FINDING TIME COMPLEXITY OF ALGORITHMS

M.MOHANA

231901031

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.

Solution:

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

```
        i++;
        count++;
        s+=i;
        count++;
    }
    count+=1;
    printf("%d",count);
}
int main()
{
    int n;
    scanf("%d",&n);
    function(n);
    return 0;
}
```

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.

Solution:

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

    count++;
    //printf("*");
    count++;
  }
  else
```

```
{

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

  {
     count++;

     for(int j=1; j<=n; j++)
     {
        count++;

       // printf("*");
        count++;
        //printf("*");
        count++;
        break;
        count++;
     }
     count++;
  }
  count++;
}
count++;
```

```
  printf("%d",count);
}
int main()
{
    int n;
    scanf("%d",&n);
    func(n);
    return 0;
}
```

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.

Solution:

```
#include<stdio.h>
void Factor( int num){
   int count=0;
```

```
    for (int i = 1; i <= num;++i)
    {
       count++;
     if (num % i== 0)
     count++;
       {

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

       }
     }
     count++;
     printf("%d",count);
  }
int main()
{

    int num;
    scanf("%d",&num);
    Factor(num);
    return 0;
}
```

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.

Solution:

```
#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)

         {

            count++;


            c++;
```

```
            count++;
          }
          count++;
       }
       count++;
   }
   count++;
   printf("%d",count);
}
int main()
{
   int n;
   scanf("%d",&n);
   function(n);
   return 0;

}
```

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() statemets.

Solution:

```
#include<stdio.h>
void reverse(int n)
{
   int count=0;
   int rev = 0;
   count++;
   int remainder;
   while (n != 0)



   {
      count++;
      remainder = n % 10;
      count++;
      rev = rev * 10 + remainder;
      count++;
      n/= 10;
      count++;


   }
   count++;
```

```
//print(rev);
count++;
printf("%d",count);
}
int main()
{
   int n;
   scanf("%d",&n);
   reverse(n);
   return 0;
}
```