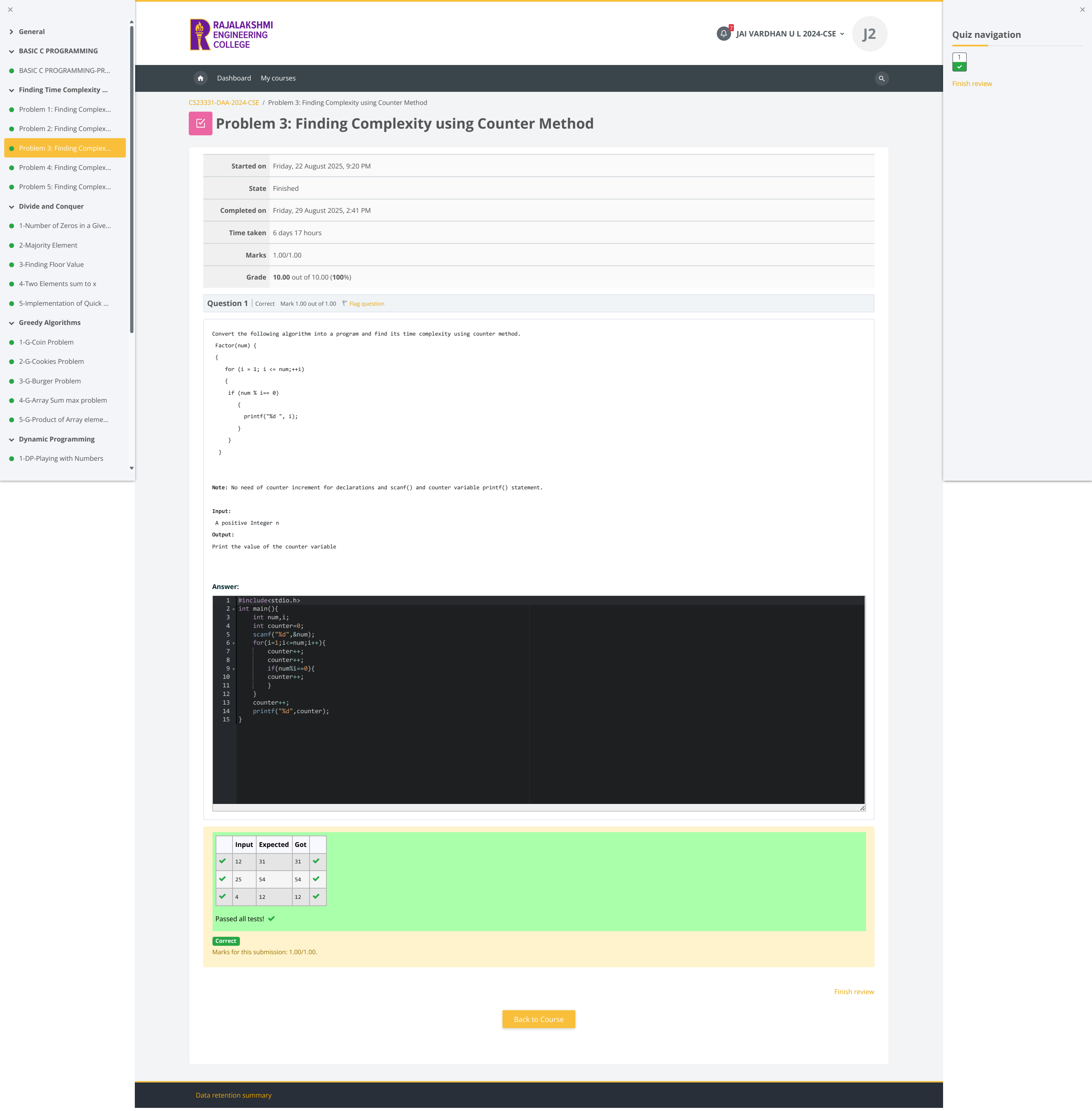
Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

