WEEK – 2

1.

Consider a sequence of the form 0, 1, 1, 2, 4, 7, 13, 24, 44, 81, 149...

Write a method program which takes as parameter an integer n and prints the nth term of the above sequence. The nth term will fit in an integer value.

Example Input:

5

Output:

4

Example Input:

8

Output:

24

```
import java.util.*;
public class Sequence
{
   public static void main(String args[])
   {
      Scanner scan=new Scanner(System.in);
      int n=scan.nextInt();
      int n1=0,n2=1,n3=1;
      for(int i=3;i<n;i++)
      {
         int temp=n1+n2+n3;
         n1=n2;
         n2=n3;
         n3=temp;
      }
      System.out.print(n3);
   }
}
```

	Input	Expected	Got	
✓	5	4	4	✓
✓	8	24	24	✓
✓	11	149	149	✓

2.

Consider the following sequence:

1st term: 1

2nd term: 1 2 1

3rd term: 1 2 1 3 1 2 1

4th term: 1 2 1 3 1 2 1 4 1 2 1 3 1 2 1

And so on. Write a program that takes as parameter an integer n and prints the nth terms of this sequence

Example Input:

1

Output:

1

Example Input:

4

Output:

1 2 1 3 1 2 1 4 1 2 1 3 1 2 1

```
import java.util.*;
public class Pattern
{
   public static void main(String args[])
   {
      Scanner scan=new Scanner(System.in);
      int n=scan.nextInt();
      String result="1";
      for(int i=1;i<n;i++)
      {
         result+=" "+(i+1)+" "+result;
      }
      System.out.print(result);
   }
}
```

	Input	Expected	Got	
✓	1	1	1	✓
✓	2	1 2 1	1 2 1	✓
✓	3	1 2 1 3 1 2 1	1 2 1 3 1 2 1	✓
✓	4	1 2 1 3 1 2 1 4 1 2 1 3 1 2 1	1 2 1 3 1 2 1 4 1 2 1 3 1 2 1	✓

3.

Write a program that takes as parameter an integer n.

You have to print the number of zeros at the end of the factorial of n.

For example, 3! = 6. The number of zeros are 0. 5! = 120. The number of zeros at the end are 1.

Note: n! < 10^5

Example Input:

3

Output:

0

Example Input:

60

Output:

14

```
// Java program to count trailing 0s in n!
import java.io.*;
import java.util.Scanner;
class prog {
   // Function to return trailing
   // 0s in factorial of n
   static int findTrailingZeros(int n)
   {
      if (n < 0) // Negative Number Edge Case
         return -1;

      // Initialize result
      int count=0;

      // Keep dividing n by powers
      // of 5 and update count
      for (int i = 5; n / i >= 1;i*=5)
         count += n / i;

      return count;
```

```
    }

    // Driver Code
    public static void main(String[] args)
    {
        int n ;
        Scanner sc= new Scanner(System.in);
        n=sc.nextInt();
        System.out.println(findTrailingZeros(n));
    }
}
```

	Input	Expected	Got	
✓	3	0	0	✓
✓	60	14	14	✓
✓	100	24	24	✓
✓	1024	253	253	✓