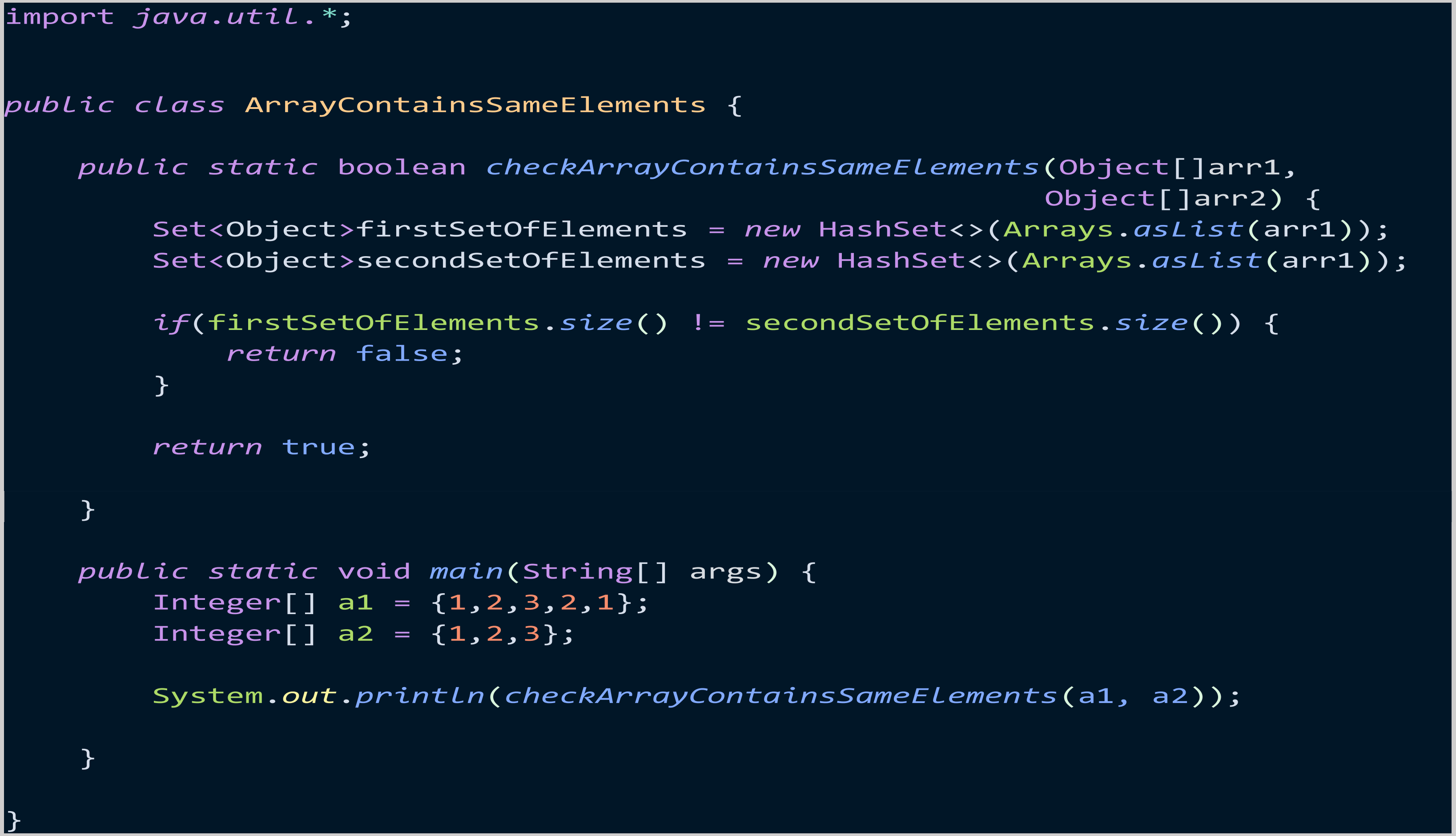
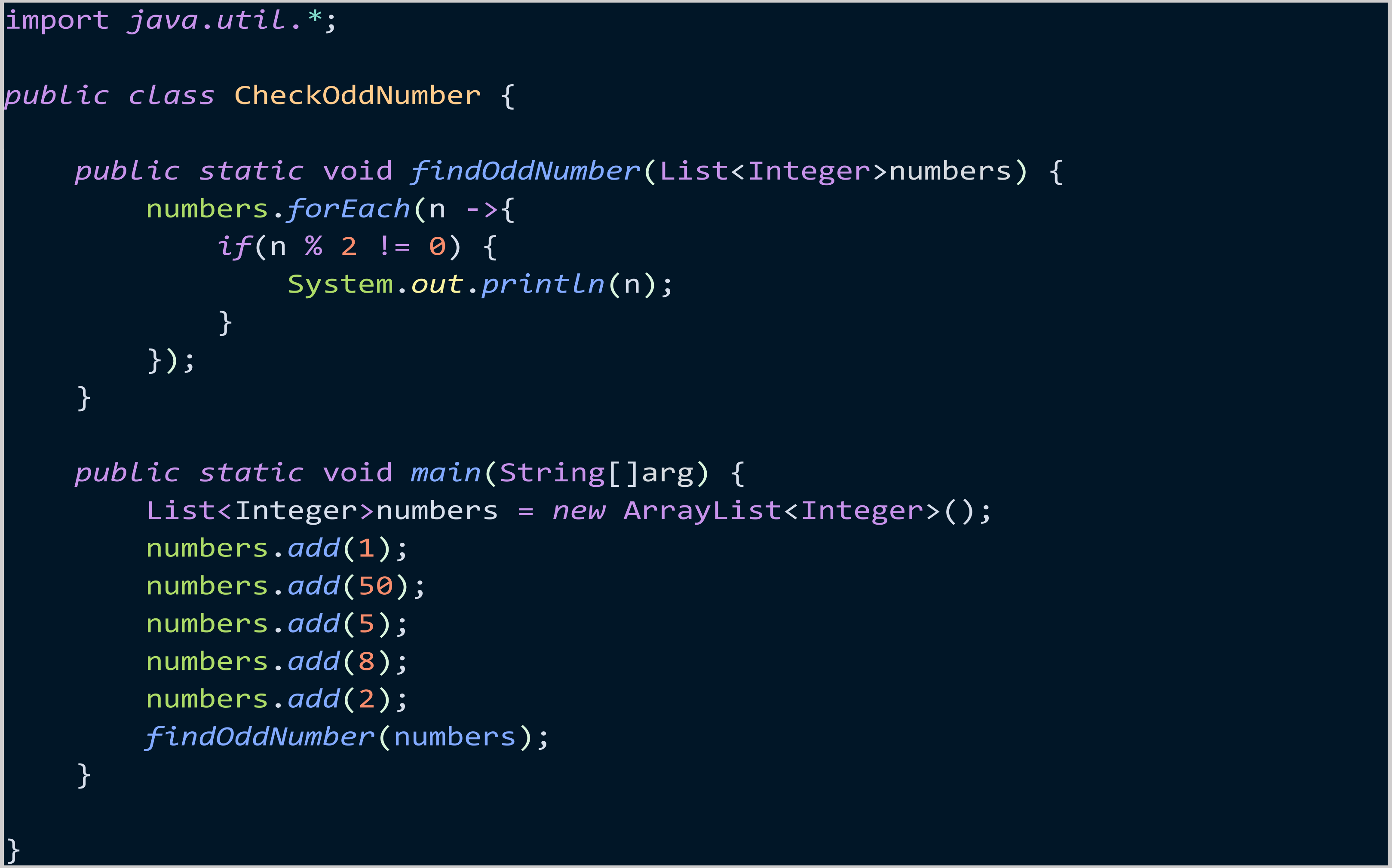
1. Determine two Arrays contain same elements.



