HARIRAJ G

230701104

**DESIGN ANALYSIS AND ALGORITHM**

**FINDING TIME COMPLEXITY**

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;

    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

**Program**



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++)

{

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

**Program:**



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);  
        }          
    }   
 }  
**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

**Program:**



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++;  
}  
   
**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

**Program:**



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

**Program:**

