

230701016

CSE-A

M. AISHWARYA

1.

AIM-

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.

CODE-

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

INPUT-

A positive Integer n

OUTPUT-

Print the value of the counter variable

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

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

2.

AIM-

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.

CODE-

```
1  #include<stdio.h>
2  void func(int n)
3  {
4      int c=0;
5      c++;
6      if(n==1)
7      {
8          c++;
9          //printf("*");
10     }
11
12     else
13     {
14         for(int i=1; i<=n; i++)
15         {
16             c++;
17             for(int j=1; j<=n; j++)
18             {
19                 c++;
20                 //printf("*");
21                 c++;
22                 //printf("*");
23                 c++;
24                 break;
25             }
26             c++;
27         }
28         c++;
29     }
30     c++;
31     printf("%d",c);
32 }
33
34 int main()
35 {
36     int n;
37     scanf("%d",&n);
38     func(n);
39 }
40
```

INPUT-

A positive Integer n

OUTPUT-

Print the value of the counter variable

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

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

3.

AIM-

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.

CODE-

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

INPUT-

A positive Integer n

OUTPUT-

Print the value of the counter variable

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

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

4.

AIM-

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.

CODE-

```
1  #include<stdio.h>
2  void function(int n)
3  {
4      int ct=0;
5      int c= 0;
6      ct++;
7      for(int i=n/2; i<n; i++)
8      {
9          ct++;
10         for(int j=1; j<n; j = 2 * j)
11         {
12             ct++;
13             for(int k=1; k<n; k = k * 2)
14             {
15                 ct++;
16                 c++;
17                 ct++;
18             }
19             ct++;
20         }
21         ct++;
22     }
23     ct++;
24     printf("%d",ct);
25 }
26
27
28 int main()
29 {
30     int n;
31     scanf("%d",&n);
32     function(n);
33 }
```

INPUT-

A positive Integer n

OUTPUT-

Print the value of the counter variable

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

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

5.

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.

CODE-

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

INPUT-

A positive Integer n

OUTPUT-

Print the value of the counter variable

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

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.