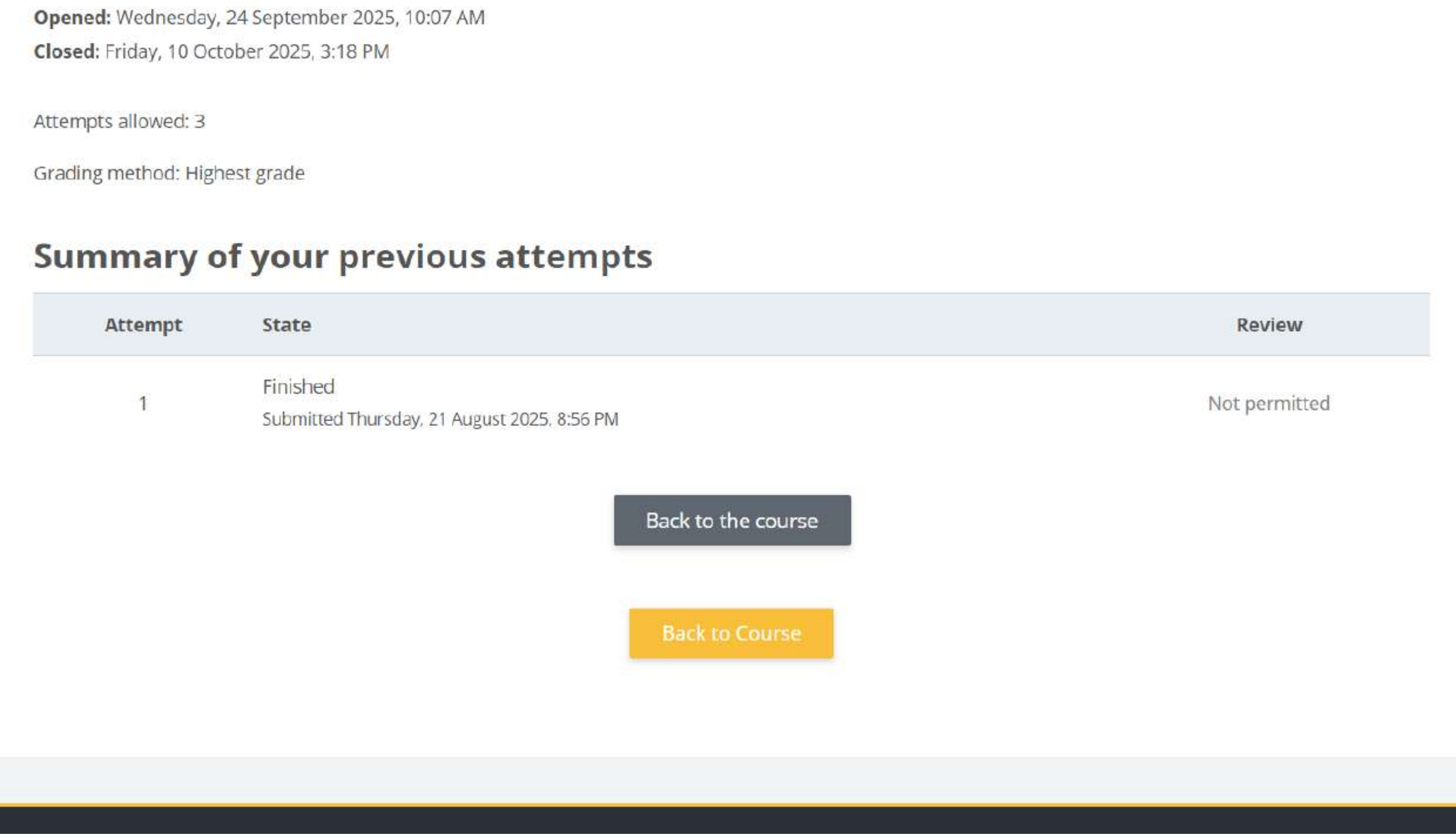


Opened: Wednesday, 24 September 2025, 10:07 AM

Closed: Friday, 10 October 2025, 3:18 PM

Attempts allowed: 3

Grading method: Highest grade

Summary of your previous attempts

Attempt	State	Review
1	Finished Submitted Thursday, 21 August 2025, 8:56 PM	Not permitted

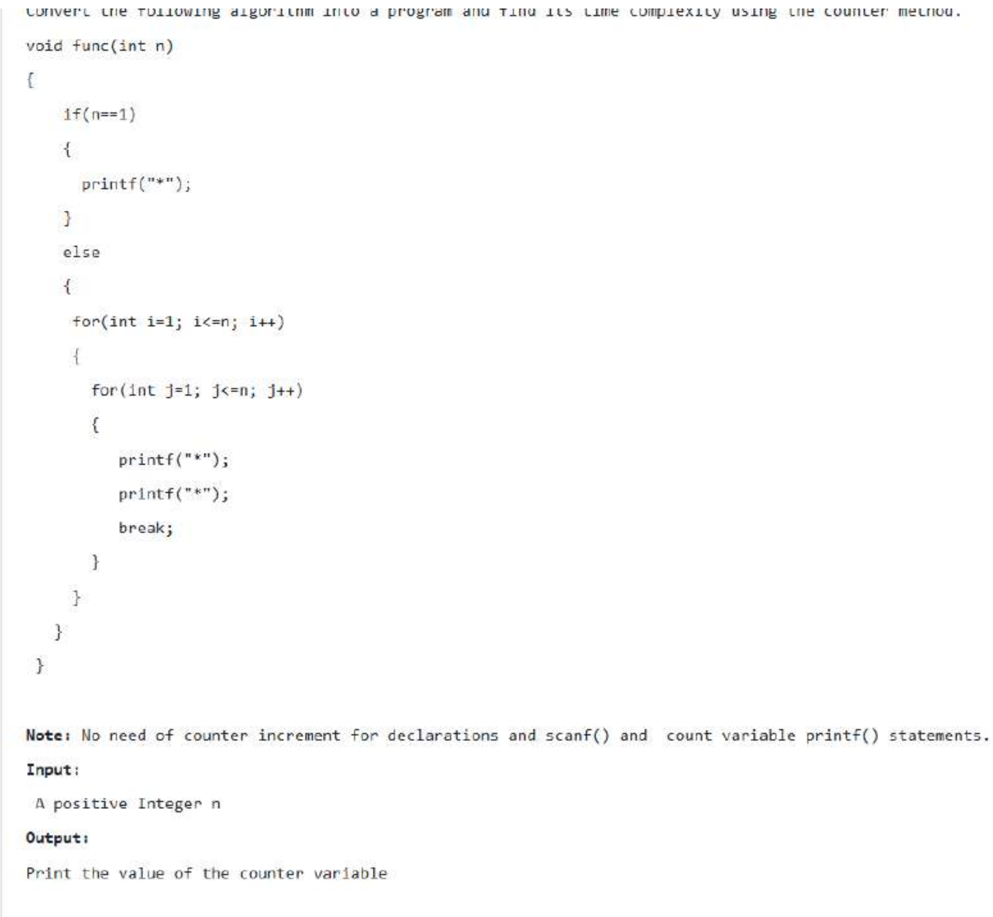
Problem 2: Finding Complexity using Counter method

