1) Create a class EvenOdd with main method in it to find whether the number is even or odd. Note :

```
package q29581;
import java.util.*;
public class EvenOdd{
public static void main(String[] args)
{
if(args.length==0)
{
System.out.println("Noinput provided");
}
else{
try{
int n=Integer.parseInt(args[0]);
if(n<0)
{
System.out.println("Error : Invalid Input");
}
else{
if((n & 1)==0)
{
System.out.println("Even");
}
else{
System.out.println("Odd");
}
}
}
catch(NumberFormatException e){
System.out.println("Invalid Input");
}
}
}
}
```

2. Write a class NumberPalindrome with a public method isNumberPalindrome that takes one parameter number of type int. Write a code to check whether the given number is palindrome or not

```
package oops;

public class Boxdemo {

   public static void main(String[] args) {

      Boxdemo boxDemo = new Boxdemo(); // Create an instance of the Boxdemo class

      boxDemo.isNumberPalindrome(12321); // Call the isNumberPalindrome method

   }

 public void isNumberPalindrome(int number) {

      // Write your code here
```

```
    int copy = number;
    int reverse = 0;
    int remainder;
    while (number > 0) {
      remainder = number % 10;
      reverse = (reverse * 10) + remainder;
      number /= 10;
    }
    if (copy == reverse) {
      System.out.println(copy + " is a palindrome");
    } else {
      System.out.println(copy + " is not a palindrome");
    }
  }
}
```

3. Create a class called BitwiseXOR with main method in it to perform BitwiseXOR operation by taking two input numbers

```
package oops;
public class Boxdemo {
    public static void main(String[] args) {
        // Check if two arguments are provided
        if (args.length != 2) {
            System.out.println("Please provide exactly two integers as command-line arguments.");
            return;
        }

        try {

            int A = Integer.parseInt(args[0]);
            int C = Integer.parseInt(args[1]);
```

```
            // Perform XOR operation
            int B = A ^ C;

            // Print the result
            System.out.println("XOR Result = " + B);
        } catch (NumberFormatException e) {
            // Handle the case where arguments cannot be
parsed as integers
            System.out.println("Invalid input: Please provide
valid integers as command-line arguments.");
        }
    }
}
```

4. Write a Java program to find minimum and maximum numbers in a given array

```
import java.util.*;

public class MinMaxArray
{
public static void main(String[] args)
{
Scanner scanner=new Scanner(System.in);
System.out.print("Enter number of elements: ");
int s=scanner.nextInt();
int [] a=new int[s];
System.out.print("Enter array elements: ");
for(int i=0;i< s;i++)
{
a[i]=scanner.nextInt();
}
int min=a[0];int max=a[0];
for(int i=1;i<a.length;i++)
{
if(a[i]<min)
{
min=a[i];
}
else if(a[i]>max)
{
max=a[i];
}
```

```
}
System.out.println("Mimimum element in array is: "+min);
System.out.println("Maximum element in array is: "+max);
}
}
```

5. Write a Java program to sort a list of names in ascending order.

```
import java.util.*;

import java.util.Arrays;
public class SortNames{
public static void main(String[] args){
int i;
Scanner sc=new Scanner(System.in);
System.out.print("Enter the number of names you want to enter:
");
int n=sc.nextInt();
sc.nextLine();
List<String> names=new ArrayList<>();
System.out.print("Enter the names: ");
for(i=0;i<n;i++)
{
String name=sc.nextLine();
names.add(name);
}
Collections.sort(names);
System.out.print("Sorted names:");
for(String name:names)
System.out.print(name+" ");
}
}
```