



Finding time complexity of algorithms

Problem 1: Finding Complexity using Counter Method

Question 1 | Correct Mark 1.00 out of 1.00 [Flag question](#)

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 %)

```
1 #include <stdio.h>
2 int main(){
3     int n;
4     scanf("%d",&n);
5     int i = 1;
6     int s = 1;
7     int c = 0;
8     while (s<=n){
9         c++;
10        i++;
11        c++;
12        s += i;
13        c++;
14    }
15    c++;
16    printf("%d",c+2);
17    return 0;
18 }
```

	Input	Expected	Got	
✓	9	12	12	✓
✓	4	9	9	✓

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

Problem 2: Finding Complexity using Counter method

Question 1 | Correct Mark 1.00 out of 1.00 [Flag question](#)

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 %)

```
1 #include<stdio.h>
2 void func(int n)
3 {
4     int c=0;
5     if(n==1)
6     {c++;
7       //printf("*");
8       c++;
9     }
10    else
11    {c++;
12      for(int i=1; i<=n; i++)
13      {c++;
14        for(int j=1; j<=n; j++)
15        {c++;
16          //printf("*");
17          c++;
18          // printf("*");
19          c++;
20          break;
21        }c++;
22      }c++;
23    }
24    printf("%d",c);
25  }
26 int main(){
27     int n;
28     scanf("%d",&n);
29     func(n);
30 }
```

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

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

Problem 3: Finding Complexity using Counter Method

Question 1 | Correct Mark 1.00 out of 1.00 [Flag question](#)

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:

```
1 #include <stdio.h>
2 void Factor(int num)
3 {
4     int c = 0,i;
5     for ( i = 1; i <= num;++i){
6         c++;
7
8
9         if (num % i== 0)
10        {
11            //printf("%d ", i);
12            c++;
13        }c++;
14    }c++;
15    printf("%d",c);
16 }
17
18 int main(){
19     int num;
20     scanf("%d",&num);
21     Factor(num);
22 }
```

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

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

Problem 4: Finding Complexity using Counter Method

Question 1 | Correct Mark 1.00 out of 1.00 [Flag question](#)

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:

```
1 #include <stdio.h>
2 void function(int n)
3 {
4     int c=0;
5     c++;
6     for(int i=n/2;i<n;i++){
7         c++;
8         for(int j=1;j<n;j=2*j){
9             c++;
10            for(int k=1;k<n;k=k*2){
11                c++;
12                c++;
13            }c++;
14        }c++;
15    }c++;
16    printf("%d",c);
17 }
18 int main(){
19     int n;
20     scanf("%d",&n);
21     function(n);
22 }
```

	Input	Expected	Got	
✓	4	30	30	✓
✓	10	212	212	✓

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

Problem 5: Finding Complexity using counter method

Question 1 | Correct Mark 1.00 out of 1.00 [Flag question](#)

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:

```
1 #include <stdio.h>
2 void reverse(int n)
3 {   int c=0;
4     int rev = 0, remainder;
5     c++;
6     while (n != 0)
7     {c++;
8         remainder = n % 10;
9         c++;
10        rev = rev * 10 + remainder;
11        c++;
12        n/= 10;
13        c++;
14    }
15    c++;
16    //printf(rev);
17    c++;
18    printf("%d",c);
19 }
20 int main(){
21     int n;
22     scanf("%d",&n);
23     reverse(n);
24 }
```

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

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.