Started on	Thursday, 21 August 2025, 8:58 PM
State	Finished
Completed on	Thursday, 21 August 2025, 9:09 PM
Time taken	11 mins 22 secs
Marks	1.00/1.00
Grade	**10.00** out of 10.00 (**100**%)

Question 1 | Correct Mark 1.00 out of 1.00

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

```
void func(int n)
{
    if(n--1)
    {
```



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;
       }
     }
   }
 }
```

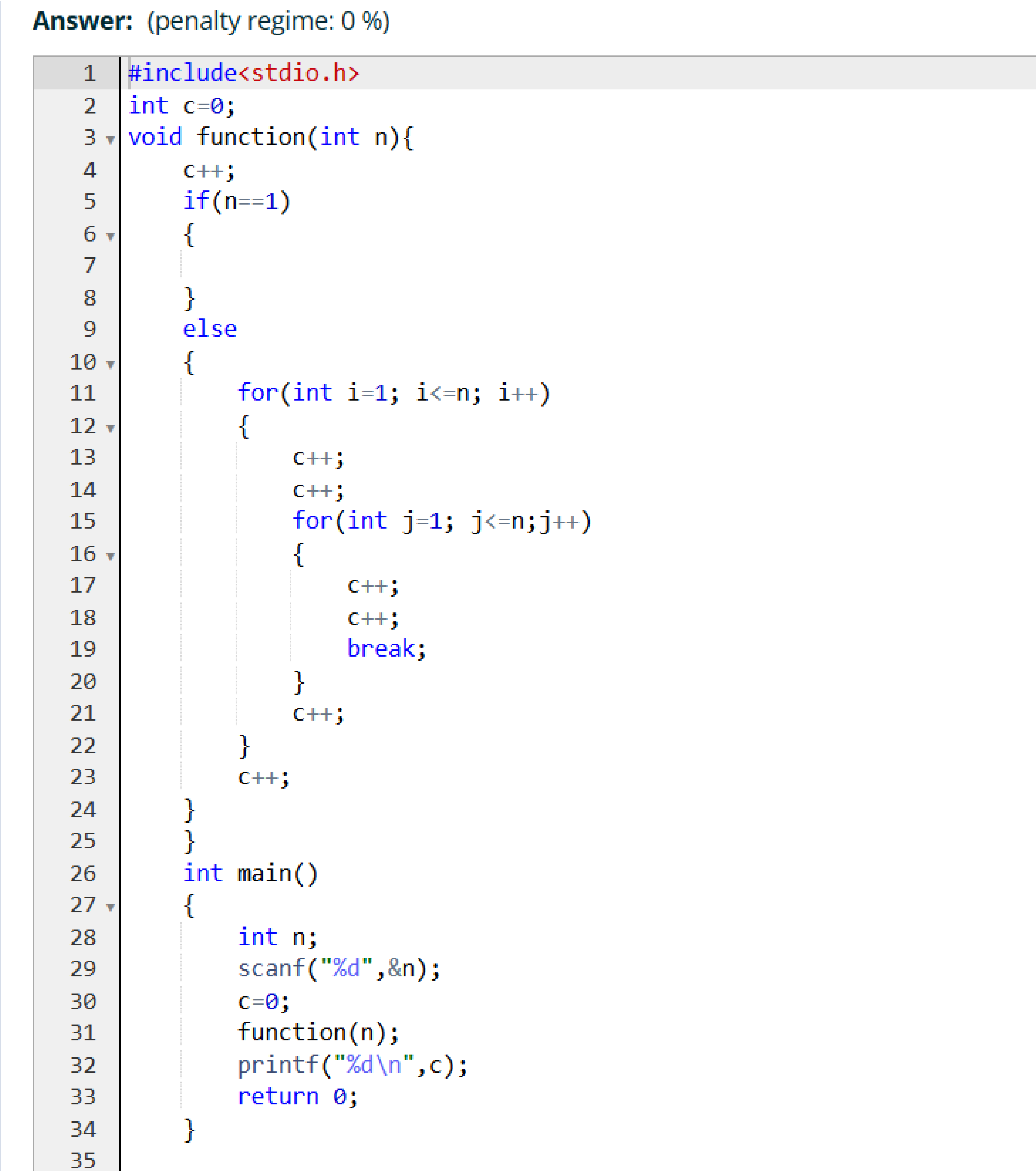
Note: No need of counter increment for declarations and scanf() and count variable printf() statements.

Input:

A positive Integer n

Output:

Print the value of the counter variable

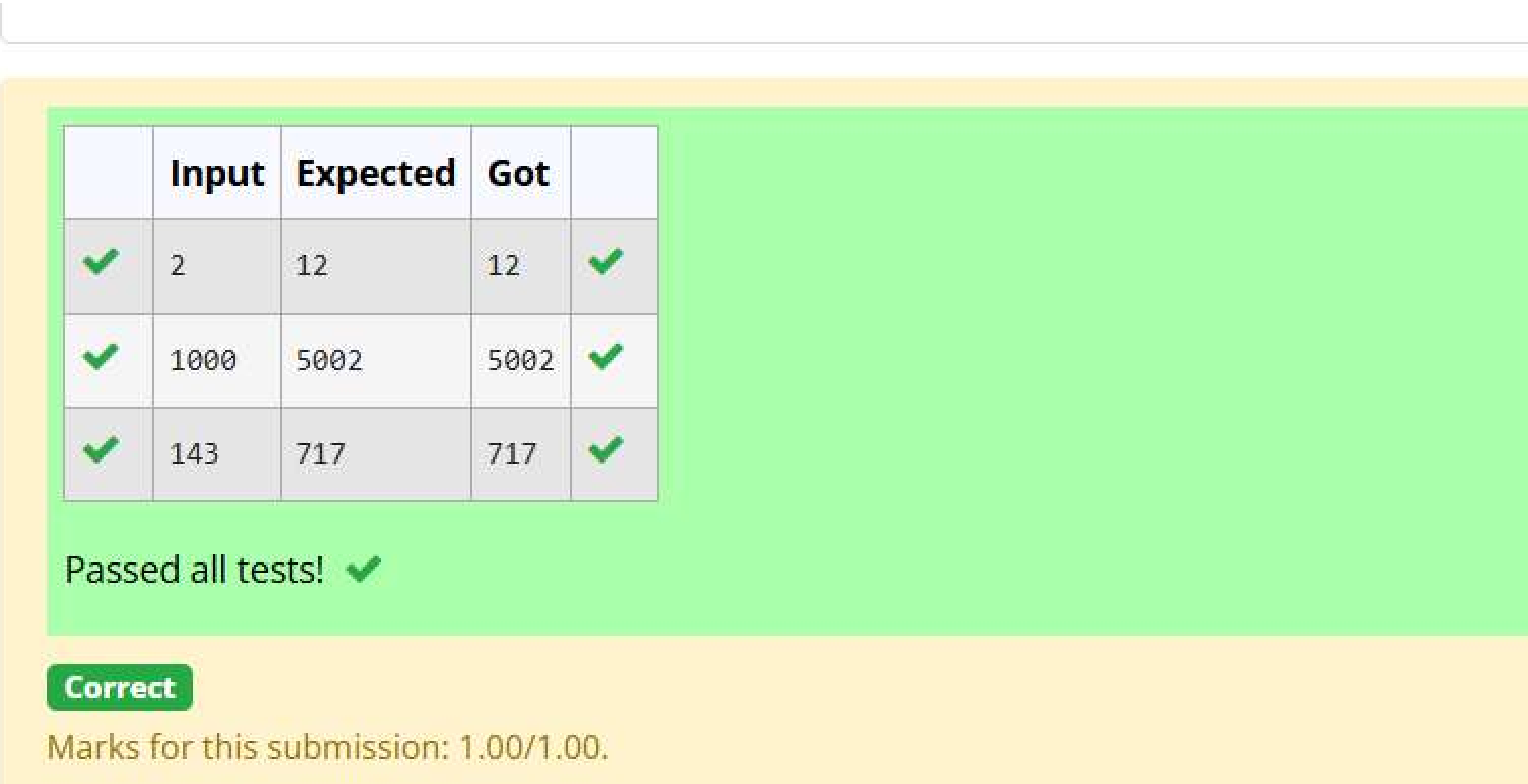


Answer: (penalty regime: 0 %)