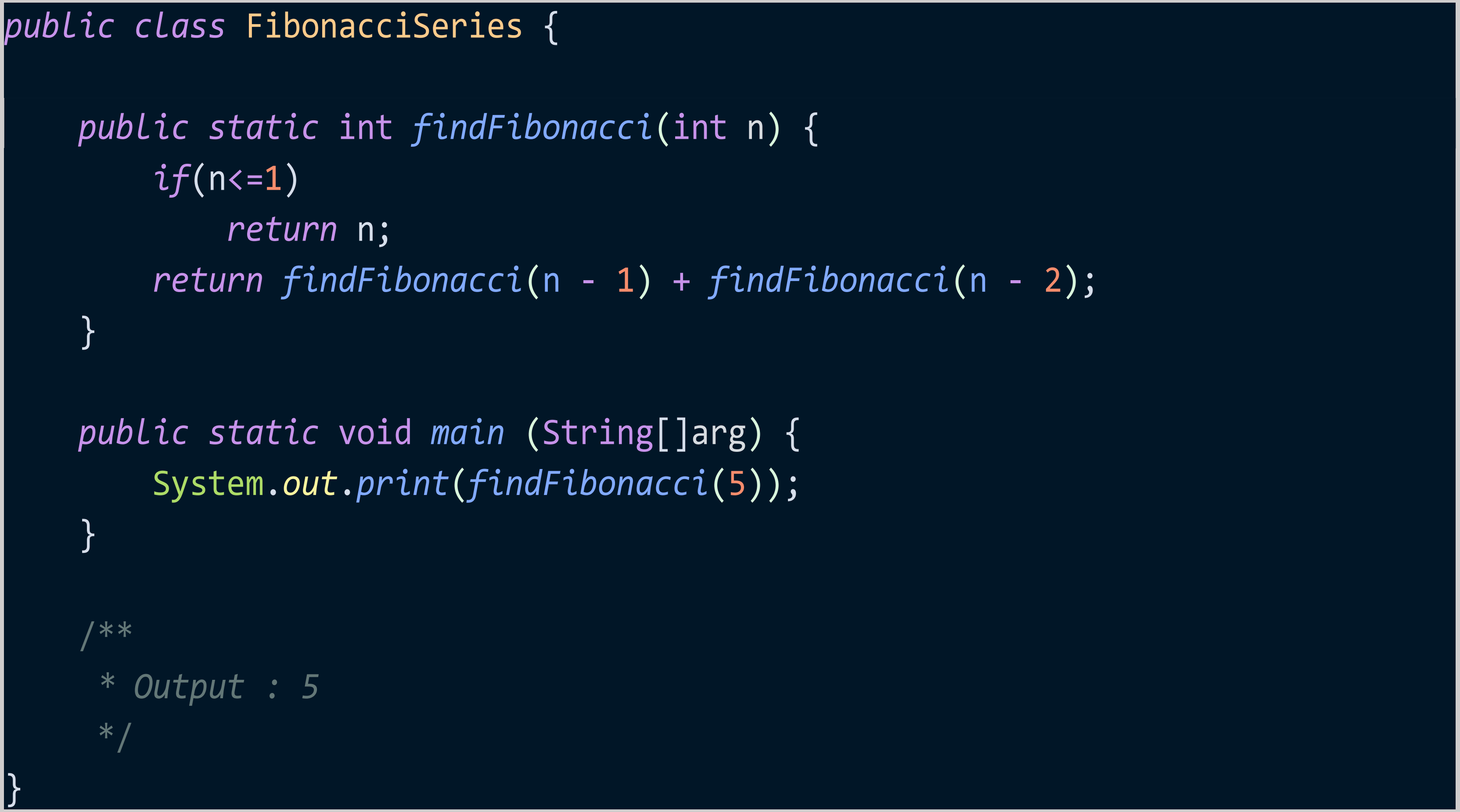
```java
import java.util.*;


public class ArrayContainsSameElements {

    public static boolean checkArrayContainsSameElements(Object[]arr1,
                                                         Object[]arr2) {
        Set<Object>firstSetOfElements = new HashSet<>(Arrays.asList(arr1));
        Set<Object>secondSetOfElements = new HashSet<>(Arrays.asList(arr1));

        if(firstSetOfElements.size() != secondSetOfElements.size()) {
            return false;
        }

        return true;

    }

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

        System.out.println(checkArrayContainsSameElements(a1, a2));

    }

}
```

