1. PRINT 1 TO N W.O USING LOOPS

```
class PrintNumbers{

    public static void printNumbers(int start, int end) {
        if (start <= end) {
            System.out.println(start);
            printNumbers(start + 1, end);
        }
    }
    public static void main(String[] args) {
        int i=1;
        int n=10;
        printNumbers(i,n);
    }
}
```

2. SUM OF NATURAL NUMBERS

```
import java.util.Scanner;
public class nNatural {
    static int sum(int n ){
        if (n>0)
            return n+sum(n-1);
        else
            return 0;
        }
        public static void main(String[] args ){
        Scanner  sc =new Scanner(System.in);
        int result = sum(sc.nextInt());
            System.out.print(result);
        }

}
```

3. MEAN OF ARRAY USING RECURSION

```
class MeanOfArray{

    public static int arraySum(int[] arr, int i) {
        if (i == arr.length) {
            return 0;
        }
        return arr[i] + arraySum(arr, i + 1);
    }
    public static double arrayAverage(int[] arr) {
        return arraySum(arr, 0) / (double) arr.length;
    }

    public static void main(String[] args) {
        int arr[]={1,5,7,8,6};
        int i=0;
        System.out.println(arrayAverage(arr));

    }
}
```

4. DECIMAL TO BINARY CONVERSION

```
import java.util.*;
public class Decimal{
        public int Deci(int a, String s){
                if(a==0)
                        return Integer.parseInt(s);
                s = Integer.toString(a%2)+s;
                        return Deci(a/2,s);
        }
        public static void main(String args[]){
                Decimal deci = new Decimal();
                Scanner sc = new Scanner(System.in);
                System.out.print("Enter The Value : ");
                int n = sc.nextInt();
                String blank = "";
                System.out.println(deci.Deci(n,blank));
        }
}
```

5. SUM OF DIGITS OF A NUMBER USING RECURSION

```
public class Sum{
        public static void main(String args[]){
        int number = 21334;
        int sum = sumDigits(number);
        System.out.println("sum: " + sum);
        }
```

```
    public static int sumDigits(int num){
        if(num==0)
            return 0;

        else
        {
            return (num%10) + sumDigits(num/10);
        }
    }
}
```

6. PRINT REVERSE OF A STRING USING RECURSION

```
public class ReverseStr {

    public static void solve(char[]s,int i,int n){
        if (i < n / 2) {
            char temp = s[i];
            s[i] = s[n - i - 1];
            s[n - i - 1] = temp;
            solve(s, i + 1, n);
        }
    }
    public static void main(String[] args) {
        String str="Shubham";
         char[] s1 = str.toCharArray();
         int n=s1.length;
        solve(s1, 0, n);
        System.out.println("Reversed string: " + String.valueOf(s1));
    }

}
```

7. PROGRAM FOR LENGTH OF A STRING USING RECURSION

```
public class Length {
 public static int findLength(String s) {
  if (s.equals("")) {
   return 0;
  } else {
   return 1 + findLength(s.substring(1));
  }
 }

 public static void main(String[] args) {
  String string = "My Name Is Shubham.";
  int length = findLength(string);
  System.out.println("Length of the string: " + length);

 }

}
```

8. TAIL RECURSION TO CALCULATE SUM OF ARRAY OF ELEMENTS

```
    public class TailRec{
    public static int SumArr(int[] arr, int index, int sum){
        if(index == arr.length){
            return sum;
        }

        sum += arr[index];
        System.out.println(sum);
        return SumArr(arr, index+1, sum);
    }

    public static void main(String[] args){
        int arr[] = {10,20,30,40,50};
        int sum = 0;
        sum = SumArr(arr, 0 , 0);
        System.out.println("Sum of arrays is: "+sum);
    }
}
```

9. RECURSIVE FUNCTION TO CHECK IF A STRING IS PALINDROME

```
import java.util.Scanner;

public class PalindromeString {
    public static boolean isPalindrome(String S, int start, int end){
        if(start > end){
            return true;
        }
        if(S.charAt(start) != S.charAt(end)){
            return false;
        }
        return isPalindrome(S, ++start, --end);
    }
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);

        String str = sc.nextLine();
        System.out.println(isPalindrome(str, 0, str.length()-1));

    }
}
```

10. PRINT FIBONACCI SERIES IN REVERSE ORDER USING RECURSION

```
public class reverseFibonacchi {
            public static int fibonacci(int n) {
        if (n <= 1)
            return n;
        return fibonacci(n - 1) + fibonacci(n - 2);
    }
            public static void printReverseFibonacci(int n) {
        for (int i = n - 1; i >= 0; i--) {
            System.out.print(fibonacci(i) + " ");
        }
    }

            public static void main(String[] args) {
                        int n = 10;
                    System.out.println("Fibonacci Series in reverse order Is :");
                    printReverseFibonacci(n);
            }

}
```