```c
#include<stdio.h>
int c=0;
void function(int n){
    c++;
    if(n==1)
    {

    }
    else
    {
        for(int i=1; i<=n; i++)
        {
            c++;
            c++;
            for(int j=1; j<=n;j++)
            {
                c++;
                c++;
                break;
            }
            c++;
        }
        c++;
    }
    }
    int main()
    {
        int n;
        scanf("%d",&n);
        c=0;
        function(n);
        printf("%d\n",c);
        return 0;
    }

```

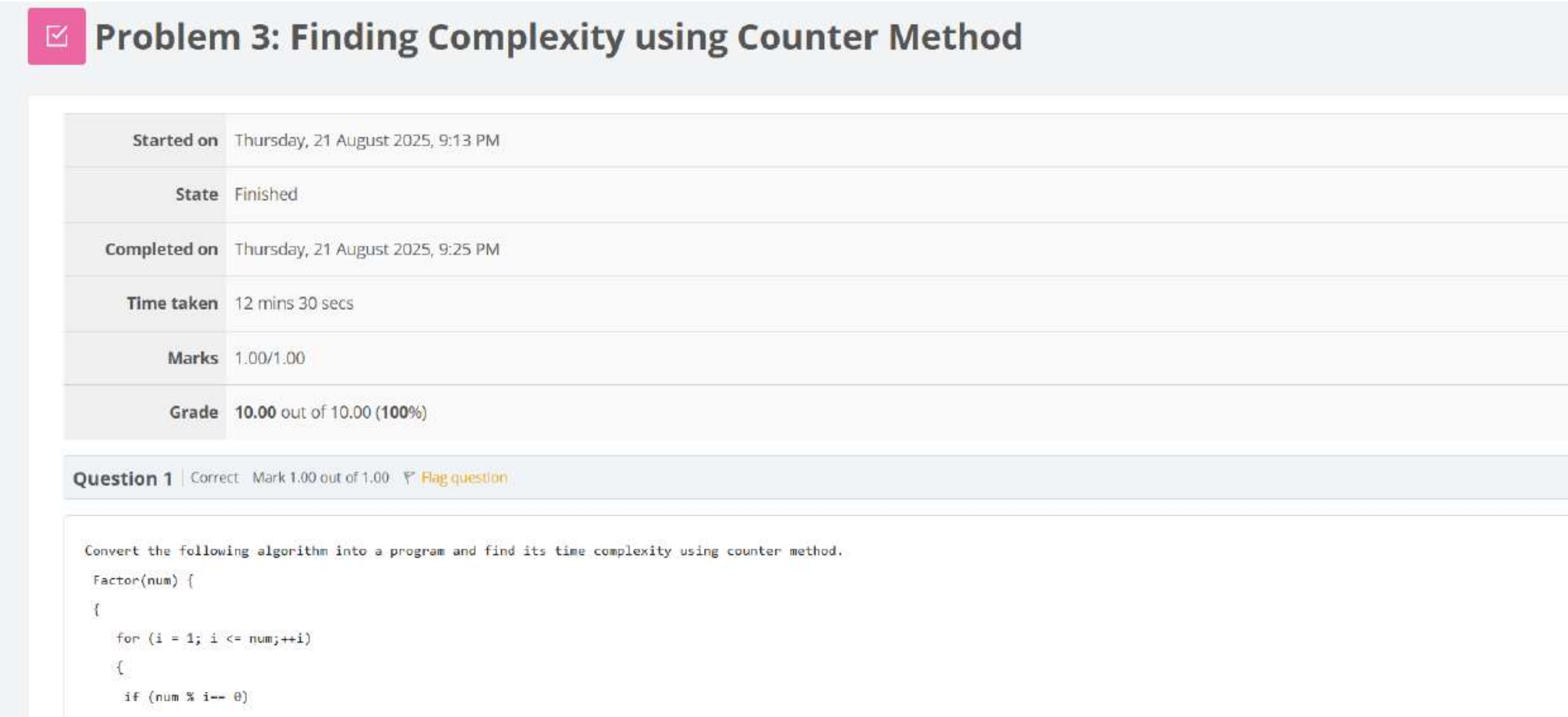


	Input	Expected	Got	
✓	2	12	12	✓
✓	1000	5002	5002	✓
✓	143	717	717	✓

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.



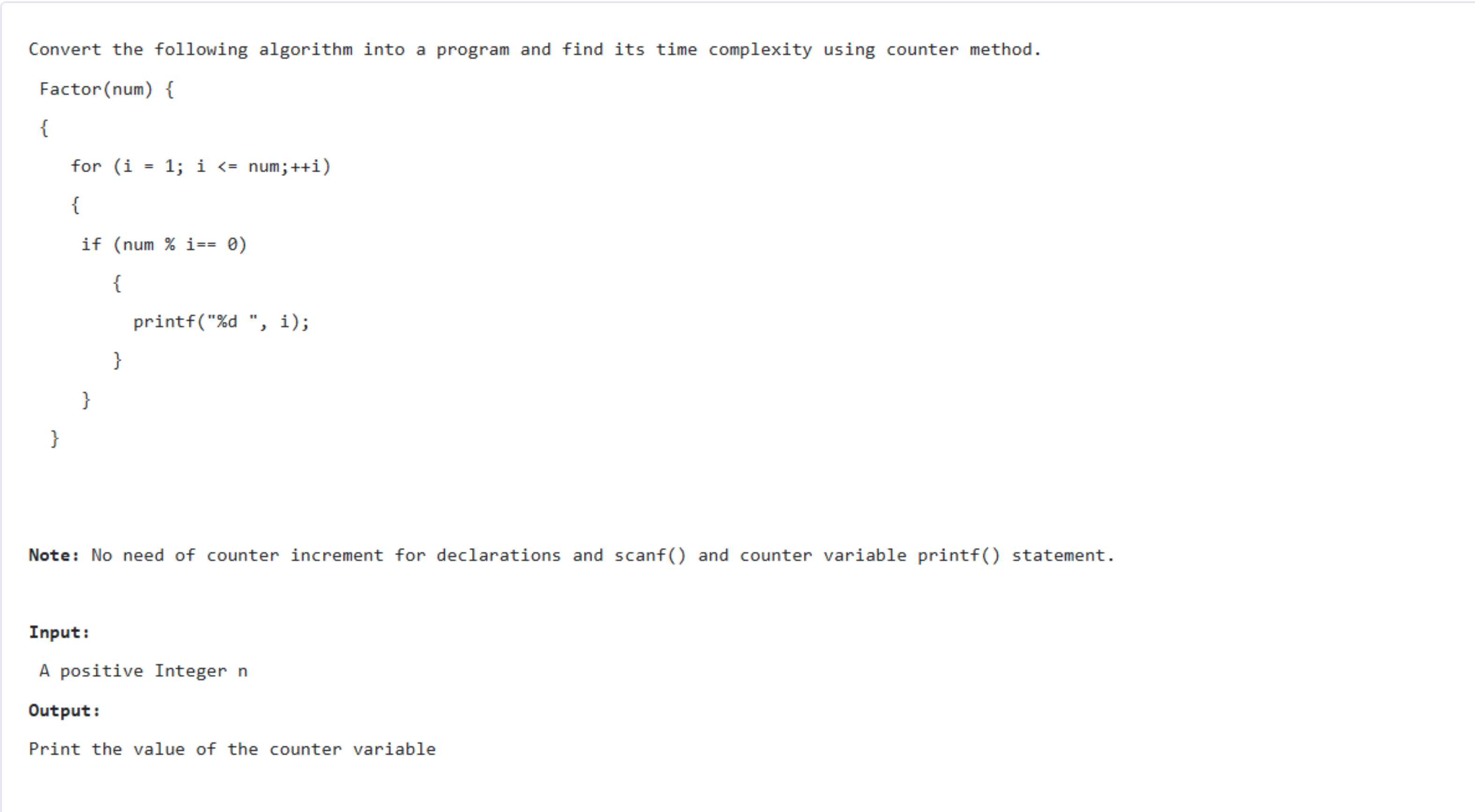
Problem 3: Finding Complexity using Counter Method

Started on	Thursday, 21 August 2025, 9:13 PM
State	Finished
Completed on	Thursday, 21 August 2025, 9:25 PM
Time taken	12 mins 30 secs
Marks	1.00/1.00
Grade	**10.00** out of 10.00 (**100%**)

Question 1 Correct Mark 1.00 out of 1.00

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)
```



