# Finding Complexity using Counter Method

Roll no.: 241801266

Name: shyam ganesh d

1

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 %)

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

| | Input | Expected | Got | |
|---|---|---|---|---|
| ✔ | 9 | 12 | 12 | ✔ |
| ✔ | 4 | 9 | 9 | ✔ |

Passed all tests! ✔

2

**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 %)

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

| | Input | Expected | Got | |
|---|---|---|---|---|
| ✔ | 2 | 12 | 12 | ✔ |
| ✔ | 1000 | 5002 | 5002 | ✔ |
| ✔ | 143 | 717 | 717 | ✔ |

Passed all tests! ✔

3

Started on     Saturday, 9 August 2025, 5:53 PM

State   Finished

Completed on   Saturday, 9 August 2025, 5:58 PM

Time taken   4 mins 15 secs

Marks   1.00/1.00

Grade   **10.00** out of 10.00 (**100%**)

**Question 1** | Correct   Mark 1.00 out of 1.00   ⚑ Flag question

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

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

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

**Input:**
 A positive Integer n
**Output:**
Print the value of the counter variable

**Answer:**

```c
#include<stdio.h>
void factor(int num){
    int count=0;
    for(int i=1;i<=num;++i){
        count++;
        if(num%i==0){
            count++;

        }
        count++;

    }
    count++;
    printf("%d ",count);
}
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! ✔

4

| Started on | Saturday, 9 August 2025, 5:58 PM |
|---|---|
| State | Finished |
| Completed on | Saturday, 9 August 2025, 6:03 PM |
| Time taken | 4 mins 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

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
1   #include<stdio.h>
2   void function(int n)
3   {
4       int c= 0;
5       int count=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! ✔

5

| | |
|---|---|
| **Started on** | Saturday, 9 August 2025, 6:03 PM |
| **State** | Finished |
| **Completed on** | Saturday, 9 August 2025, 6:06 PM |
| **Time taken** | 3 mins 24 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:**

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

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

| | Input | Expected | Got | |
|---|---|---|---|---|
| ✔ | 12 | 11 | 11 | ✔ |
| ✔ | 1234 | 19 | 19 | ✔ |

Passed all tests! ✔