Name	Aravinthaa S
Register No	230701032
Dept	CSE A

TIME COMPLEXITY OF ALGORITHMS

Ex No – 1.1	Finding Time Complexity using Counter Method

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

Aim:

To convert the given algorithm into a program and determine its time complexity using the counter method.

Program :

```
#include<stdio.h>

int main()

{

   int count=0;

   int i=1;

   count++;

   int s=1;

   count++;
```

```
    int n;
    scanf("%d",&n);

    while(s<=n)
    {
        count++;
        i++;
        count++;
        s+= i;
        count++;
    }
    count++;
    printf("%d",count);
}
```

Input and Output:

	Input	Expected	Got	
✔	9	12	12	✔
✔	4	9	9	✔

Passed all tests! ✔

Ex No – 1.2	Finding Time Complexity using Counter Method

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

Aim:

To convert the given algorithm into a program and determine its time complexity using the counter method.

Program:

```
#include <stdio.h>

void func(int n)

{

   int c = 0;

   if (n == 1)

   {

      c++;

      c++;

   }

   else

   {
```

```
        for (int i = 1; i <= n; i++)
        {
            c++;
            for (int j = 1; j <= n; j++)
            {
                c++;
                c++;
                c++;
                break;
            }
            c++;
        }
        c++;
    }
    c++;
    printf("%d\n", c);
}
int main()
{
    int n;
    scanf("%d", &n);
    func(n);
    return 0;
}
```

Input and Outut :

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

Passed all tests! ✔

Ex No – 1.3	Finding Time Complexity using Counter Method

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

Aim:

To convert the given algorithm into a program and determine its time complexity using the counter method.

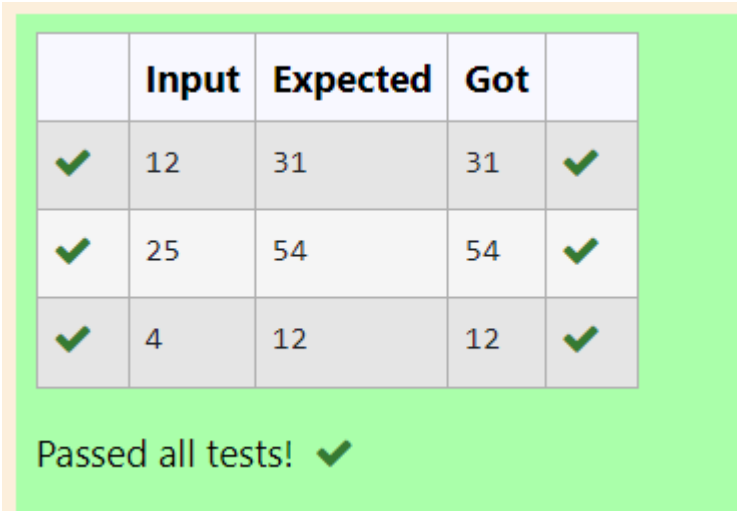
Program:

```
#include<stdio.h>

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

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

Input and Output :



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

Passed all tests! ✔

Ex No – 1.4	Finding Time Complexity using Counter Method

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

Aim:

To convert the given algorithm into a program and determine its time complexity using the counter method.

Program :

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

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

Input and Output:

	Input	Expected	Got	
✔	4	30	30	✔
✔	10	212	212	✔

Passed all tests! ✔

Ex No – 1.5	Finding Time Complexity using Counter Method

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

Aim:

To convert the given algorithm into a program and determine its time complexity using the counter method.

Program:

```
#include<stdio.h>

void reverse(int n)

{

  int c=0;

  c++;

  int rev = 0, remainder;

  while (n != 0)

  {

     c++;

     remainder = n % 10;

     c++;

     rev = rev * 10 + remainder;
```

```
        c++;
        n/= 10;
        c++;

    }
    c++;
    c++;
    printf("%d",c);
}
int main(){
    int n;
    scanf("%d",&n);
    reverse(n);
   // printf("%d",rev);
}
```

Input and Output:

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

Passed all tests! ✔