2. Find Odd number from the given list and print.



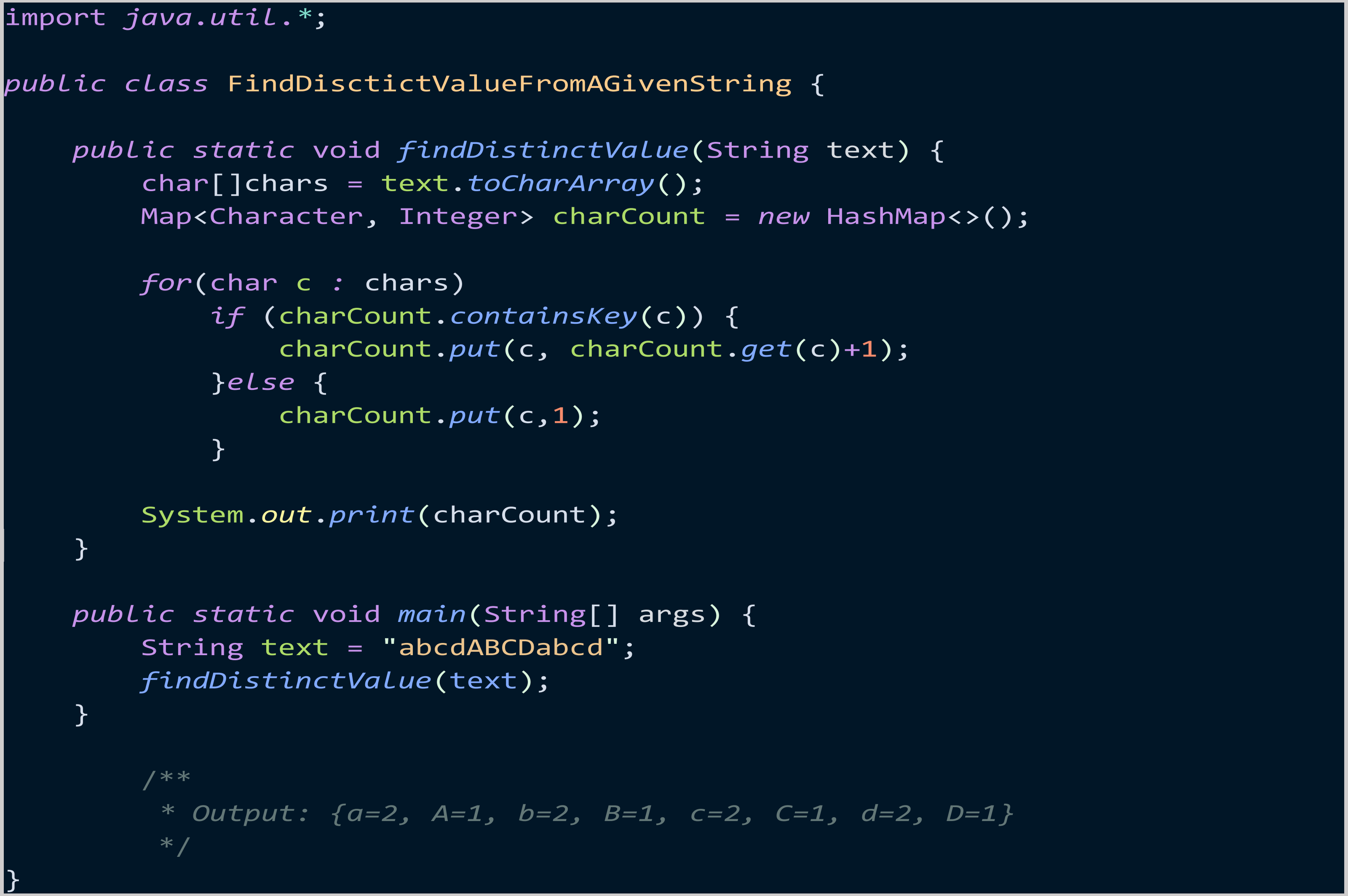
```java
import java.util.*;

public class CheckOddNumber {

    public static void findOddNumber(List<Integer>numbers) {
        numbers.forEach(n ->{
            if(n % 2 != 0) {
                System.out.println(n);
            }
        });
    }

    public static void main(String[]arg) {
        List<Integer>numbers = new ArrayList<Integer>();
        numbers.add(1);
        numbers.add(50);
        numbers.add(5);
        numbers.add(8);
        numbers.add(2);
        findOddNumber(numbers);
    }

}
```

