ArrayOps:

```java
import java.util.Arrays;

public class ArrayOps {
    public static void main(String[] args) {

    }

    public static int findMissingInt(int[] array) {
        // Write your code here:
        int arrl = array.length;
        for (int i = 0; i <= arrl; i++) {
            int m = i;
            boolean doesexist = false;
            for (int j = 0; j < arrl; j++) {
                if (m == array[j]) {
                    doesexist = true;
                    break;
                }
            }
            if (doesexist == false) {
                return m;
            }
        }
        return -1;
    }

    public static int secondMaxValue(int[] array) {
        // Write your code here:
        int max = array[0];
        int secondMax = array[0];
        int maxCount = 1;
        for (int i = 1; i < array.length; i++) {
            if (array[i] > max) {
                secondMax = max;
                max = array[i];
                maxCount = 1;
            } else if (array[i] == max) {
                maxCount++;
            } else if (array[i] > secondMax) {
                secondMax = array[i];
            }
        }
        return maxCount > 1 ? max : secondMax;
    }

    public static boolean containsTheSameElements(int[] array1, int[] array2)
{
```

```
        // Write your code here:
        String arr1 = Arrays.toString(array1);
        String arr2 = Arrays.toString(array2);
        for (int i = 0; i < arr1.length(); i++) {
            if (arr2.indexOf(arr1.charAt(i)) == -1) {
                return false;
            }
        }
        for (int i = 0; i < arr2.length(); i++) {
            if (arr1.indexOf(arr2.charAt(i)) == -1) {
                return false;
            }
        }
        return true;
    }

    public static boolean isSorted(int[] array) {
        // Write your code here:
        if (array.length < 2) {
            return true;
        }
        boolean increasing = array[0] <= array[1];
        for (int i = 1; i < array.length - 1; i++) {
            if (increasing && array[i] > array[i + 1]) {
                return false;
            } else if (!increasing && array[i] < array[i + 1]) {
                return false;
            }
        }
        return true;
    }

}
```

StringOps:

```
public class StringOps {
    ////////////////////////////////////////////////////////
    ////// ///////
    ////// Reminder: ///////
    ////// allowed methods ///////
    ////// ///////
    ////// 1.charAt(int index) ///////
    ////// 2.length() ///////
    ////// 3.substring(int start) ///////
    ////// 4.substring(int start,int ends) ///////
    ////// 5.indexOf(String str) ///////
    ////// ///////
    ////// The rest are not allowed ! ///////
    ////// if you want to use a different ///////
    ////// method, and you can implement ///////
    ////// it using material from the course ///////
    ////// you need to implement a version of ///////
    ////// the function by yourself. ///////
    ////// ///////
    ////// see example for substring ///////
    ////// in Recitation 3 question 5 ///////
    ////// ///////
    ////////////////////////////////////////////////////////
    public static void main(String[] args) {

    }

    public static String capVowelsLowRest(String string) {
        // Write your code here:
        char[] result = new char[string.length()];
        for (int i = 0; i < string.length(); i++) {
            char ch = string.charAt(i);
            if (ch >= 'A' && ch <= 'Z') {
                ch += 32; // Convert to lowercase
            }
            if (ch == 'a' || ch == 'e' || ch == 'i' || ch == 'o' || ch == 'u')
{
                ch -= 32; // Convert to uppercase
            }
            result[i] = ch;
        }
        return new String(result);
    }

    public static String camelCase(String string) {
        String result = "";
        boolean newWord = true;
```

```
        int wordCount = 0;
        for (int i = 0; i < string.length(); i++) {
            char ch = string.charAt(i);
            if (ch == ' ') {
                newWord = true;
                continue;
            }
            if (newWord) {
                if (wordCount == 0 && ch >= 'A' && ch <= 'Z') {
                    ch += 32; // Convert first letter of the first word to
lowercase
                } else if (wordCount > 0 && ch >= 'a' && ch <= 'z') {
                    ch -= 32; // Convert first letter of each word (except the
first word) to uppercase
                }
                newWord = false;
                wordCount++;
            } else if (ch >= 'A' && ch <= 'Z') {
                ch += 32; // Convert remaining letters to lowercase
            }
            result += ch;
        }
        return result;
    }

    public static int[] allIndexOf(String string, char chr) {
        // Write your code here:
        int sum = 0, j = 0;
        // this for loop to know how big is the array
        for (int i = 0; i < string.length(); i++) {
            if (chr == string.charAt(i))
                sum++;
        }
        int[] arr = new int[sum];
        for (int i = 0; i < string.length(); i++) {
            if (chr == string.charAt(i)) {
                arr[j] = i;
                j++;
            }
        }
        return arr;
    }
}
```