ROLL NO:230701036

NAME: ARUN MC

TOPIC: FINDING TIME COMPLIXITY OF ALGORITHM

## 1: Finding Complexity using Counter Method

**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;
    }
}
```

**CODE:**

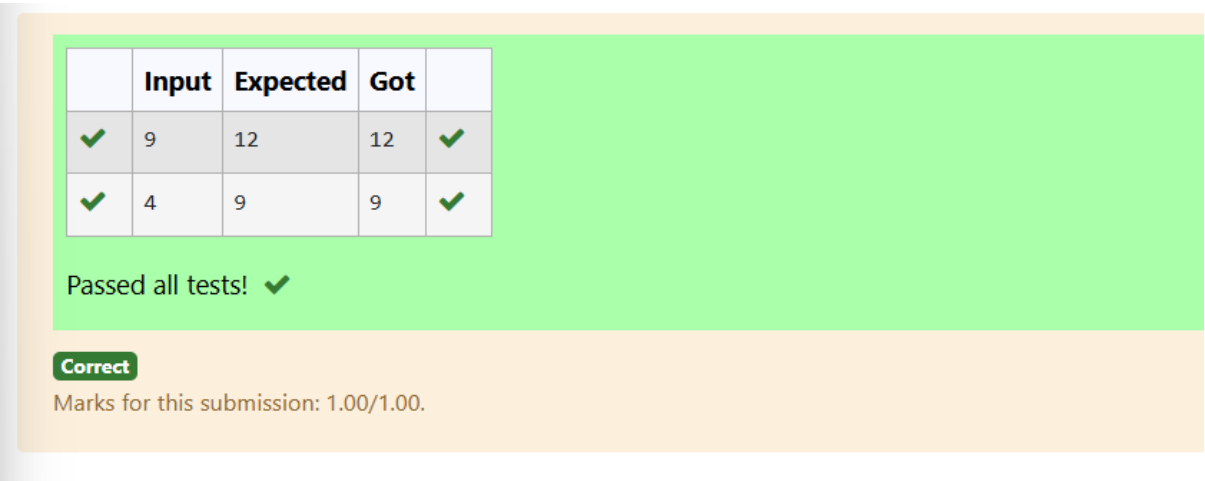
```
#include<stdio.h>
int main()
{
  int n,i=1,s=1;
  scanf("%d",&n);
  while(s<=n)
{
  i++;
```

```
  s+=i;
}
printf("%d",3*i);
}
```

**INPUT:**

| Input | Result |
|---|---|
| 9 | 12 |

**OUTPUT**:



| | Input | Expected | Got | |
|---|---|---|---|---|
| ✔ | 9 | 12 | 12 | ✔ |
| ✔ | 4 | 9 | 9 | ✔ |

Passed all tests! ✔

Correct

Marks for this submission: 1.00/1.00.

## 2: Finding Complexity using Counter method

## 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;
      }
    }
  }
}
```

**CODE:**

```
#include<stdio.h>
int main()
{
   int n;
   scanf("%d",&n);
   if(n==1)
     printf("*");
   else
   {
      for(int i=1;i<=n;i++)
        for(int j=1;j<=n;j++)
        {
           //printf("*");
          // printf("*");
           break;
        }
   }
   printf("%d",(n*5)+2);
}
```

**INPUT:**

A positive Integer n

**OUTPUT:**

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

Passed all tests! ✔

**Correct**

Marks for this submission: 1.00/1.00.

## 3: Finding Complexity using Counter Method

## AIM:

To determine the minimum distance a person needs to run to burn calories after eating burgers, using a greedy approach to find the optimal order of burger consumption that minimizes the total running distance.

**CODE:**

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);
      }
    }
 }
```

## INPUT:

A positive Integer n

## OUTPUT:

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

Passed all tests! ✔

Correct

Marks for this submission: 1.00/1.00.

## 4: Finding Complexity using Counter Method

## 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++;
}
```

## INPUT:

A positive Integer n

## CODE:

```
#include<stdio.h>
#include<math.h>
int main()
{
   int c= 0,n,a=0;
   scanf("%d",&n);
   for(int i=n/2; i<n; i++)
   {a++;
      for(int j=1; j<n; j = 2 * j)
      {a++;
```

```
        for(int k=1; k<n; k = k * 2)
        {a++;
           c++;
           a++;
          }a++;
           }a++;
           }a++;
    printf("%d ",a+1);
}
```

**OUTPUT:**

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

Passed all tests! ✔

Correct

Marks for this submission: 1.00/1.00.

## 5: Finding Complexity using counter method

## 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);
}
```

## INPUT:

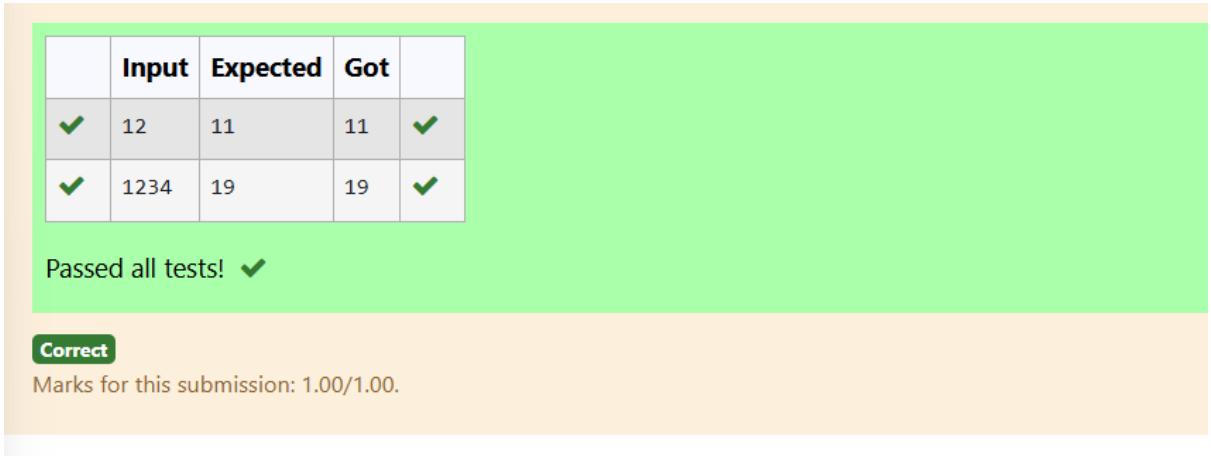
A positive Integer n

## CODE:

```
#include<stdio.h>

int main()

{

   int cnt=0;

   int rev = 0, remainder,n;

   cnt++;

   scanf("%d",&n);

  while (n != 0)

   { cnt++;

      remainder = n % 10;
```

```
        cnt++;
        rev = rev * 10 + remainder;
        cnt++;
        n/= 10;
        cnt++;

    }cnt++;
//print(rev);
cnt++;
printf("%d",cnt);
}
```

**OUTPUT:**



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

Passed all tests! ✔

Correct

Marks for this submission: 1.00/1.00.