Recursion in java
Assignment java

Q1 : Given an integer, find out the sum of its digits using recursion.
Input: n= 1234
Output: 10
Explanation: 1+2+3+4=10
Ans:

```
public class Main {
    public static void main(String[] args) {
        int n = 1234;
        System.out.println("Sum of digits: " + sumOfDigits(n));
    }

    public static int sumOfDigits(int n) {
        // Base case: if n is a single digit
        if (n < 10) {
            return n;
        }

        // Recursive case: sum last digit and remaining digits
        return n % 10 + sumOfDigits(n / 10);
    }
}
```

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.

Constraints : 0<=n<=1e6
Input1 : n = 10

Output 1 : -5
Explanation : 1-2+3-4+5-6+7-8+9-10 = -5
Input 2 : n = 5
Output 2 : 3

Ans:

```
public class Main {
    public static void main(String[] args) {
        int n = 10;
        System.out.println("Sum with alternate signs: " + sumWithAlternateSigns(n));

        n = 5;
        System.out.println("Sum with alternate signs: " + sumWithAlternateSigns(n));
    }

    public static int sumWithAlternateSigns(int n) {
        int sum = 0;
        int sign = 1; // 1 for positive, -1 for negative
        for (int i = 1; i <= n; i++) {
            sum += sign * i;
            sign *= -1; // Alternate sign
        }
        return sum;
    }
}
```

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

```
public class Main {
    public static void main(String[] args) {
        int[] array = {13, 1, -3, 22, 5};
        System.out.println("Maximum value: " + findMax(array));
    }

    public static int findMax(int[] array) {
        int max = array[0];
        for (int i = 1; i < array.length; i++) {
            if (array[i] > max) {
                max = array[i];
            }
        }
        return max;
    }
}
```

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

```
public class Main {
    public static void main(String[] args) {
        int[] array = {92, 23, 15, -20, 10};
        System.out.println("Sum: " + findSum(array));
    }

    public static int findSum(int[] array) {
        int sum = 0;
        for (int value : array) {
            sum += value;
        }
        return sum;
    }
}
```

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)

Input1 : 153

Output1 : Yes

Input 2 : 134

Output2 : No

Ans:

```
public class Main {
   public static void main(String[] args) {
      System.out.println(isArmstrong(153) ? "Yes" : "No"); // Yes
      System.out.println(isArmstrong(134) ? "Yes" : "No"); // No
   }

   public static boolean isArmstrong(int n) {
      int original = n;
      int sum = 0;
      int digitCount = countDigits(n);

      while (n > 0) {
         int digit = n % 10;
         sum += (int) Math.pow(digit, digitCount);
         n /= 10;
      }

      return sum == original;
   }

   public static int countDigits(int n) {
      int count = 0;
      while (n > 0) {
         n /= 10;
         count++;
```

```
        }
        return count;
    }
}
```