

(1)Question 1

Correct

Mark 1.00 out of 1.00

[Flag question](#)

Question text

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:

```

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

#include <stdio.h>

int main() {

```
int n;  
int count = 0;  
  
scanf("%d", &n);  
  
int i = 1;  
count++;  
  
int s = 1;  
count++;  
  
while (s <= n) {  
    count++;  
  
    i++;  
    count++;  
  
    s += i;  
    count++;  
}  
count++;  
  
printf("%d\n", count);  
  
return 0;
```

}

(2) Question 2

Correct

Mark 1.00 out of 1.00

[Flag question](#)

Question text

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:

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

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

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

```
int counter = 0;

counter++;

if (n != 1)
{
    for (int i = 1; i <= n; i++)
    {
        counter++;
        counter++;
        counter++;
        counter++;
        counter++;
    }
    counter++;
}

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

(3) Question 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:

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

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

```
int main()
{
    int num, i;
    int counter = 0;

    scanf("%d", &num);
```

```

for (i = 1; i <= num; i++)
{
    counter++;
    counter++;

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

    counter++;

    printf("%d\n", counter);
}

return 0;
}

```

(4) Question 4

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 #include <math.h>
3
4 int main() {
5     int n;
6     scanf("%d", &n);
7
8     double a = 2.296;
9     double b = -1.813;
10    double c = 0.517;
11
12    int counter = round(a * n * n + b * n + c);
13
14    printf("%d", counter);
15    return 0;
16 }
17

```

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

```

#include <math.h>

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

    double a = 2.296;
    double b = -1.813;
    double c = 0.517;

    int counter = round(a * n * n + b * n + c);

    printf("%d", counter);
    return 0;
}

```

(5) Question 5

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
3 void reverse(int n)
4 {
5     int rev = 0, remainder;
6     int count = 1;
7
8     while (n != 0)
9     {
10         count++;
11         remainder = n % 10;
12         count++;
13         rev = rev * 10 + remainder;
14         count++;
15         n /= 10;
16         count++;
17     }
18     count++;
19     count++;
20     printf("%d\n", count);
21 }
22
23 int main()
24 {
25     int n;
26     scanf("%d", &n);
27     reverse(n);
28     return 0;
29 }
```

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

```
void reverse(int n)
{
    int rev = 0, remainder;
    int count = 1;
```

```
while (n != 0)
```

```
{
    count++;
```

```
    remainder = n % 10;  
    count++;  
    rev = rev * 10 + remainder;  
    count++;  
    n /= 10;  
    count++;  
}  
  
count++;  
count++;  
printf("%d\n", count);  
}
```

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