3. Print Fibonacci of given value using Recursion.



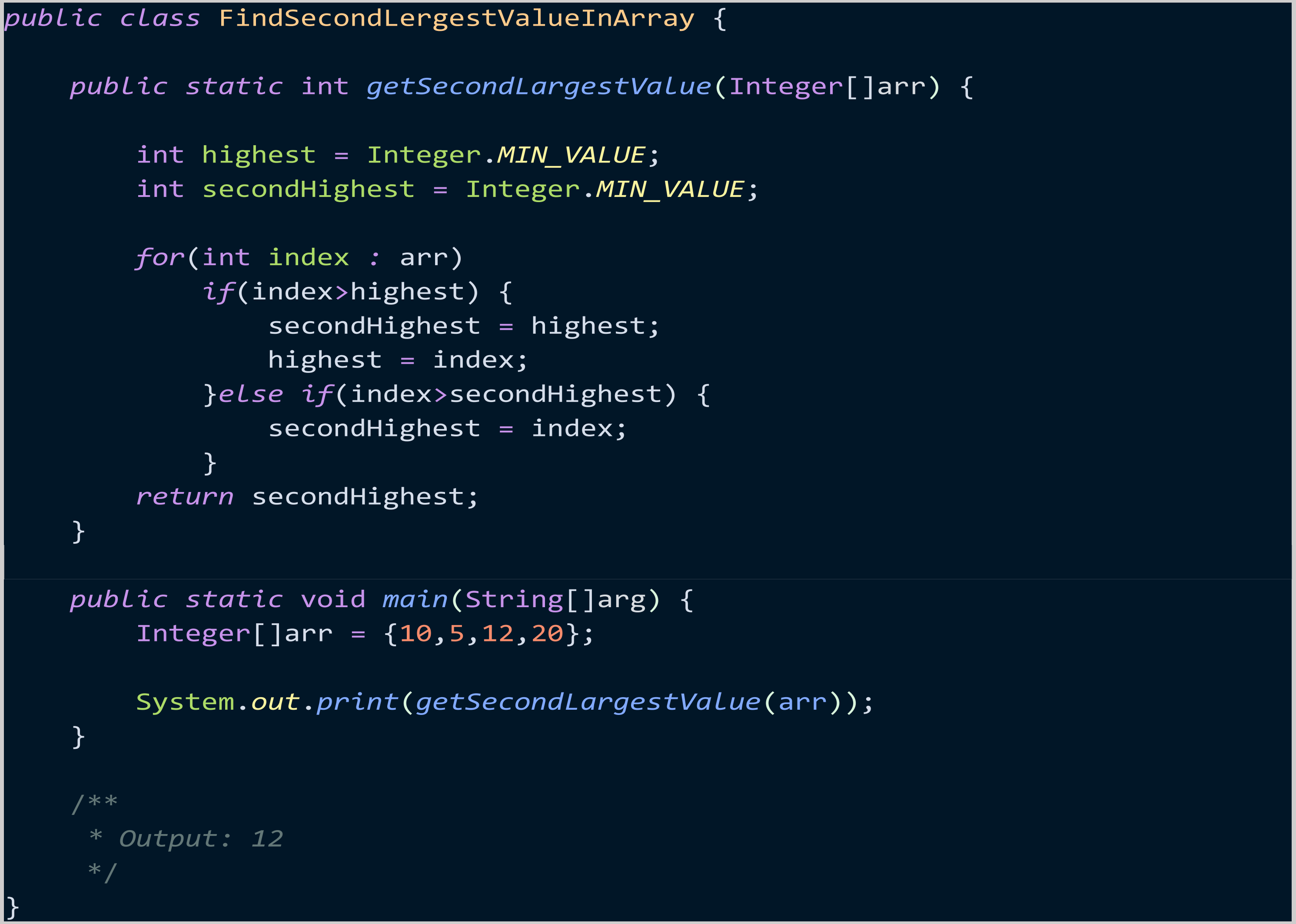
```java
public class FibonacciSeries {

    public static int findFibonacci(int n) {
        if(n<=1)
            return n;
        return findFibonacci(n - 1) + findFibonacci(n - 2);
    }

    public static void main (String[]arg) {
        System.out.print(findFibonacci(5));
    }

    /**
     * Output : 5
     */
}
```

4. Find Distinct value from a given String.



```java
import java.util.*;

public class FindDisctictValueFromAGivenString {

    public static void findDistinctValue(String text) {
        char[]chars = text.toCharArray();
        Map<Character, Integer> charCount = new HashMap<>();

        for(char c : chars)
            if (charCount.containsKey(c)) {
                charCount.put(c, charCount.get(c)+1);
            }else {
                charCount.put(c,1);
            }

        System.out.print(charCount);
    }

    public static void main(String[] args) {
        String text = "abcdABCDabcd";
        findDistinctValue(text);
    }

        /**
         * Output: {a=2, A=1, b=2, B=1, c=2, C=1, d=2, D=1}
         */
}
```

