Problem 5: Finding Complexity using counter method

Started on	Thursday, 21 August 2025, 9:45 PM
State	Finished
Completed on	Thursday, 21 August 2025, 9:51 PM
Time taken	6 mins 1 sec
Marks	1.00/1.00
Grade	**10.00** out of 10.00 (**100**%)

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() state

Input:

A positive Integer n

Output:

Print the value of the counter variable

Answer:

```c
#include <stdio.h>
#include <math.h>

int main() {
    int n;
    scanf("%d", &n);

    int d = (n == 0) ? 1 : (int)log10(n) + 1;
    int count = 4 * d + 3;

    printf("%d", count);
    return 0;
}
```

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

Passed all tests! ✔

Correct

Marks for this submission: 1.00/1.00.

Finish review

Back to Course

Problem 4: Finding Complexity using Counter Method

Started on	Thursday, 21 August 2025, 9:38 PM
State	Finished
Completed on	Thursday, 21 August 2025, 9:45 PM
Time taken	7 mins 31 secs
Marks	1.00/1.00
Grade	**10.00** out of 10.00 (**100%**)

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() state

Input:

A positive Integer n

Output:

Print the value of the counter variable

Answer:

```c
#include <stdio.h>

int main() {
    int n;
    scanf("%d", &n);

    int count = 0;

    for (int i = n/2; i < n; i++) {
        count++; // for i < n check (each iteration)
        for (int j = 1; j < n; j = 2 * j) {
            count++; // for j < n check (each iteration)
            for (int k = 1; k < n; k = k * 2) {
                count++; // for k < n check (each iteration)
                count++; // for c++ operation
            }
            count++; // for k = k * 2 operation (each j iteration)
        }
        count++; // for j = 2 * j operation (each i iteration)
    }
    count++; // for final i < n check (false)

    printf("%d", count+1);
    return 0;
}
```

	Input	Expected	Got	
✔	4	30	30	✔
✔	10	212	212	✔

Passed all tests! ✔

Correct

Marks for this submission: 1.00/1.00.

Finish review

Back to Course

Problem 3: Finding Complexity using Counter Method

Started on	Thursday, 21 August 2025, 9:14 PM
State	Finished
Completed on	Thursday, 21 August 2025, 9:30 PM
Time taken	15 mins 46 secs
Marks	1.00/1.00
Grade	**10.00** out of 10.00 (**100**%)

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() stat

Input:

A positive Integer n

Output:

Print the value of the counter variable

Answer:

```c
#include <stdio.h>

int main() {
    int num;
    scanf("%d", &num);

    int count = 0;

    for (int i = 1; i <= num; ++i) {
        count++;
        count++;

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

    printf("%d", count);
    return 0;
}
```

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

Passed all tests! ✔

Correct

Marks for this submission: 1.00/1.00.

Finish review

Back to Course

Problem 2: Finding Complexity using Counter method

Started on	Thursday, 21 August 2025, 9:00 PM
State	Finished
Completed on	Thursday, 21 August 2025, 9:14 PM
Time taken	14 mins 9 secs
Marks	1.00/1.00
Grade	**10.00** out of 10.00 (**100**%)

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 meth

```
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() state

Input:

A positive Integer n

Output:

Print the value of the counter variable

Answer: (penalty regime: 0 %)

```c
#include <stdio.h>

int main() {
    int n;
    scanf("%d", &n);

    if (n == 1) {
        printf("1");
    } else {
        printf("%d", 5 * n + 2);
    }

    return 0;
}
```

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

Passed all tests! ✔

Correct

Marks for this submission: 1.00/1.00.

Finish review

Back to Course

Problem 1: Finding Complexity using Counter Method

Started on	Thursday, 21 August 2025, 8:55 PM
State	Finished
Completed on	Thursday, 21 August 2025, 8:59 PM
Time taken	4 mins 1 sec
Marks	1.00/1.00
Grade	**10.00** out of 10.00 (**100**%)

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 meth

```
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() state

Input:

A positive Integer n

Output:

Print the value of the counter variable

For example:

Input	Result
9	12

Answer: (penalty regime: 0 %)

```c
#include <stdio.h>

void function(int n) {
    int i = 1;
    int s = 1;
    int count = 0;

    count++; // first condition check
    while (s <= n) {
        i++; count++;        // i++
        s += i; count++;     // s += i
        count++;             // next condition check
    }

    printf("%d\n", count+2);
}

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

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

Passed all tests! ✔

Correct

Marks for this submission: 1.00/1.00.

Finish review

Back to Course