

Problem 1: Finding Complexity using Counter Method

Started on Thursday, 7 August 2025, 8:07 PM

State Finished

Completed on Thursday, 7 August 2025, 8:13 PM

Time taken 5 mins 29 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 %)

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

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

Passed all tests! ✓

Correct



Problem 2: Finding Complexity using Counter method



Started on Wednesday, 6 August 2025, 9:05 AM

State Finished

Completed on Wednesday, 6 August 2025, 9:22 AM

Time taken 17 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](#)

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 int co=0;
3 void func(int n)
4 {
5
6     co=(5*n)+2;
7     printf("%d",co);
8 }
9
10 int main(){
11
12     int n;
13     scanf("%d",&n);
14     func(n);
15 }
```

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

Passed all tests! ✓



Problem 3: Finding Complexity using Counter Method



Started on Wednesday, 6 August 2025, 9:31 AM



State Finished

Completed on Saturday, 9 August 2025, 5:42 PM

Time taken 3 days 8 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:

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

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

Passed all tests! ✓



Problem 4: Finding Complexity using Counter Method



Started on Saturday, 9 August 2025, 5:50 PM

State Finished

Completed on Saturday, 9 August 2025, 6:13 PM

Time taken 22 mins 52 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:

```
1 #include<stdio.h>
2 int count=0;
3 void function(int n)
4 {
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                c++;
14                count++;
15            }
16            count++;
17        }
18        count++;
19    }
20    count++;
21    printf("%d",count);
22 }
23 int main(){
24     int n;
25     scanf("%d",&n);
26     function(n);
27     return 0;
28 }
```

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

Passed all tests! ✓



Problem 5: Finding Complexity using counter method



Started on Saturday, 9 August 2025, 6:16 PM

State Finished

Completed on Saturday, 9 August 2025, 6:27 PM

Time taken 11 mins 29 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 int count=0;
3 void reverse(int n)
4 {
5     int rev=0,remainder;
6     count++;
7     while(n!=0)
8     {
9         count++;
10        remainder=n%10;
11        count++;
12        rev=rev*10+remainder;
13        count++;
14        n/=10;
15        count++;
16    }
17    count++;
18    count++;
19    printf("%d",count);
20 }
21 int main()
22 {
23     int n;
24     scanf("%d",&n);
25     reverse(n);
26     return 0;
27 }
```

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

Passed all tests! ✓