B.NIKITHA
230701211
CSE -'D'
III SEM
FINDING TIME 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;
    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

Answer:(penalty regime: 0 %)

```
#include<stdio.h>
void function(int n);
int main(){
    int n;
    scanf("%d",&n);
    function(n);
}
void function(int n){
    int i=1;
    int s=1;
    int count=2;
    while(s<=n){
        count++;
        count++;
        i++;
        s+=i;
        count++;
    }
    count++;
    printf("%d",count);
}
```

Feedback

Input Expected Got
9 12 12
4 9 9
Passed all tests!

Correct
Marks for this submission: 1.00/1.00.

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

Answer:(penalty regime: 0 %)

```
#include<stdio.h>
void func(int n);
int main(){
    int n;
    scanf("%d",&n);
    func(n);
}
void func(int n){
    int count=0;
    count++;
    if(n==1){
        count++;
    }
    else{
        for(int i=1;i<=n;i++){
            count++;
```

```
            count++;
            for(int j=1;j<=n;j++){
                count++;
                count++;
                break;
            }
            count++;
        }
        count++;
    }
    printf("%d",count);
}
```

Feedback

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.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

Answer:

```
#include<stdio.h>
void factor(int num);
int main(){
    int n;
    scanf("%d",&n);
    factor(n);
}
void factor(int num){
    int count=0;
    for(int i=1;i<=num;++i){
        count++;
        count++;
```

```
        if(num%i==0){
            count++;

         }

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

Feedback

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.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

Answer:

```
#include<stdio.h>
void function(int n);
int main(){
    int n;
    scanf("%d",&n);
    function(n);

}
void function(int n){
 int c=0;
 int count=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);
}
```

Feedback

Input	Expected	Got
4	30	30
10	212	212

Passed all tests!

Correct

Marks for this submission: 1.00/1.00.

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

Answer:

```
#include<stdio.h>
void reverse(int n);
int main(){
    int n;
    scanf("%d",&n);
    reverse(n);

}
void reverse(int n)
{
    int count=2;
   int rev = 0, remainder;
   while (n != 0)
```

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

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

Feedback

Input	Expected	Got
12	11	11
1234	19	19

Passed all tests!

Correct

Marks for this submission: 1.00/1.00.