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.

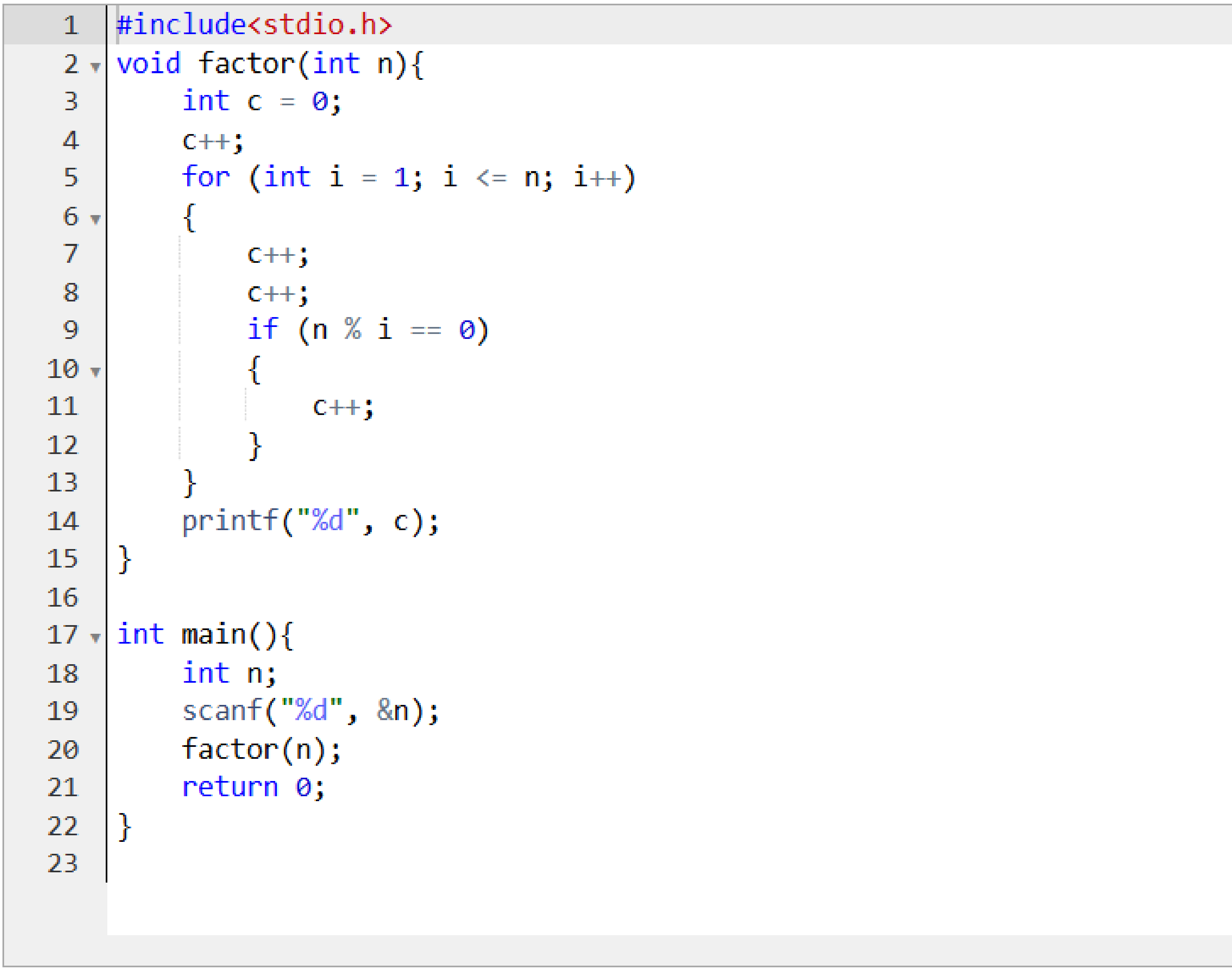
Input:

A positive Integer n

Output:

