RAJALAKSHMI ENGINEERING COLLEGE RAJALAKSHMI NAGAR, THANDALAM – 602 105



CS23331 DESIGN AND ANALYSIS OF ALGORITHM LAB

Laboratory Observation Note Book

Name: KAVIYAM
Year / Branch / Section : SECOND YearAIMIB
231501075 Register No. :
Semester: 3 rd Semester
Academic Year : 2024-2025

<u>WEEK – 02</u> 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;
```

Note: No need of counter increment for declarations and scanf() and count variable printf() statements.

Input:

A positive Integer n Output:

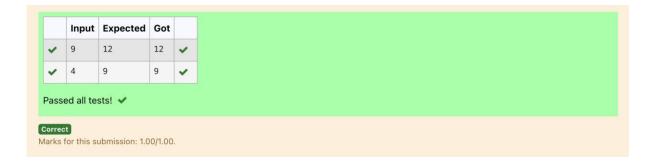
Print the value of the counter variable

For example:

Input	Result
9	12

CODE:

```
#include<stdio.h>
void function (int n)
{ int
c=0; int
i= 1;
      int
C++;
s =1;
C++;
  while(s <= n)
  {
C++;
i++;
      S
C++;
+= i;
     }
C++;
C++;
  printf("%d",c);
int main()
{
   int
n;
  scanf("%d",&n);
  function(n);
}
```



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++)
       {
            printf("*");
            printf("*");
            break;
       }
       }
    }
}</pre>
```

Note: No need of counter increment for declarations and scanf() and count variable printf() statements.

Input:

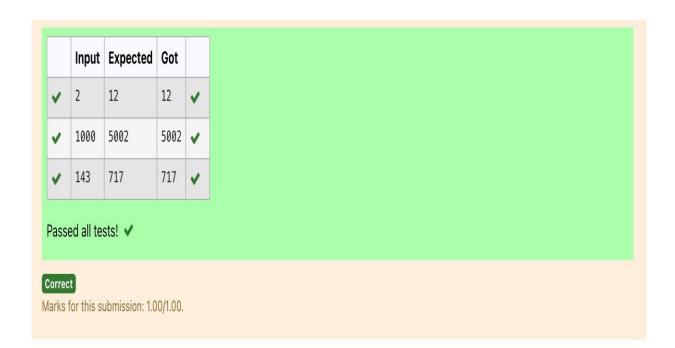
A positive Integer n

Output:

Print the value of the counter variable

```
#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("*");
          break;
C++;
C++;
   C++;
```

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



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);
      }
    }
}</pre>
```

Note: No need of counter increment for declarations and scanf() and counter variable printf() statement.

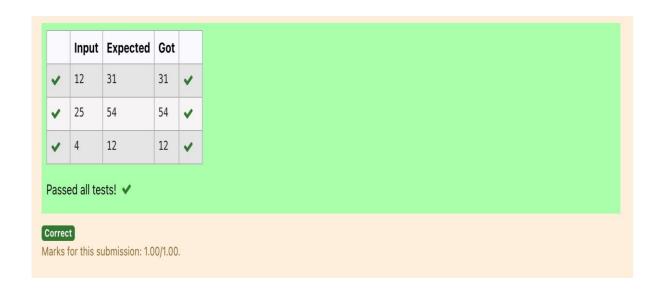
Input:

A positive Integer n Output:

Print the value of the counter variable

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

```
{ c++;
      if (num %
C++;
i== 0)
    {
      //printf("%d ", i);
C++;
  }
  C++;
printf("%d",c);
 }
int main()
{
  int num;
scanf("%d",&num);
  Factor(num);
```



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++;
}</pre>
```

Note: No need of counter increment for declarations and scanf() and count variable printf() statements.

Input:

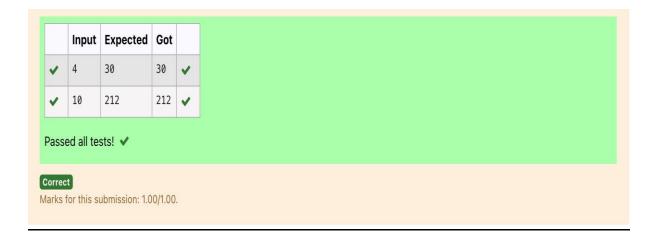
A positive Integer n Output:

Print the value of the counter variable

```
#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)
      {
}</pre>
```

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

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



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

```
#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);
}
```

