Recursion in java

Q1 : Given an integer, find out the sum of its digits using recursion.

```
Ans import java.io.*;

import java.util.*;


public class Recurrsion_1
{   static int func(int n)
{
    if (n == 0)
    return 0;
    return (n % 10 + func(n / 10));
}
    public static void main(String[] args) {
        System.out.println("Enter the number");
        Scanner sc = new Scanner(System.in);
        int n=sc.nextInt();
        int ans = func(n);
        System.out.println(" Sum of digits in " + n
       + " is " + ans );
    }
}
```

Q2: Given a number n. Find the sum of natural numbers till n but with alternate signs.

That means if n = 5 then you have to return 1-2+3-4+5 = 3 as your answer.

```
Ans:

import java.io.*;
import java.util.*;

public class Recurrsion_2
{   static int go(int n  , int i){
    if(i == n + 1)return 0;
    if(i%2 == 0)return go(n , i + 1) - i;
    else return go(n , i + 1) + i;
}
    public static void main(String[] args) {

        Scanner sc = new Scanner(System.in);
        int n = sc.nextInt();
        System.out.println(go(n,1));


    }
```

```
}
```

Q3: Print the max value of the array [13, 1, -3, 22, 5].

Ans :
```
// Recursive java program to find maximum element in an array

// function to print maximum element using recursion
import java.io.*;
import java.util.*;
public class Recurrsion_3{
    public static int maxValue(int arr[], int n, int idx){
        if(idx == n - 1)
        return arr[n - 1];
        int maxOfLaterIndices = maxValue(arr, n,idx + 1);
        int maxVal = Math.max(arr[idx], maxOfLaterIndices);
        return maxVal;
    }
    public static void main(String[] args) {
        int[] arr={13,1,-3,22,5};
        int n=arr.length;
        System.out.println("The max value is : " + maxValue(arr,n,0));
    }
}
```

Q4 : Find the sum of the values of the array [92, 23, 15, -20, 10].

Ans
```
import java.util.*;

import java.util.Scanner;
public class Recurrsion_4{
    public static int totalSum(int []arr , int idx){
        if(idx == 0)return arr[idx];

        return arr[idx] + totalSum(arr , idx - 1);
    }
    public static void main(String[] args){
        int arr[] = {92,23, 15, -20, 10};
        int n = arr.length;

        System.out.println("The sum of all elements of the array is : " +
totalSum(arr , n - 1));
    }
}
```

Q5. Given a number n. Print if it is an armstrong number or not. An armstrong number is a number if the sum

of every digit in that number raised to the power of total digits in that number is equal to the number.

Example : 153 = 1^3 + 5^3 + 3^3 = 1 + 125 + 27 = 153 hence 153 is an armstrong number. (Easy)

Ans

```java
import java.util.*;
import java.util.Scanner;
public class Recurrsion_5{
        public static void main(String[] args){
                Scanner scn = new Scanner(System.in);
                System.out.println("Enter the number n: ");
                int n = scn.nextInt();
                int digits = 0;
                int temp = n;
                while(temp > 0){
                        digits++;
                        temp/=10;
                }
                if(n == isArmstrong(n , digits))
                        System.out.println("yes");
                else
                        System.out.println("no");
        }
        public static int power(int a, int b){
                if(b == 0)return 1;
                if(b%2 == 0)return power(a , b/2) * power(a , b/2);
                return a * power(a  , b/2) * power(a , b/2);
        }
        public static int isArmstrong(int n , int digits){
                if(n == 0)return 0;
                return power(n%10 , digits) + isArmstrong(n/10 , digits);
        }
}
```