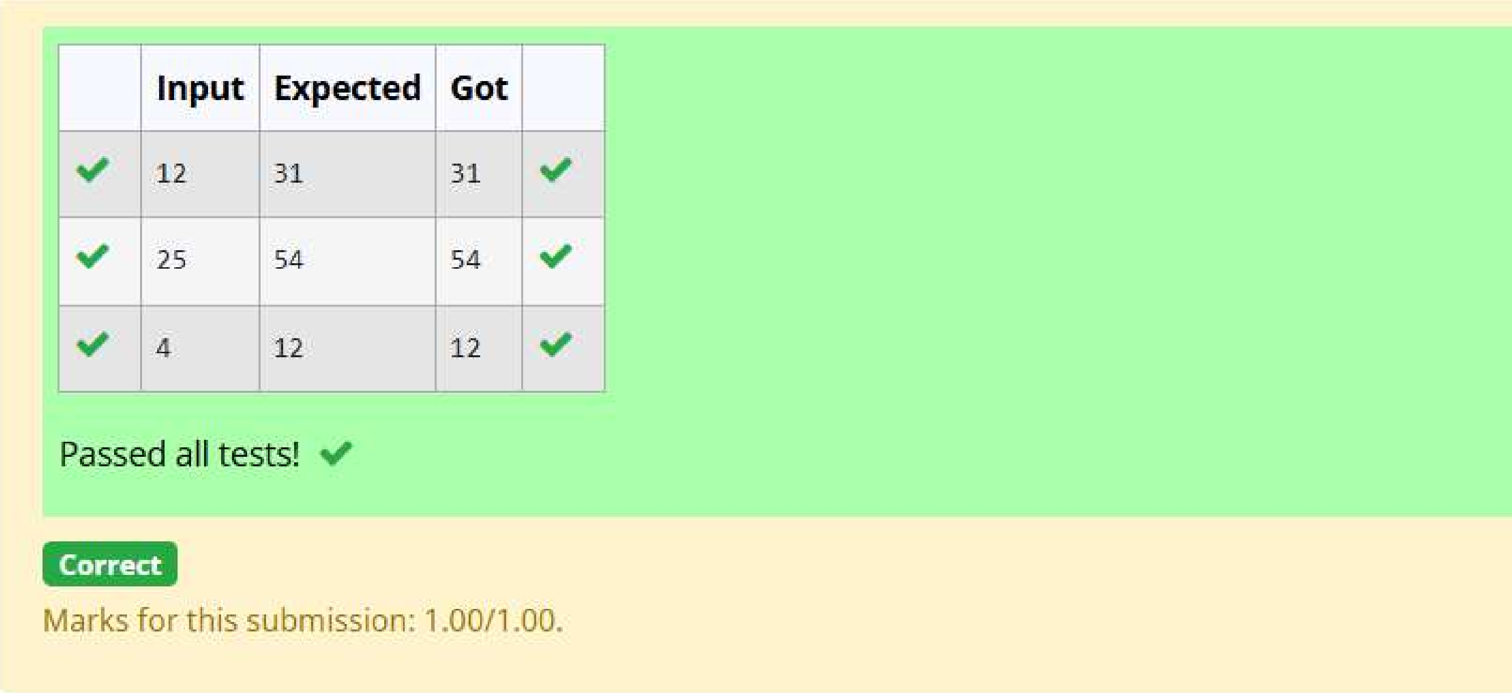
Print the value of the counter variable

Answer:



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

int main(){
    int n;
    scanf("%d", &n);
    factor(n);
    return 0;
}
```

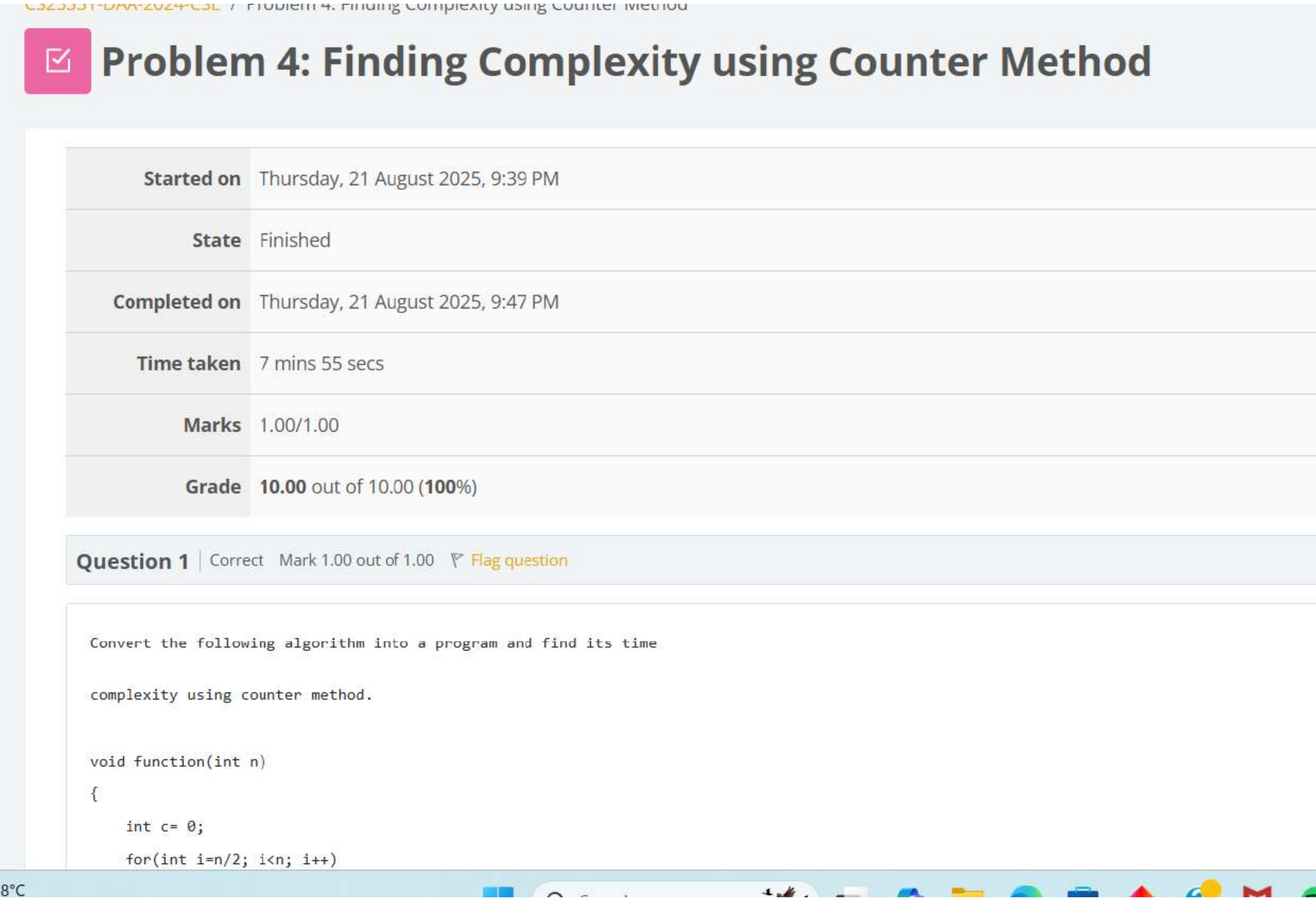


	Input	Expected	Got	
✓	12	31	31	✓
✓	25	54	54	✓
✓	4	12	12	✓

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.



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

Problem 4: Finding Complexity using Counter Method

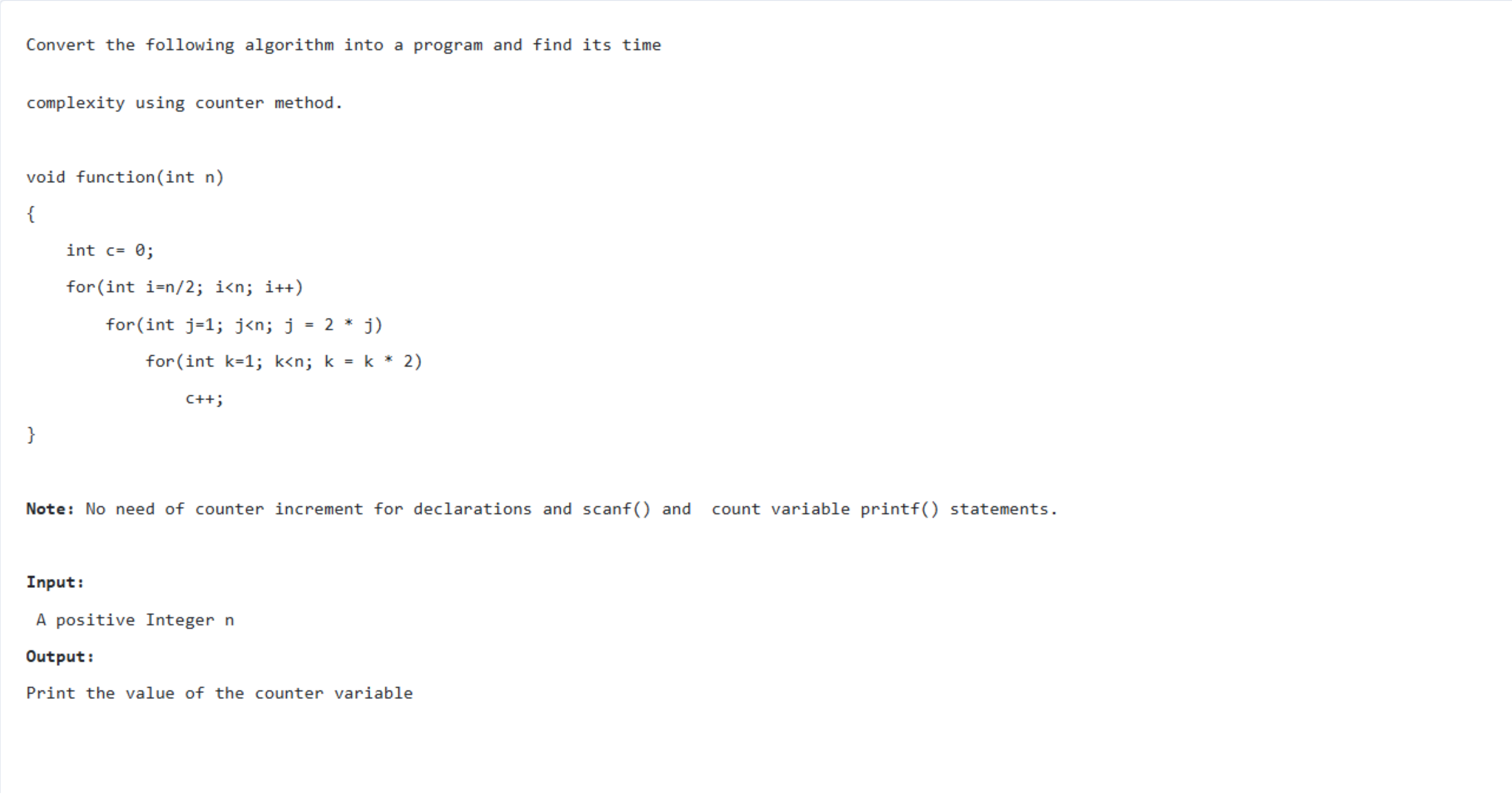
Started on	Thursday, 21 August 2025, 9:39 PM
State	Finished
Completed on	Thursday, 21 August 2025, 9:47 PM
Time taken	7 mins 55 secs
Marks	1.00/1.00
Grade	**10.00** out of 10.00 (**100**%)

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 counter method.


void function(int n)
{
    int c= 0;
    for(int i=n/2; i<n; i++)
```