Finish review

Back to Course



- General
- BASIC C PROGRAMMING
  - BASIC C PROGRAMMING-PR...
- Finding Time Complexity ...
  - Problem 1: Finding Complex...
  - Problem 2: Finding Complex...
  - Problem 3: Finding Complex...
  - Problem 4: Finding Complex...
  - Problem 5: Finding Complex...
- Divide and Conquer
  - 1-Number of Zeros in a Give...
  - 2-Majority Element
  - 3-Finding Floor Value
  - 4-Two Elements sum to x
  - 5-Implementation of Quick ...
- Greedy Algorithms
  - 1-G-Coin Problem
  - 2-G-Cookies Problem
  - 3-G-Burger Problem
  - 4-G-Array Sum max problem
  - 5-G-Product of Array eleme...
- Dynamic Programming
  - 1-DP-Playing with Numbers

RAJALAKSHMI ENGINEERING COLLEGE

JAI VARDHAN U L 2024-CSE J2

Dashboard My courses

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

Problem 3: Finding Complexity using Counter Method

Started on	Friday, 22 August 2025, 9:20 PM
State	Finished
Completed on	Friday, 29 August 2025, 2:41 PM
Time taken	6 days 17 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.

```
 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:

```
#include<stdio.h>
int main(){
    int num,i;
    int counter=0;
    scanf("%d",&num);
    for(i=1;i<=num;i++){
        counter++;
        counter++;
        if(num%i==0){
        counter++;
        }
    }
    counter++;
    printf("%d",counter);
}
```

	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

Back to Course

Quiz navigation

1 ✓

Finish review

Data retention summary

×

- General
- BASIC C PROGRAMMING
- BASIC C PROGRAMMING-PR...
- Finding Time Complexity ...
- Problem 1: Finding Complex...
- Problem 2: Finding Complex...
- Problem 3: Finding Complex...
- Problem 4: Finding Complex...
- Problem 5: Finding Complex...
- Divide and Conquer
- 1-Number of Zeros in a Give...
- 2-Majority Element
- 3-Finding Floor Value
- 4-Two Elements sum to x
- 5-Implementation of Quick ...
- Greedy Algorithms
- 1-G-Coin Problem
- 2-G-Cookies Problem
- 3-G-Burger Problem
- 4-G-Array Sum max problem
- 5-G-Product of Array eleme...
- Dynamic Programming
- 1-DP-Playing with Numbers

RAJALAKSHMI ENGINEERING COLLEGE

7 JAI VARDHAN U L 2024-CSE J2

Dashboard My courses

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

Problem 4: Finding Complexity using Counter Method

Started on	Friday, 29 August 2025, 2:08 PM
State	Finished
Completed on	Friday, 29 August 2025, 2:41 PM
Time taken	32 mins 51 secs
Marks	1.00/1.00
Grade	**10.00** out of 10.00 (**100**%)

Question 1 Correct Mark 1.00 out of 1.00 Flag question

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:

