

# FINDING TIME COMPLEXITY OF ALGORITHM

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

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

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

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

```
void func(int n)
{
    if(n==1)
    {
        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         count++;
10    }
11    else
12    {
13        count++;
14        for(int i=1; i<=n; i++)
15        {
16            count++;
17            for(int j=1; j<=n; j++)
18            {
19                count++;
20                // printf("*");
21                count++;
22                //printf("*");
23                count++;
24                break;
25            }
26            count++;
27        }
28        count++;
29    }
30    printf("%d",count);
31 }
32 int main(){
33     int n;
34     scanf("%d",&n);
35     func(n);
36     return 0;
37 }
38
```

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

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

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

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

Passed all tests! ✓

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 void function(int n)
3 {
4     int count = 0;
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! ✓

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 void reverse(int n)
3 {
4     int count =0;
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    //printf("%d",rev);
19    count++;
20    printf("%d",count);
21 }
22 int main(){
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! ✓