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++;
}
```

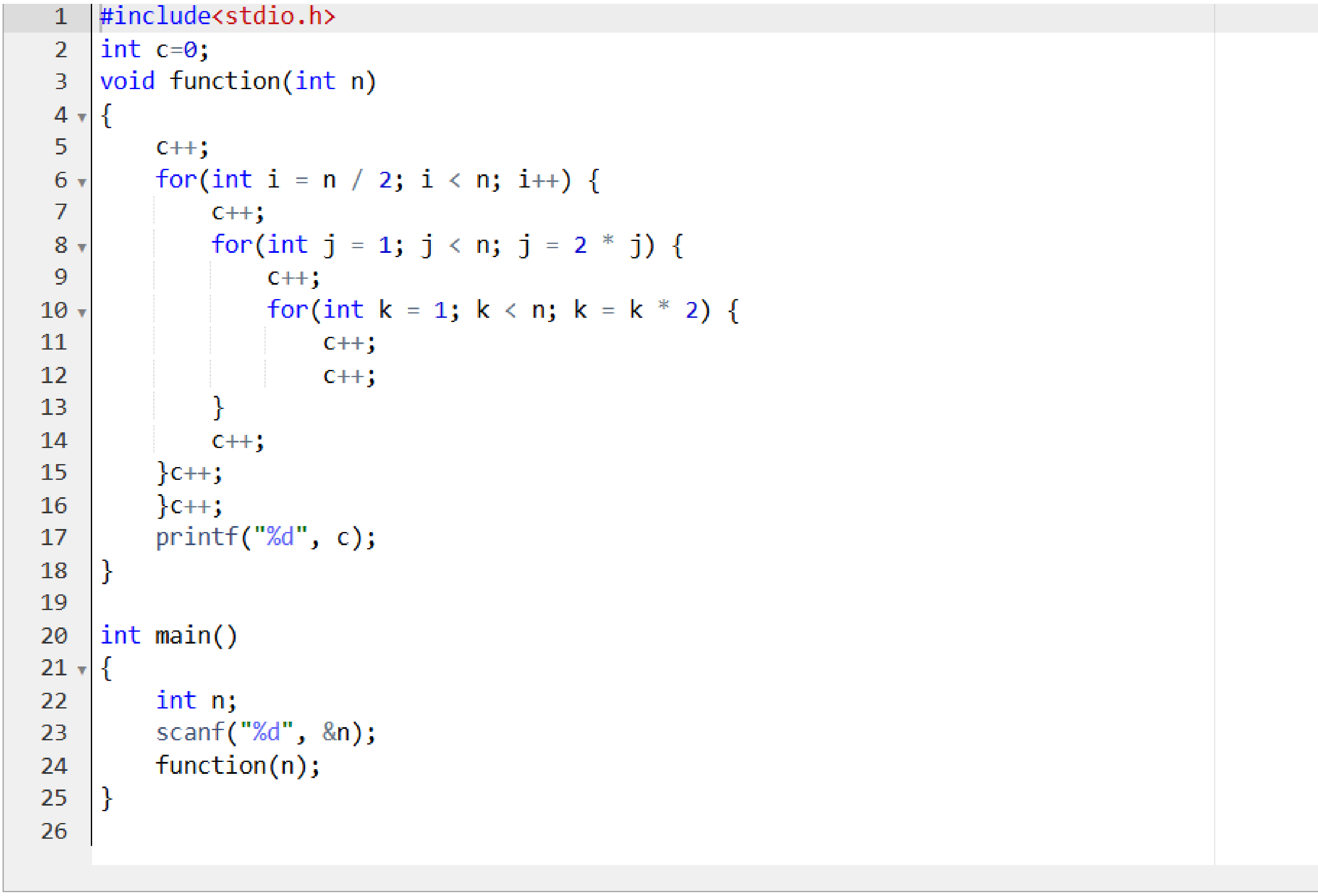
Note: No need of counter increment for declarations and scanf() and count variable printf() statements.

Input:

A positive Integer n

Output:

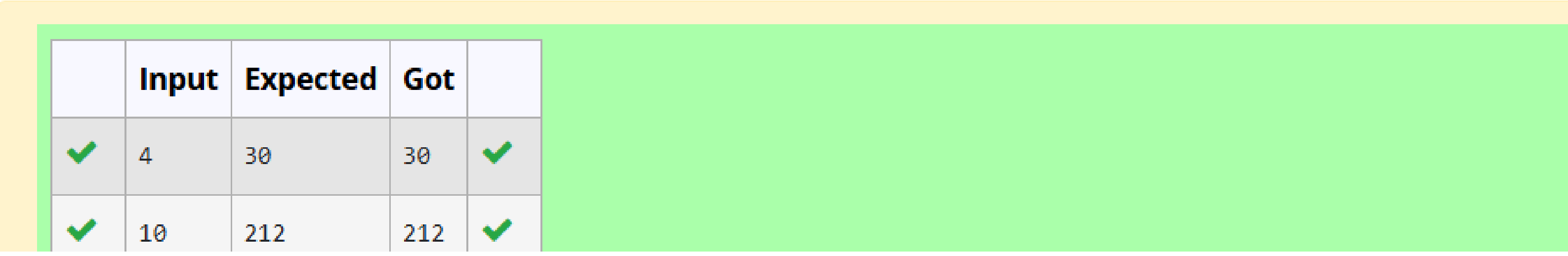
Print the value of the counter variable



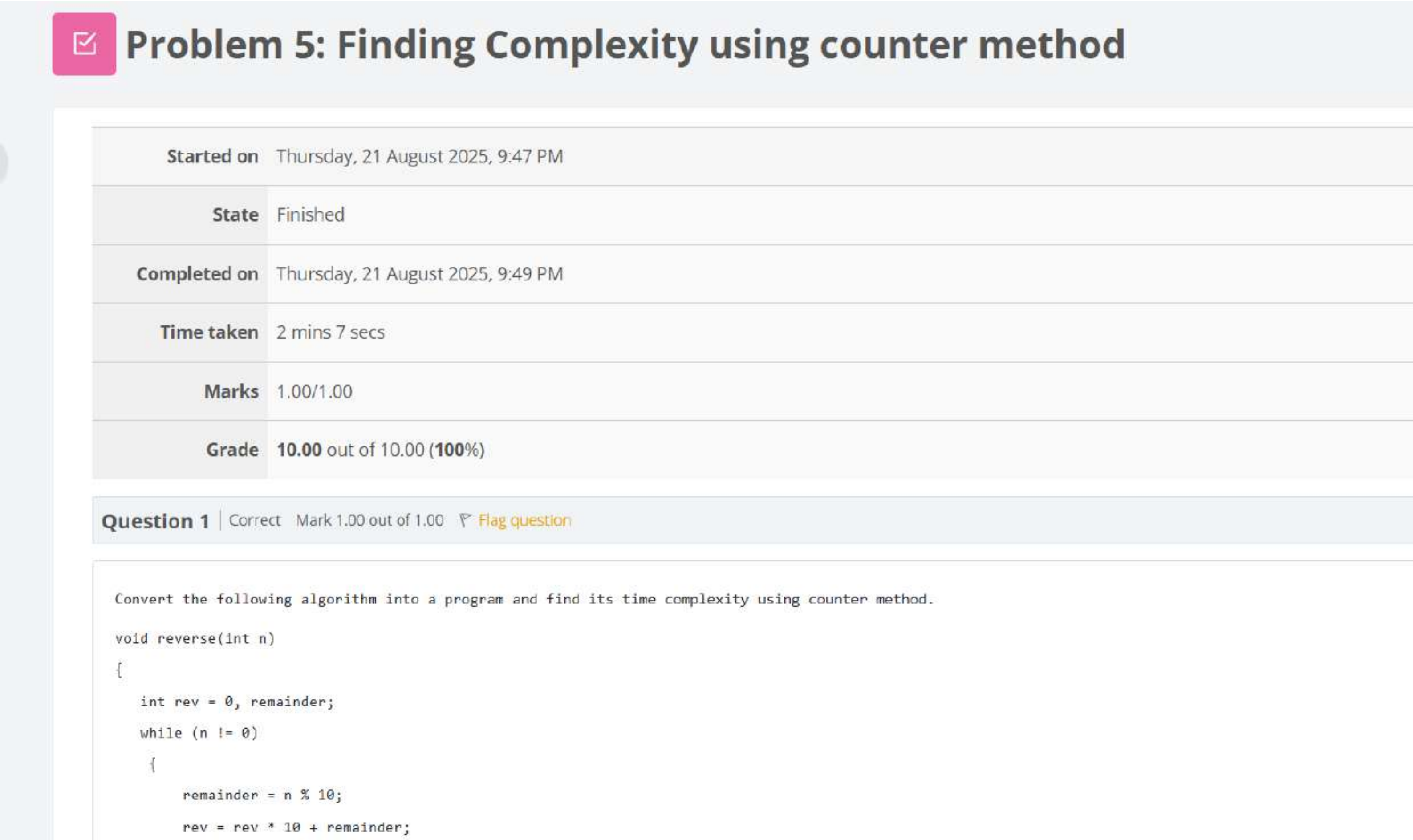
```c
#include<stdio.h>
int c=0;
void function(int n)
{
    c++;
    for(int i = n / 2; i < n; i++) {
        c++;
        for(int j = 1; j < n; j = 2 * j) {
            c++;
            for(int k = 1; k < n; k = k * 2) {
                c++;
                c++;
            }
            c++;
      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

