```java
import java.util.Arrays;
import java.util.HashSet;
import java.util.Set;

public class ArraySameElements {

  public static void main(String[] args) {
    Integer[] a1 = {1,2,3,2,1};
    Integer[] a2 = {1,2,3};
    Integer[] a3 = {1,2,3,4};

    System.out.println(sameElements(a1, a2));
    System.out.println(sameElements(a1, a3));
  }
  static boolean sameElements(Object[] array1, Object[] array2) {
    Set<Object> uniqueElements1 = new HashSet<>(Arrays.asList(array1));
    Set<Object> uniqueElements2 = new HashSet<>(Arrays.asList(array2));
    if (uniqueElements1.size() != uniqueElements2.size()) return false;

    for (Object obj : uniqueElements1) {
      if (!uniqueElements2.contains(obj)) return false;
    }

    return true;
  }

}
```

STDIN

Output:

```
true
false
```

```java
public class StringPrograms {

  public static void main(String[] args) {
    String str = "123";

    System.out.println(reverse(str));
  }

  public static String reverse(String in) {
    if (in == null)
      throw new IllegalArgumentException("Null is not valid input");

    StringBuilder out = new StringBuilder();

    char[] chars = in.toCharArray();

    for (int i = chars.length - 1; i >= 0; i--)
      out.append(chars[i]);

    return out.toString();
  }

}
```

STDIN

Output:

```
321
```

```java
public class SwapNumbers {

public static void main(String[] args) {
  int a = 10;
  int b = 20;

    System.out.println("a is " + a + " and b is " + b);

  a = a + b;
  b = a - b;
  a = a - b;

    System.out.println("After swapping, a is " + a + " and b is " + b);
    }

}
```

STDIN

Output:

```
a is 10 and b is 20
After swapping, a is 20 and b is 10
```

```java
import java.util.*;

public class Main {
    public static void main(String[] args) {
      System.out.println("Hello, mritsha!");
  }
}
```

```java
public class StringContainsVowels {

  public static void main(String[] args) {
    System.out.println(stringContainsVowels("mritsha")); // true
    System.out.println(stringContainsVowels("murali")); // false
  }

  public static boolean stringContainsVowels(String input) {
    return input.toLowerCase().matches(".*[aeiou].*");
  }

}
```

STDIN

Output:

```
true
true
```

```java
public class PrimeNumberCheck {

  public static void main(String[] args) {
    System.out.println(isPrime(19)); // true
    System.out.println(isPrime(49)); // false
  }

  public static boolean isPrime(int n) {
    if (n == 0 || n == 1) {
      return false;
    }
    if (n == 2) {
      return true;
    }
    for (int i = 2; i <= n / 2; i++) {
      if (n % i == 0) {
        return false;
      }
    }

    return true;
  }

}
```

STDIN

Output:

```
true
false
```

```java
public class PrintFibonacci {

  public static void printFibonacciSequence(int count) {
    int a = 0;
    int b = 1;
    int c = 1;

    for (int i = 1; i <= count; i++) {
      System.out.print(a + ", ");

            a = b;
      b = c;
      c = a + b;
    }
  }

  public static void main(String[] args) {
      printFibonacciSequence(10);
  }

}
```

STDIN

Output:

```
0, 1, 1, 2, 3, 5, 8, 13, 21, 34,
```

```java
import java.util.Arrays;
import java.util.HashSet;
import java.util.Set;

public class ArraySameElements {

  public static void main(String[] args) {
    Integer[] a1 = {1,2,3,2,1};
    Integer[] a2 = {1,2,3};
    Integer[] a3 = {1,2,3,4};

    System.out.println(sameElements(a1, a2));
    System.out.println(sameElements(a1, a3));
  }
  static boolean sameElements(Object[] array1, Object[] array2) {
    Set<Object> uniqueElements1 = new HashSet<>(Arrays.asList(array1));
    Set<Object> uniqueElements2 = new HashSet<>(Arrays.asList(array2));
    if (uniqueElements1.size() != uniqueElements2.size()) return false;

    for (Object obj : uniqueElements1) {
      if (!uniqueElements2.contains(obj)) return false;
    }

    return true;
  }

}
```

STDIN

Output:

```
true
false
```

```java
public class MergeSort {
  public static void main(String[] args) {
    int[] arr = { 70, 50, 30, 10, 20, 40, 60 };
    int[] merged = mergeSort(arr, 0, arr.length - 1);
    for (int val : merged) {
      System.out.print(val + " ");
    }
  }
  public static int[] mergeTwoSortedArrays(int[] one, int[] two) {
    int[] sorted = new int[one.length + two.length];
    int i = 0;
    int j = 0;
    int k = 0;
    while (i < one.length && j < two.length) {
      if (one[i] < two[j]) {
        sorted[k] = one[i];
        k++;
        i++;
      } else {
        sorted[k] = two[j];
        k++;
        j++;
      }
    }
    if (i == one.length) {
      while (j < two.length) {
        sorted[k] = two[j];
        k++;
        j++;
      }
    }
    if (j == two.length) {
      while (i < one.length) {
        sorted[k] = one[i];
        k++;
        i++;
      }
    }
    return sorted;
  }
  public static int[] mergeSort(int[] arr, int lo, int hi) {
    if (lo == hi) {
      int[] br = new int[1];
```

STDIN

Input for the program (Optional)

Output:

```
10 20 30 40 50 60 70
```

```java
public class Armstrong {

    public static void main(String[] args) {

        int number = 1634, originalNumber, remainder, result = 0, n = 0;

        originalNumber = number;

        for (;originalNumber != 0; originalNumber /= 10, ++n);

        originalNumber = number;

        for (;originalNumber != 0; originalNumber /= 10)
        {
            remainder = originalNumber % 10;
            result += Math.pow(remainder, n);
        }

        if(result == number)
            System.out.println(number + " is an Armstrong number.");
        else
            System.out.println(number + " is not an Armstrong number.");
    }
}
```

STDIN

Output:

```
1634 is an Armstrong number.
```

```java
public class Armstrong {

    public static void main(String[] args) {

        int number = 371, originalNumber, remainder, result = 0;

        originalNumber = number;

        while (originalNumber != 0)
        {
            remainder = originalNumber % 10;
            result += Math.pow(remainder, 3);
            originalNumber /= 10;
        }

        if(result == number)
            System.out.println(number + " is an Armstrong number.");
        else
            System.out.println(number + " is not an Armstrong number.");
    }
}
```

STDIN

Output:

```
371 is an Armstrong number.
```