

**NAME: MADHAN RV**

**REG NO: 241801142**

## **FINDING TIME COMPLEXITY OF ALGORITHMS USING COUNTER METHOD**

### **PROGRAM 1:**

**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

Ace editor not ready. Perhaps reload page?

Falling back to raw text area.

```
#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);
```

	<b>Input</b>	<b>Expected</b>	<b>Got</b>	
✓	9	12	12	✓
✓	4	9	9	✓

Passed all tests! ✓

## PROGRAM 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

**Answer:** (penalty regime: 0 %)

```
1 #include<stdio.h>
2 void func(int n)
3 {
4     int count=0;
5     if(n==1)
6     {
7         count++;
8         //printf("*");
9     }
10    else
11    {
12        count++;
13        for(int i=1; i<=n; i++)
14        {
15            count++;
16            for(int j=1; j<=n; j++)
17            {
18                count++;
19                count++;
20                count++;
21                //printf("*");
22                //printf("*");
23                break;
24            }
25            count++;
26        }
27        count++;
28    }
29    printf("%d",count);
30 }
31 int main()
32 {
33     int n;
34     scanf("%d",&n);
35     func(n);
36 }
```

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

Passed all tests! ✓

Correct

## PROGRAM 3:

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 void Factor(int num)
3 {
4     int count=0;
5     for (int i = 1; i <= num; ++i)
6     {
7         count++;
8         if (num % i== 0)
9         {
10             count++;
11             //printf("%d ", i);
12         }
13         count++;
14     }
15     count++;
16     printf("%d",count);
17 }
18 int main()
19 {
20     int n;
21     scanf("%d",&n);
22     Factor(n);
23 }
```

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

## PROGRAM 4:

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

```
1 #include<stdio.h>
2 void reverse(int n)
3 {
4     int count=0;
5     int rev = 0, remainder;
6     count++;
7     count++;
8     while (n != 0)
9     {
10         count++;
11         remainder = n % 10;
12         count++;
13         rev = rev * 10 + remainder;
14         count++;
15         n/= 10;
16         count++;
17     }
18     count++;
19     //print(rev);
20     printf("%d",count);
21 }
22 int main()
23 {
24     int n;
25     scanf("%d",&n);
26     reverse(n);
27 }
```

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

Passed all tests! ✓