5. Find Second largest number from the given array of integers.

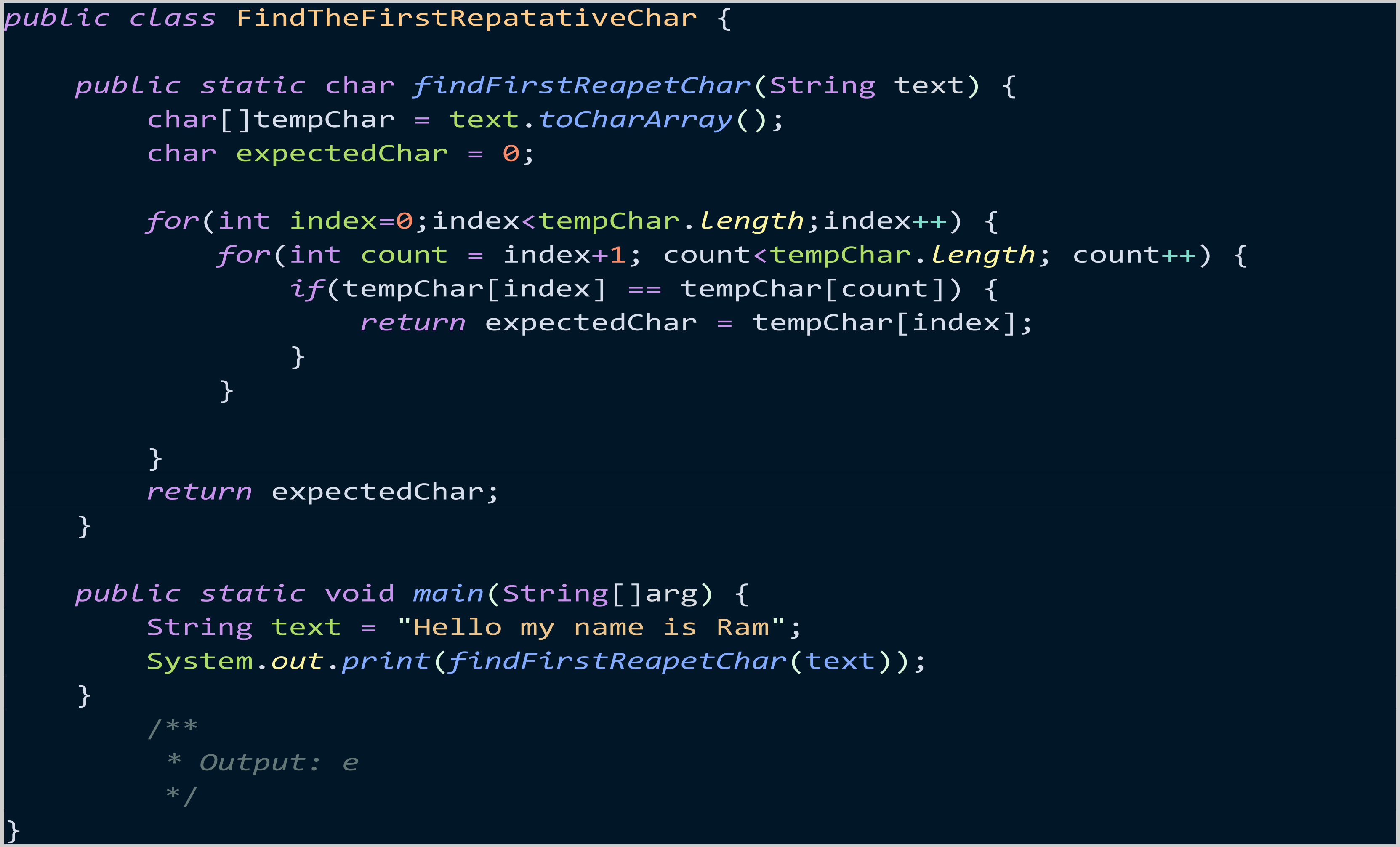


```java
public class FindSecondLergestValueInArray {

    public static int getSecondLargestValue(Integer[]arr) {

        int highest = Integer.MIN_VALUE;
        int secondHighest = Integer.MIN_VALUE;

        for(int index : arr)
            if(index>highest) {
                secondHighest = highest;
                highest = index;
            }else if(index>secondHighest) {
                secondHighest = index;
            }
        return secondHighest;
    }

    public static void main(String[]arg) {
        Integer[]arr = {10,5,12,20};

        System.out.print(getSecondLargestValue(arr));
    }

    /**
     * Output: 12
     */
}
```

