ASSIGNMENT- (RECURSION)

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

Soln.

```
import java.util.Scanner;

public class Recursion_Assg1 {

    static int sumOfDigit(int n) {
        int rem;
        if (n >= 0 && n < 9) {
            return n;
        }
        rem = n % 10;
        // result += n;

        return rem + sumOfDigit(n / 10);
    }

    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter the number: ");
        int n = sc.nextInt();
        int result = sumOfDigit(n);
        System.out.println("Sum of given digits are: " + result);
    }
}
```

\

Q2. 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.

Soln.

```
import java.util.Scanner;

public class Recursion_Assg2 {
    static int sonn(int n) {
        if (n == 1)
            return 1;
        if (n % 2 == 0) {
            return sonn(n - 1) - n;
        } else {
            return sonn(n - 1) + n;
        }
    }

    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter the value of n: ");
        int n = sc.nextInt();
        int result = sonn(n);
        System.out.println("Sum of natural numbers till " + n + "
is: " + result);
    }
}
```

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

Soln.

```
public class Recursion_Assg3 {

    static int maxVal(int[] arr, int idx) {
        if (arr.length - 1 == idx) {
            return arr[idx];
        }
        int smallAns = maxVal(arr, idx + 1);
        return Math.max(arr[idx], smallAns);

    }

    public static void main(String[] args) {
        int[] arr = { 13, 1, -3, 22, 5 };
        System.out.println("Max value in the given array is: " +
maxVal(arr, 0));
    }
}
```

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

Soln.

```
public class Recursion_Assg4 {

    static int sum(int[] arr, int idx) {
        if (idx == arr.length - 1)
            return arr[idx];

        return sum(arr, idx + 1) + arr[idx];
    }

    public static void main(String[] args) {
        int[] arr = { 92, 23, 15, -20, 10 };
        System.out.println("Sum of all the values in the array is: "
+ sum(arr, 0));
    }
}
```

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.

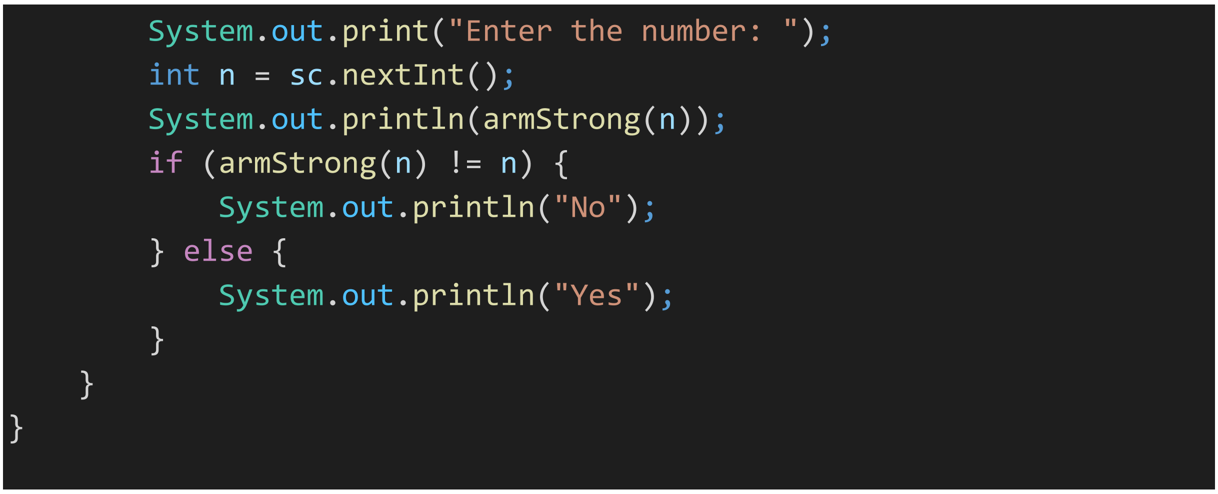
Soln.

```
import java.util.Scanner;

public class Recursion_Assg5 {

    static int armStrong(int n) {
        int rem, result = 0;
        if (n >= 0 && n < 9)
            return n;
        rem = n % 10;
        return (rem * 3) + armStrong(n / 10);
    }

    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
```



```
        System.out.print("Enter the number: ");
        int n = sc.nextInt();
        System.out.println(armStrong(n));
        if (armStrong(n) != n) {
            System.out.println("No");
        } else {
            System.out.println("Yes");
        }
    }
}
```