

**RAJALAKSHMI ENGINEERING COLLEGE**  
**RAJALAKSHMI NAGAR, THANDALAM – 602 105**



**RAJALAKSHMI**  
**ENGINEERING COLLEGE**

**CS23331**  
**DESIGN AND ANALYSIS OF ALGORITHM LAB**

**Laboratory Observation Note Book**

Name : Nithesh.kumar S .....

Year / Branch / Section : . 2<sup>nd</sup> Year/ AIML / B .....

Register No. : . . . 231501114 .....

Semester : . . . 3<sup>rd</sup> Semester .....

Academic Year : . . 2024-2025 .....

## **WEEK – 02**

### **Finding Complexity using Counter Method**

**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

**CODE:**

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

```
while(s <= n)
{
    c++;
    i++;
    c++;
    s += i;
    c++;
}
c++;
printf("%d",c);
}
int main()
{
    int n;
    scanf("%d",&n);
    function(n);
}
```

## OUTPUT:

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

Passed all tests! ✓

**Correct**

Marks for this submission: 1.00/1.00.

**2) 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**

**CODE:**

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

```
{
    c++;
    printf("*");
    c++;
}
else
{
    c++;
    for(int i=1; i<=n; i++)
    {
        c++;
        for(int j=1; j<=n; j++)
        {
            c++;
            //printf("*");
            c++;
            //printf("*");
            c++;
            break;
        }
        c++;
    }
    c++;
}
printf("%d",c);
}
int main()
{
    int n;
    scanf("%d",&n);
    func(n);
}
```

## OUTPUT:

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

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

**3) 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**

**CODE:**

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



```

        //printf("%d ", i);
        c++;
    }

}
c++;
printf("%d",c);

}
int main()
{
    int num;
    scanf("%d",&num);
    Factor(num);
}

```

### **OUTPUT:**

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

Passed all tests! ✓

**Correct**

Marks for this submission: 1.00/1.00.

**4) 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**

**CODE:**

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

```

        count++;
        c++;
        count++;
    }
    count++;
}
count++;
printf("%d",count);
}

```

```

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

```

## OUTPUT:

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

Passed all tests! ✓

**Correct**

Marks for this submission: 1.00/1.00.

**5) 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**

**CODE:**

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

```

        count++;
        rev = rev * 10 + remainder;
        count++;
        n/= 10;
        count++;
    }
    count++;
    //printf("%d",rev);
    count++;
    printf("%d",count);
}

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

```

## OUTPUT:

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

Passed all tests! ✓

**Correct**

Marks for this submission: 1.00/1.00.