Name: Valluru Varshini

Class: CSE – F

Reg no: 230701369

WEEK 2: FINDING TIME COMPLEXITY

PROGRAM 1:

AIM:

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;
}
}
```

ALGORITHM:

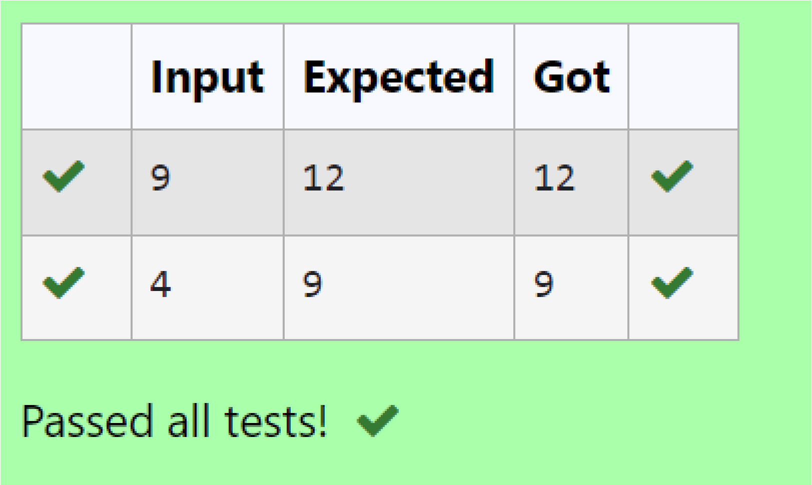
Step 1: Initiliaze a counter variable c=0 Step 2: Place c++ after each statement Step 3: Display c

PROGRAM:

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

```
    }
    count++;
}

int main()
{
    int n;
    scanf("%d",&n);
    func(n);
    printf("%d",count);
}
```

OUTPUT:



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

Passed all tests! ✓

RESULT:

Thus the program is executed successfully.

PROGRAM 2:

AIM:

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;
}
}

}
}
```

ALGORITHM:

Step 1: initialize a counter variable c=0

Step 2: Place c++ after each iteration of a loop and declaration of a statement. Step 3: Display c

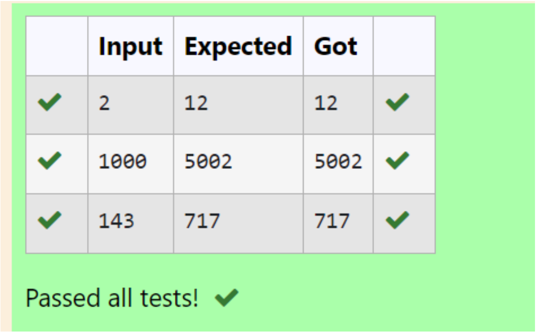
PROGRAM:

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

```
{
    count++;
    //printf("*");
}
else{
    count++;
    for(int i=1;i<=n;i++)
    {
        count++;
        for(int j=1;j<=n;j++)
        {
            count++;
            //printf("*");
            count++;
            //printf("*");
            count++;
            break;
        }
        count++;
    }
```

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

OUTPUT:



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

Passed all tests! ✓

RESULT:

Thus the program executed successfully.

PROGRAM 3:

AIM:

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);
}
}
}
```

ALGORITHM:

Step 1: initialize a variable c=0

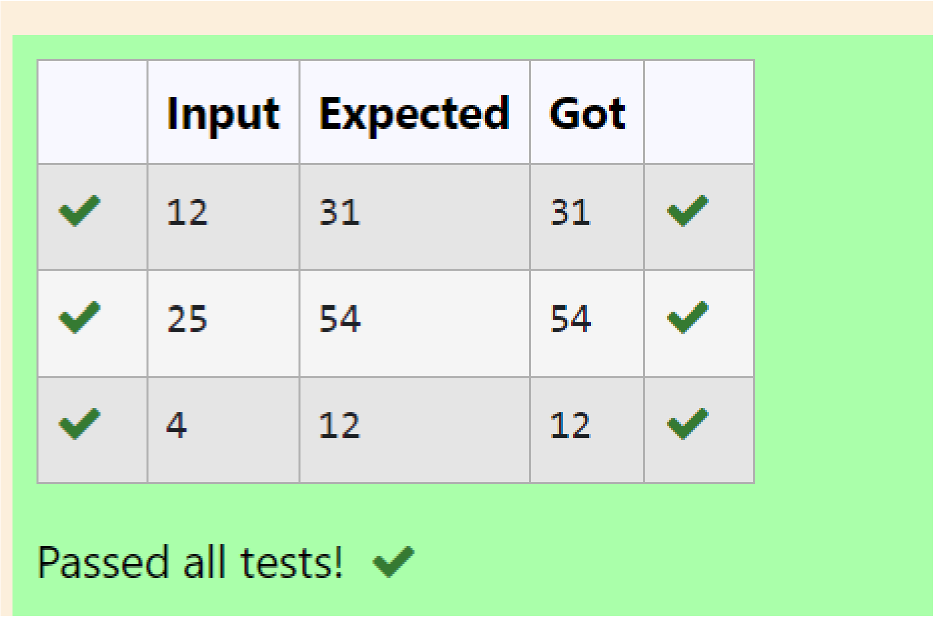
Step 2: Place c++ after each iteration of a loop. Step 3: display c

PROGRAM :