6. Find Sum of all integers from the given Array.

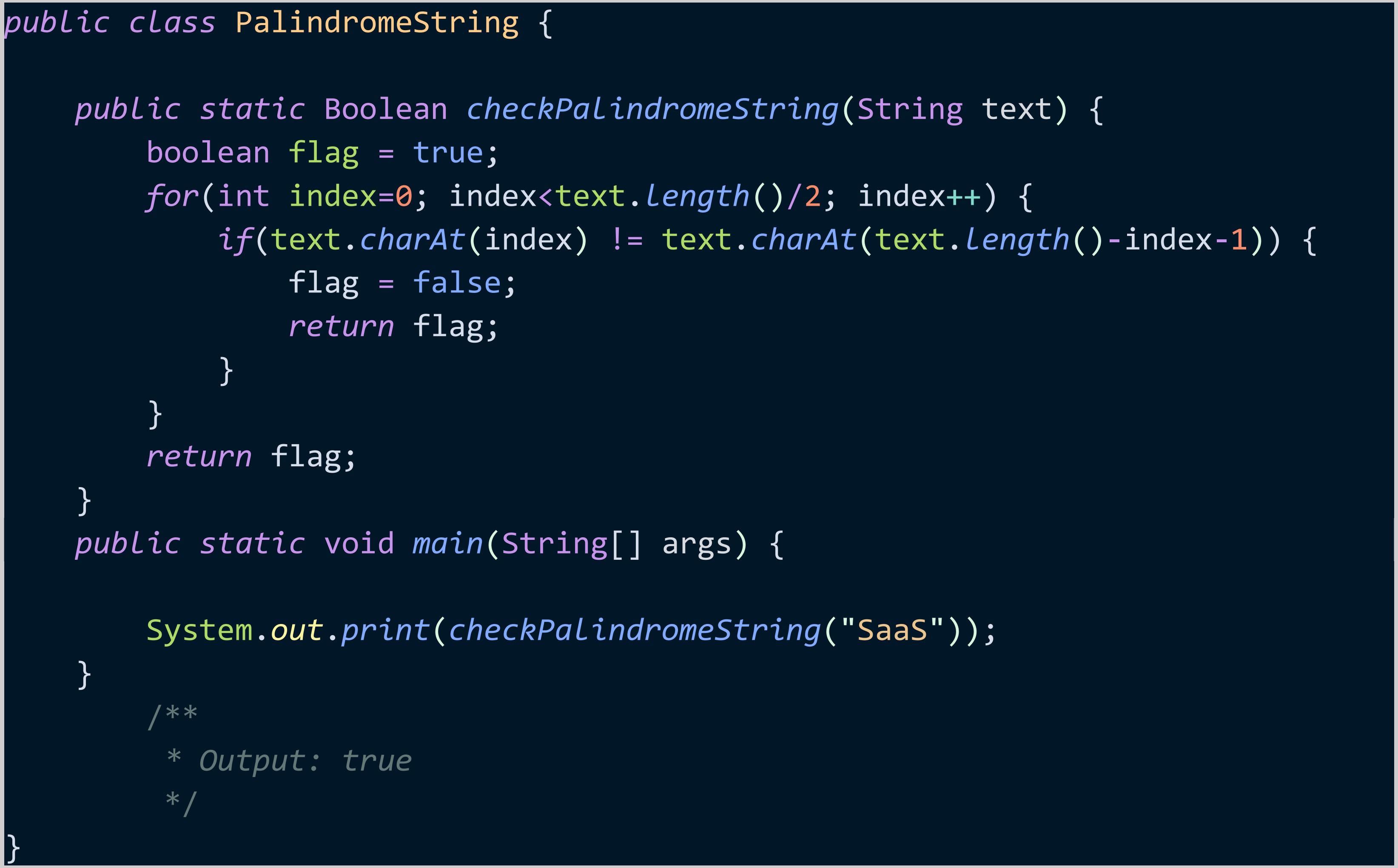
```java
public class FindSumOfAllIntegerInArray {

    public static int findSumOfallInteger(Integer[]arrayOfNumbers) {

        int sum = 0;
        for(int index: arrayOfNumbers) {
            sum += index;
        }

        return sum;
    }

    public static void main(String[]arg) {
        Integer[]arr = {10,5,20,5};

        System.out.print(findSumOfallInteger(arr));
    }

    /**
     * Output: 40
     */
}
```

7. Find the first repetitive character from the given String.



```
public class FindTheFirstRepatativeChar {

    public static char findFirstReapetChar(String text) {
        char[]tempChar = text.toCharArray();
        char expectedChar = 0;

        for(int index=0;index<tempChar.length;index++) {
            for(int count = index+1; count<tempChar.length; count++) {
                if(tempChar[index] == tempChar[count]) {
                    return expectedChar = tempChar[index];
                }
            }

        }
        return expectedChar;
    }

    public static void main(String[]arg) {
        String text = "Hello my name is Ram";
        System.out.print(findFirstReapetChar(text));
    }
        /**
         * Output: e
         */
}
```

