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

Ans.

Code:

```
// Java program to find the sum of digits in a number using the recursion

import java.util.*;

class Sum_Recursion_PW
{
	public static int sumOfDigits(int n)
	{
		if(n == 0)
			return 0;
		else
			return n%10 + sumOfDigits(n/10);
	}


	public static void main(String arg[])
	{
		Scanner s = new Scanner(System.in);
		System.out.print("Enter the Number: ");
		int num = s.nextInt();

		int result = sumOfDigits(num);
      System.out.println("Sum of digits: " + result);

	}
}
```

Output:

```
PS V:\java-prog_notepad++> javac Sum_Recursion_PW.java
PS V:\java-prog_notepad++> java Sum_Recursion_PW
Enter the Number: 1234
Sum of digits: 10
PS V:\java-prog_notepad++>
```

2. **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.

Code:

```
import java.util.*;

public class AlternateSum_PW {

   // Method to find the sum of natural numbers with alternate signs
   public static int alternateSum_PW(int n)
         {
      int sum = 0;

      for (int i = 1; i <= n; i++) {
         if (i % 2 == 0) {
            sum -= i; // Subtract the number if it's even
         } else {
            sum += i; // Add the number if it's odd
         }
      }

      return sum;
   }

   public static void main(String[] args)
         {
                  Scanner s = new Scanner(System.in);
                  System.out.println("Enter the number: ");
      int n = s.nextInt();

      int result = alternateSum_PW(n);

      System.out.println("Alternate sum for n = " + n + ": " + result);
   }
}
```

Output:

```
PS V:\java-prog_notepad++> javac AlternateSum_PW.java
PS V:\java-prog_notepad++> java AlternateSum_PW
```

```
Enter the number:
5
Alternate sum for n = 5: 3
PS V:\java-prog_notepad++>
```

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

Ans.

Code:

```
import java.util.*;

public class MaxValueArray_PW {

   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;
   }

   public static void main(String[] args) {
      int[] array = {13, 1, -3, 22, 5};
      int maxValue = findMax(array);
      System.out.println("The maximum value in the array is: " + maxValue);
   }
}
```

Output:

```
PS V:\java-prog_notepad++> javac MaxValueArray_PW.java
PS V:\java-prog_notepad++> java MaxValueArray_PW
The maximum value in the array is: 22
PS V:\java-prog_notepad++>
```

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

Ans.

Code:

```
import java.util.*;

public class SumArray_PW
{

   public static int findSum(int[] array)
         {
      int sum = 0;

      for (int value : array)
         sum += value;

      return sum;
   }

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

Output:

```
PS V:\java-prog_notepad++> javac SumArray_PW.java
PS V:\java-prog_notepad++> java SumArray_PW
The sum of the values in the array is: 120
PS V:\java-prog_notepad++>
```

5. **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.**

Ans.

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

Code:

```
import java.util.*;

public class ArmstrongNumber_PW
{
   public static boolean isArmstrong(int n)
         {
      int originalNumber = n;
      int sum = 0;
      int numberOfDigits = String.valueOf(n).length();

      while (n != 0)
                {
         int digit = n % 10;
         sum += Math.pow(digit, numberOfDigits);
         n /= 10;
      }
      return sum == originalNumber;
   }

   public static void main(String[] args)
         {
      int input = 153;

      if (isArmstrong(input))
         System.out.println(input + " is an Armstrong number: Yes");
                else
         System.out.println(input + " is an Armstrong number: No");
   }
}
```

Output:

```
PS V:\java-prog_notepad++> javac ArmstrongNumber_PW.java
PS V:\java-prog_notepad++> java ArmstrongNumber_PW
153 is an Armstrong number: Yes
PS V:\java-prog_notepad++>
```