```
#include <stdio.h>

int count=0;

void Factor(int);

void Factor(int num)

 {

   for(int i=1;i<=num;++i)

   {

      count++;

      if(num%i==0)

      {

         count++;

         //printf("%d",i);

      }
```

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

OUTPUT:



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

Passed all tests! ✓

RESULT:

Thus the program is executed successfully.

PROGRAM 4:

AIM:

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++;
}
```

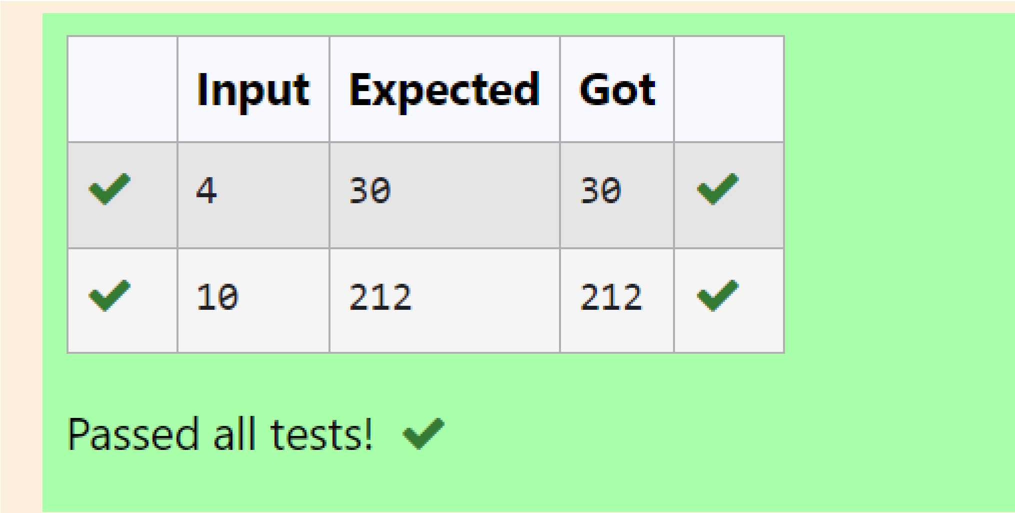
ALGORITHM:

Step 1: Initialize a counter variable c=0 Step 2: Place c++ after every loop Step 3: display c

PROGRAM:

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

OUTPUT:



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

Passed all tests! ✓

RESULT: Thus the program is executed successfully

PROGRAM 5:

AIM:

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);

}
```

ALGORITHM:

Step 1: Initialise the counter variable c=0

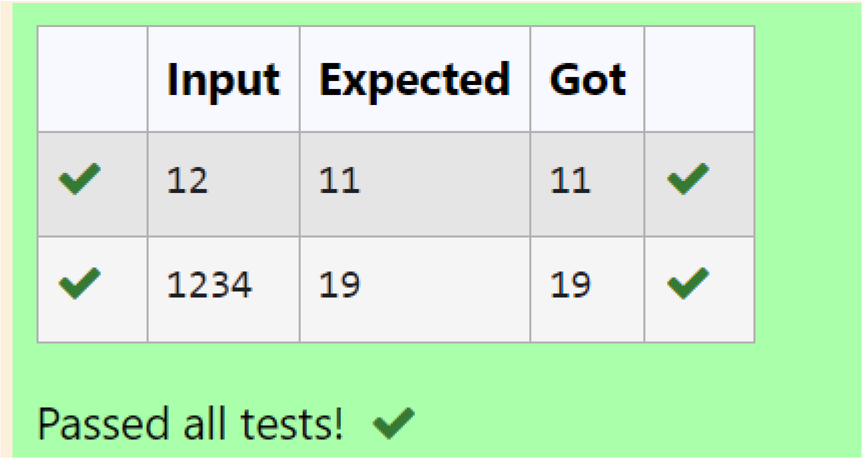
Step 2: After every iteration of a loop place a c++ Step 3: Display c

PROGRAM:

```
#include<stdio.h>

int count=0;

void reverse(int n)

{

   int rev=0,remainder;

   count++;

   while(n!=0)
```

```
    {
        count++;
        remainder=n%10;
        count++;
        rev=rev*10+remainder;
        count++;
        n/=10;
        count++;
    }
    count++;
    //print(rev)
    count++;
}
int main()
{
    int n;
    scanf("%d",&n);
    reverse(n);
    printf("%d",count);
}
```

OUTPUT:



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

Passed all tests! ✓

RESULT:

Thus the program executed successfully.