NAME: Mageshwaran Ganesh

REG NO: 241801147

FINDING TIME COMPLEXITY OF ALGORITHMS

PROGRAM 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

Ace editor not ready. Perhaps reload page?

Falling back to raw text area.

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

	Input	Expected	Got	
✔	9	12	12	✔
✔	4	9	9	✔

Passed all tests! ✔

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

Answer: (penalty regime: 0 %)

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

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

Passed all tests! ✔

Correct

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

Answer:

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

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

PROGRAM 4:

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

```c
#include<stdio.h>
void reverse(int n)
{
    int count=0;
    int rev = 0, remainder;
    count++;
    count++;
    while (n != 0)
    {
        count++;
        remainder = n % 10;
        count++;
        rev = rev * 10 + remainder;
        count++;
        n/= 10;
        count++;
    }
    count++;
    //print(rev);
    printf("%d",count);
}
int main()
{
    int n;
    scanf("%d",&n);
    reverse(n);
}
```

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

Passed all tests! ✔