```
#include<stdio.h>
int function(int n){
    int c=0;
    int count=0;
    count++;
    for(int i=n/2;count++,i<n;i++){
        for(int j=1;count++,j<n;j=2*j){
            for(int k=1;count++,k<n;k=k*2){
                count++;
                c++;
            }
        }
    }
    return count;
}
int main(){
    int a;
    scanf("%d",&a);
    printf("%d",function(a));
}
```

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

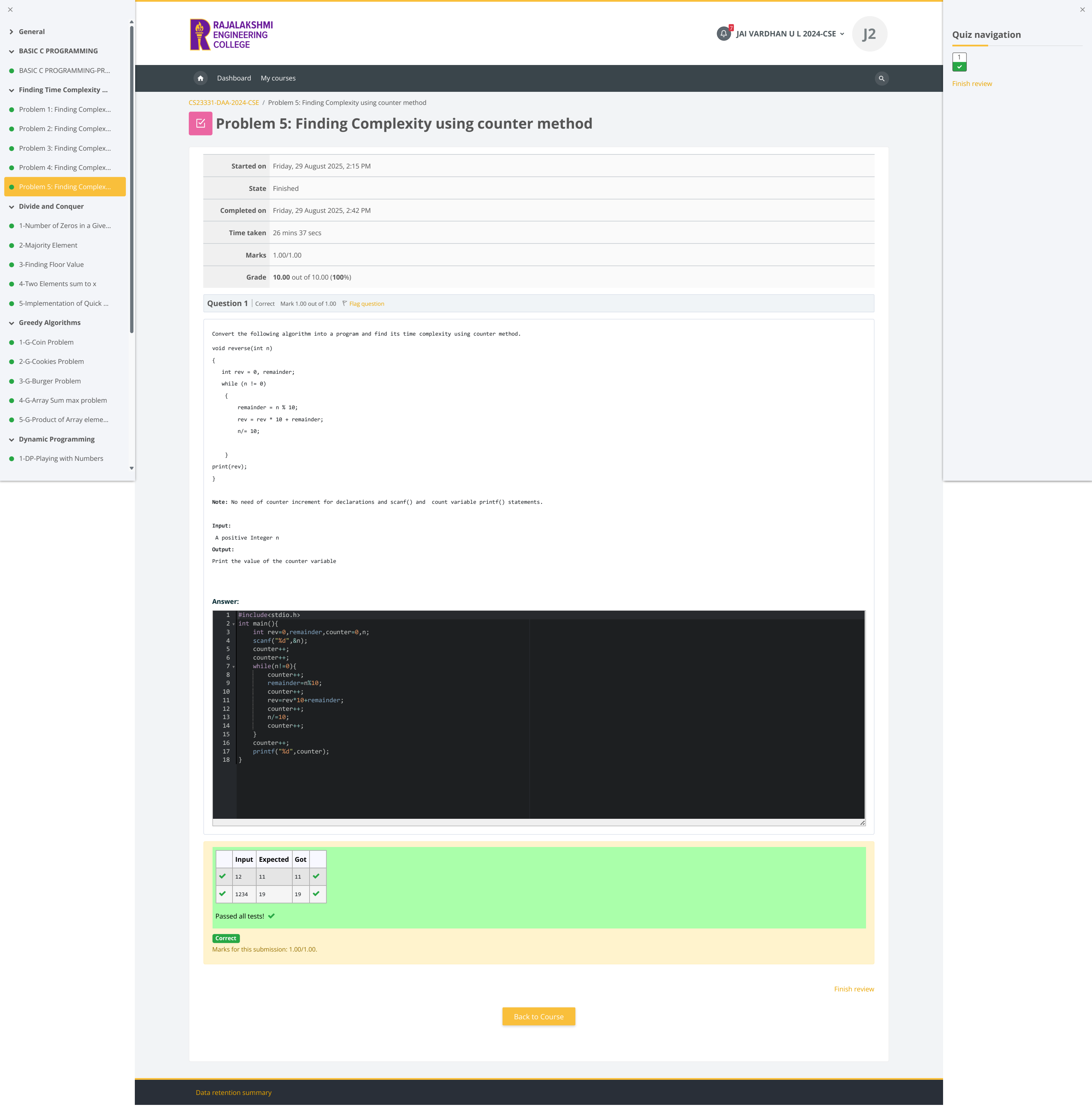
Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

Finish review

Back to Course


General
BASIC C PROGRAMMING
BASIC C PROGRAMMING-PR...
Finding Time Complexity ...
Problem 1: Finding Complex...
Problem 2: Finding Complex...
Problem 3: Finding Complex...
Problem 4: Finding Complex...
Problem 5: Finding Complex...
Divide and Conquer
1-Number of Zeros in a Give...
2-Majority Element
3-Finding Floor Value
4-Two Elements sum to x
5-Implementation of Quick ...
Greedy Algorithms
1-G-Coin Problem
2-G-Cookies Problem
3-G-Burger Problem
4-G-Array Sum max problem
5-G-Product of Array eleme...
Dynamic Programming
1-DP-Playing with Numbers
RAJALAKSHMI ENGINEERING COLLEGE
JAI VARDHAN U L 2024-CSE
J2
Dashboard
My courses
Quiz navigation
1
Finish review

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

Problem 5: Finding Complexity using counter method

Started on	Friday, 29 August 2025, 2:15 PM
State	Finished
Completed on	Friday, 29 August 2025, 2:42 PM
Time taken	26 mins 37 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:

```
#include<stdio.h>
int main(){
    int rev=0,remainder,counter=0,n;
    scanf("%d",&n);
    counter++;
    counter++;
    while(n!=0){
        counter++;
        remainder=n%10;
        counter++;
        rev=rev*10+remainder;
        counter++;
        n/=10;
        counter++;
    }
    counter++;
    printf("%d",counter);
}
```

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

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

Finish review

Back to Course

Data retention summary