int main()
{
    int n;
    scanf("%d", &n);
    function(n);
}

```



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



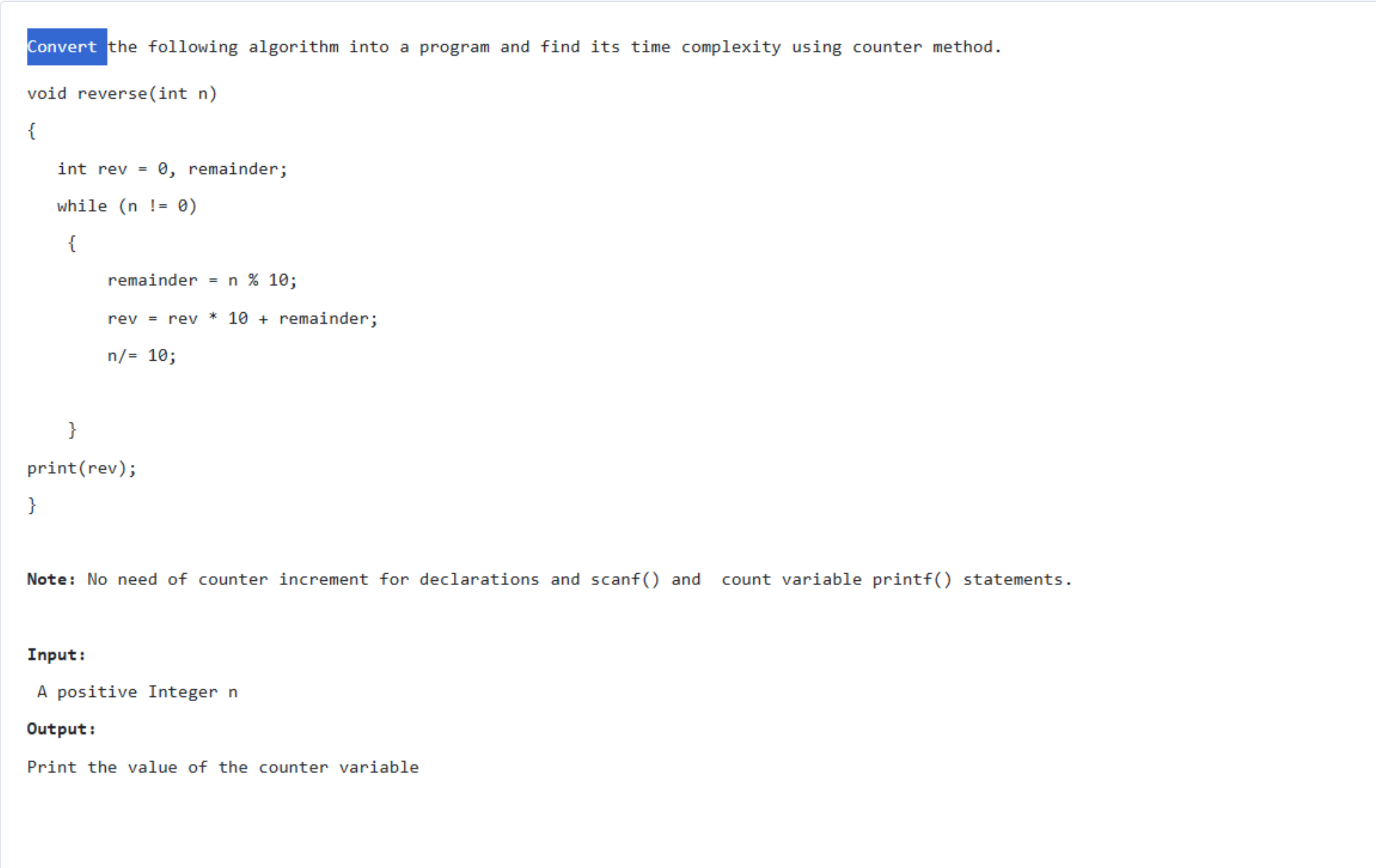
Problem 5: Finding Complexity using counter method

Started on	Thursday, 21 August 2025, 9:47 PM
State	Finished
Completed on	Thursday, 21 August 2025, 9:49 PM
Time taken	2 mins 7 secs
Marks	1.00/1.00
Grade	**10.00** out of 10.00 (**100**%)

Question 1 Correct Mark 1.00 out of 1.00

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;
```