8. Check the given string is a Palindrome.

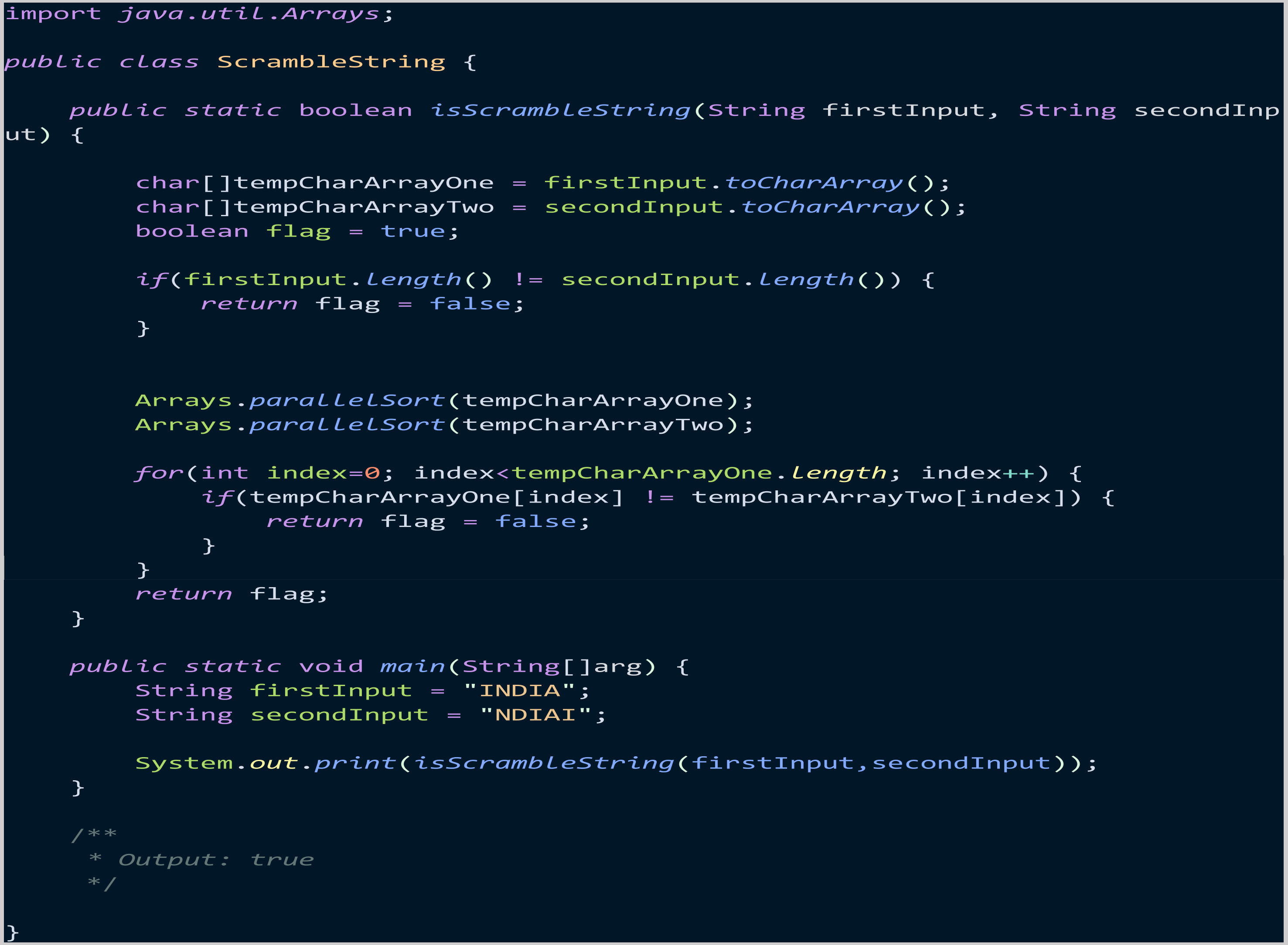


```java
public class PalindromeString {

    public static Boolean checkPalindromeString(String text) {
        boolean flag = true;
        for(int index=0; index<text.length()/2; index++) {
            if(text.charAt(index) != text.charAt(text.length()-index-1)) {
                flag = false;
                return flag;
            }
        }
        return flag;
    }
    public static void main(String[] args) {

        System.out.print(checkPalindromeString("SaaS"));
    }
        /**
         * Output: true
         */
}
```

9. Print the given string in Reverse order.

```java
public class ReverseString {

    public static void reverseString(String givenString) {

        char[]temp = givenString.toCharArray();
        StringBuilder sb = new StringBuilder();

        for(int index=temp.length-1; index>=0; index--) {
            sb.append(temp[index]);
        }

        System.out.println(sb);
    }


    public static void main(String[]arg) {
        reverseString("India");
    }
        /**
         * Output: aidnI
         */
}
```

10. Check the given strings are scramble to each other.



```
import java.util.Arrays;

public class ScrambleString {

    public static boolean isScrambleString(String firstInput, String secondInp
ut) {

        char[]tempCharArrayOne = firstInput.toCharArray();
        char[]tempCharArrayTwo = secondInput.toCharArray();
        boolean flag = true;

        if(firstInput.length() != secondInput.length()) {
            return flag = false;
        }


        Arrays.parallelSort(tempCharArrayOne);
        Arrays.parallelSort(tempCharArrayTwo);

        for(int index=0; index<tempCharArrayOne.length; index++) {
            if(tempCharArrayOne[index] != tempCharArrayTwo[index]) {
                return flag = false;
            }
        }
        return flag;
    }

    public static void main(String[]arg) {
        String firstInput = "INDIA";
        String secondInput = "NDIAI";

        System.out.print(isScrambleString(firstInput,secondInput));
    }

    /**
     * Output: true
     */

}
```