2.e. 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);
}
```

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

Algorithm:

```
function reverse(n) {
    initialize count to 0
    initialize rev to 0
    initialize remainder


    increment count by 1 // for initialization


    // loop until n is not equal to 0
    while n is not equal to 0 {
        increment count by 1 // start of loop


        remainder = n modulo 10
        increment count by 1 // after calculating remainder

        rev = rev * 10 + remainder
        increment count by 1 // after updating rev

        n = n divided by 10
        increment count by 1 // after updating n
    }

    increment count by 1 // after loop ends

    // simulate printing rev (e.g., print(rev))
    increment count by 1 // for print statement

    print count
}
```

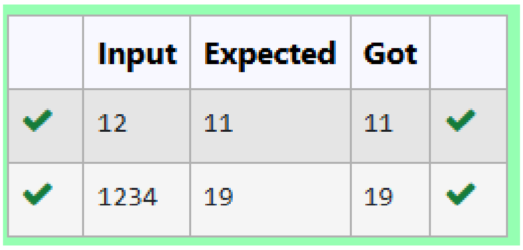
Program:

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

Output:



	Input	Expected	Got	
✓	12	11	11	✓
✓	1234	19	19	✓