

**Question 1** : Correct. Mark 1.00 out of 1.00. ⚡ Register now

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 <cs50.h>
2
3 void Find(int n) {
4     int i = 1;
5     int s = 1;
6     int counter = 0;
7
8     counter = counter + 2;
9
10    while (s < n) {
11        counter++;
12
13        i++;
14        counter++;
15        s = s + i;
16        counter--;
17    }
18}
```

9 12

Answer: (penalty regime: 0 %)

```
1 #include <stdio.h>
2
3 void find(int n) {
4     int i = 1;
5     int s = 1;
6     int counter = 0;
7
8     counter = counter + 2;
9
10    while (s < n) {
11        counter++;
12
13        i++;
14        counter++;
15        s = s + i;
16        counter++;
17    }
18
19    counter++;
20
21    printf("%d\n", counter);
22 }
23
24 int main() {
25     int n;
26     scanf("%d", &n);
27     Find(n);
28     return 0;
29 }
```

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

Started on: Friday, 8 August 2025, 9:10 PM

State: Finished

Completed on: Friday, 8 August 2025, 9:21 PM

Time taken: 11 mins.

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 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("\n");
                printf("\n");
                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
3 void func(int n) {
4     int counter = 0;
5
6     counter = counter + 1;
7
8     if (n == 10) {
9         }
10    else {
11        counter = counter - 3;
12
13        for (int i = 1; i <= n; i++) {
14            counter = counter + 5;
15        }
16    }
17    printf("%d\n", counter);
18 }
19
20 int main() {
21     int n;
22     scanf("%d", &n);
23     func(n);
24     return 0;
25 }
```

	Input	Expected	Got
✓	2	12	12 ✓
✓	1809	5862	5862 ✓
✓	143	717	717 ✓

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

Problem 3: Finding Complexity using Counter Method

Started on Friday, 8 August 2025, 9:22 PM

State Finished

Completed on Friday, 8 August 2025, 9:32 PM

Time taken 10 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.

```
Factorium() {
    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

Output:
Print the value of the counter variable

Answer:

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

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

Started on Friday, 18 August 2023, 9:34 PM

State Finished

Completed on Tuesday, 19 August 2023, 12:53 PM

Time taken 10 days 15 hours

Marks 1.00/1.00

Grade 10.00 out of 10.00 (100%)

Question 1 : Correct. Mark 1.00 out of 1.00 T Regrettably

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=3; 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:

A positive integer n
Output:
Print the value of the counter variable

Answer:

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

Input	Expected	Got
✓ 4	38	38 ✓
✓ 18	212	212 ✓

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00

Problem 5: Finding Complexity using counter method

Started on: Sunday, 17 August 2025, 8:11 PM

State: Finished

Completed on: Sunday, 17 August 2025, 8:21 PM

Time taken: 10 mins 52 secs

Marks: 1.00/1.00

Grade: 10.00 out of 10.00 (100%)

Question 1 Correct | Mark 1.00 out of 1.00 ⏺ Page version

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

Input:
A positive integer n .
Output:
Print the value of the counter variable

Answer:

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

Input	Expected	Get
✓ 12	11	11 ✓
✓ 1234	19	19 ✓

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00