1) Given that calculator can perform only base 10 logarithms

  hence log base 2 of 2048= [(log base 10 of 2048)/(log base 10 of 2)]

---

2) 3 + 5 + 7 + 9 + ... + 2k+1= (2*1+1) + (2*2+1) + (2*3+1) + (2*4+1) +.....+(2*k+1)

=> 2*(1 + 2 + 3 +.....+ k) + (1 + 1 + 1.......+ 1 ktimes)
=> 2*k(k+1)/2 + k
=> k(k+1)+k
=> k(k+2) or 2*[k(k+1)/2] + k

---

3) Given statement is : n^3 > 2^n for any n.

let us consider n=1
then the statement goes false i.e 1^3 > 2^1 => 1 > 2 (False)

Hence the given statement "n^3 > 2^n for any n" is false.

---

4) Given statement is : The square of an even number is also even.

let us assume the above statement to be false. i.e The square of an even number is odd.
if x^2 is odd then x*x = 2c+1 for some c.
and x=2*a then x^2 = (2*a)*(2*a) = 4* (a^2) = 2*(2*(a^2)) => Even number
=> Hence the above step contradicts our assumption.
=> Hence the square of an even number is also an even number.
=> Hence the statement given is proved.

---

5)a) Given

$$\sum_{i=1}^{n} i^3 = [n^2][(n+1)^2]/4$$

let us consider the base case for n=1 then the equation holds good as

$$\sum_{i=1}^{1} i^3 = [1^2][(1+1)^2]/4 \Rightarrow 1^3 = [1][2^2]/4 \Rightarrow 1=4/4$$

1=1
Hence L.H.S=R.H.S for base case

Let us consider the given equation is true for n=k

k
sum i^3 = [k^2][(k+1)^2]/4 is true from our assumption. -------(1)
i=1

Let us know consider the next case of n=k+1

k+1
sum i^3 = 1^3 + 2^3 + 3^3 +......+ k^3 + (k+1)^3
i=1 ----------

|
|
V

k+1
sum i^3 = {[k^2][(k+1)^2]/4} + (k+1)^3 -------(2)
i=1

Taking (k+1)^2 as common from equation (2)

=>((k+1)^2)[(k^2)/4 + (k+1)]
=>((k+1)^2)[(k^2 + 4k + 4)/4]
=>((k+1)^2)[((k+2)^2)/4] since ((k+2)^2)= k^2 + 4k +4)
=>((k+1)^2)[(((k+1)+1)^2)/4] ------(3)
let us assume k+1=p and substitute in equation ---(3)
=>[p^2][(p+1)^2]/4 ------(4)

From the above equation --(4) The given statement is true for p=k+1.

Hence from the Induction principle , we can say that the given statement n

sum i^3 = [n^2][(n+1)^2]/4
i=1

is true for all Natural numbers.

Hence the given statement is proved.

---

5)b) Given statement is: n^2-n is even for any n>=1

let us consider the base case of n=1 then the given statement will turn as follows

1^2-1=0
since 0 is an even number

Hence the given statement is true for base case n=1

let us assume the statement is true for n=k
The given statement would turn as k^2-k = 2*c --------(i)

Let us know consider the next case of n=k+1
(k+1)^2-(k+1) = k^2 + 2*k + 1 - k -1 = k^2 + k --------(ii)

From equation (i) the above equation (ii) will turn as

k^2 + k = k^2 - k + 2*k = 2*c + 2*k =2*(c + k) => Even number

Hence from the Induction principle , we can say that the given statement "n^2 - n is even for any n >= 1" is true for all natural numbers.

Hence the given statement is proved.

---

6)a)

Using 1 parameter

```
class RecursionPrint{
   public static int flag=0;
   private static int i;
   public static void setI(int n){
   i=n;
   }
   public static void print(int n){
   if(n>0 && flag==0){
   System.out.print(n+" ");
   n=n-1;
   print(n);
   }
   else if(n==0 && flag==0){
   System.out.print(n+" ");
   flag=1;
   print(n+1);
   }
   else if(n<=i && flag==1){
   System.out.print(n+" ");
   print(n+1);
   }
   else{
   flag=2;
   }

   }
   public static void main(String[] args){
   java.util.Scanner s= new java.util.Scanner(System.in);
   System.out.println("Enter any value to display the series:");
   int n=s.nextInt();
   setI(n);
   print(n);
   }
}
```

Using 3 parameters

```
/*import java.util.Scanner;
class DisplayRecursion{


       public static void recursion(int n,int flag,int k){
             if(flag==0 && n>0){
             System.out.print(n);
```

```
            recursion(n-1,flag,k);
            }
            else if(n==0)
            {
            flag=1;
            System.out.print(n);
            recursion(n+1,flag,k);
            }
            else if(flag==1 && n<=k){
            System.out.print(n);
            recursion(n+1,flag,k);
            }

         }


        public static void main(String[] aegs){
        Scanner s= new Scanner(System.in);
        System.out.println("Enter any value to display:");
        int n=s.nextInt();
        recursion(n,0,n);
        }
}

*/
```

---

6)b)

```
import java.util.Scanner;
class Recursion{
   public int function(int[] a,int n){

  int count=0;
   if(n==0)
   return count;
   else{
     count=function(a,n-1);
      if(a[n-1]%2!=0)
      count++;

   }
   return count;
   }

}

class Test{
   public static void main(String[] args){
      Scanner s= new Scanner(System.in);
      System.out.println("Enter the size of the array:");
      int n=s.nextInt();
      System.out.println("Enter some Integer values in the array:");
      int[] a=new int[n];
         for(int i=0;i<n;i++){
            a[i]=s.nextInt();
         }
      Recursion r= new Recursion();
      int count=r.function(a,n);
```

```
      System.out.println(" There are "+ count +" no. of odd integers in
the array");
   }
}
```

---

7)

```
import java.util.Scanner;
public class Pair <T>{
   private T s1,s2;

   public void sets1(T s1){
   this.s1=s1;
   }
   public T gets1(){
   return s1;
   }
   public void sets2(T s2){
   this.s2=s2;
   }
   public T gets2(){
   return s2;
   }

      public static void main(String[] args){
         Pair<String> p= new Pair<String>();
        /* p.sets1("Keerthi");
         p.sets2("Teja");*/
         Scanner s= new Scanner(System.in);
         System.out.println("Enter any string value for s1:");
         String s1=s.nextLine();
         System.out.println("Enter any string value for s2:");
         String s2=s.nextLine();
         p.sets1(s1);
         p.sets2(s2);
         System.out.println(p.gets1());
         System.out.println(p.gets2());
      }
}
```