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);
}
```

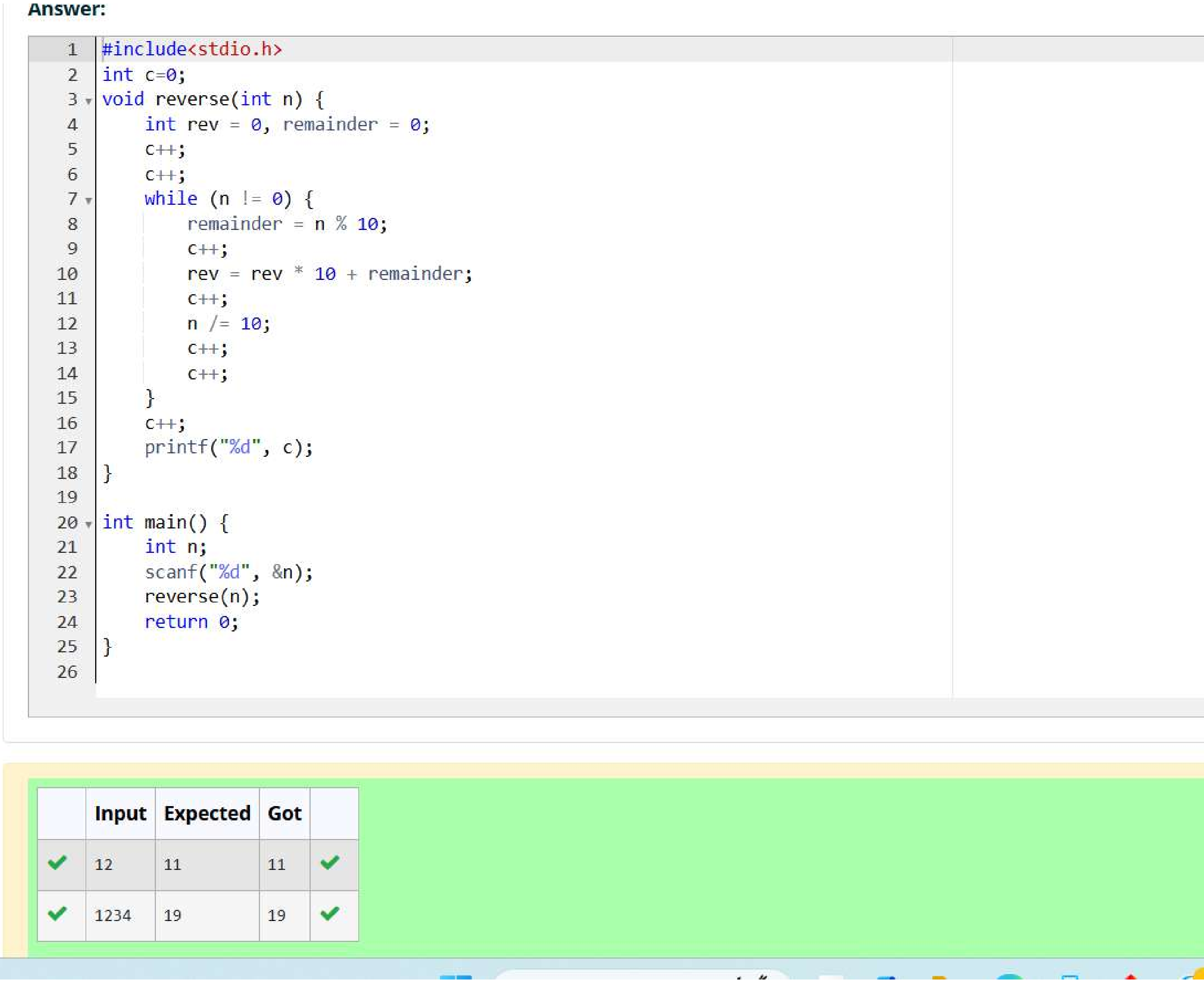
Note: No need of counter increment for declarations and scanf() and count variable printf() statements.

Input:

 A positive Integer n

Output:

Print the value of the counter variable



Answer:

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

int main() {
    int n;
    scanf("%d", &n);
    reverse(n);
    return 0;
}

```

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