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



    }

    public static int findMissingInt(int[] array) {
        int n = array.length;

        for (int i = 0; i <= n; i++) {
            boolean contains = false;
            for (int j = 0; j < n; j++) {
                if (array[j] == i) {
                    contains = true;
                }
            }
            if (contains==false){
                return i;
            }
        }
        return 1;
    }

    public static int secondMaxValue(int[] array) {
        int n = array.length;
        int[] array2 = new int[array.length - 1];
        int index2 = 0;
        int max = array[0];

        for (int i = 0; i < n; i++) {
            if (array[i] > max) {
                max = array[i];
            }
        }
        for (int j = 0; j < array.length; j++) {
            if (array[j] != max) {
                array2[index2] = array[j];
            index2++;
        }
    }
    int max2 = array2[0];
    for (int k = 0; k < array2.length; k++) {
        if (array[k] > max2) {
            max2 = array2[k];
        }
    }

    return max2;
}

public static boolean containsTheSameElements(int[] array1, int[] array2) {
    boolean equality;
    for (int i = 0; i < array1.length; i++) {
        equality = false;
        for (int j = 0; j < array2.length; j++) {
            if (array1[i] == array2[j]) {
                equality = true;
            }
        }
        if (equality == false) {
            return false;
        }
    }

    for (int i = 0; i < array2.length; i++) {
        equality = false;
        for (int j = 0; j < array1.length; j++) {
            if (array2[i] == array1[j]) {
                equality = true;
            }
        }
        if (equality == false) {
            return false;
        }
    }


    return true;
}

public static boolean isSorted(int[] array) {
    int min = array[0];
    int max = array[0];
    boolean sortedInc =false;
    boolean sortedDec =false;
     for (int i = 0; i < array.length-1; i++){ //{9,7,4,6,5,1}
         sortedInc= false;
        if(array[i+1]>array[i]){
            sortedInc  = true;
        }
        else {
            sortedInc =  false;
            break;
        }


     }
     for (int i = 0; i < array.length-1; i++){ //{9,7,4,6,5,1}
         sortedDec= false;
        if(array[i+1]<array[i]){
            sortedDec = true;
        }
        else {
            sortedDec =  false;
            break;
        }
    }
    if(sortedDec == false  && sortedInc == false){
        return false;
    }


    return true;
}
```

```
}
```

```
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) {
    String string = "Hola cOMo ES";
    System.out.println(capVowelsLowRest(string));
}

public static String capVowelsLowRest(String string) {
    String answer = " ";
    String finalAnswer = " ";
    char currentChar;
    int length = string.length();
    for (int i = 0; i < length; i++) {
        currentChar = string.charAt(i);
        if (65 <= currentChar && currentChar <= 90) {
            currentChar = (char) (currentChar + 32);
            answer = answer + currentChar;
        } else {
            currentChar = (char) (currentChar);
            answer = answer + currentChar;
        }


    }
    for (int i = 0; i < answer.length(); i++) {
        if (answer.charAt(i) == 'a')
            finalAnswer = finalAnswer + ((char) (answer.charAt(i) - 32));
        else if (answer.charAt(i) == 'e')
            finalAnswer = finalAnswer + ((char) (answer.charAt(i) - 32));
        else if (answer.charAt(i) == 'i')
            finalAnswer = finalAnswer + ((char) (answer.charAt(i) - 32));
        else if (answer.charAt(i) == 'o')
            finalAnswer = finalAnswer + ((char) (answer.charAt(i) - 32));
        else if (answer.charAt(i) == 'u')
            finalAnswer = finalAnswer + ((char) (answer.charAt(i) - 32));
        else
            finalAnswer = finalAnswer + answer.charAt(i);
    }
    return finalAnswer;

}

private static char characterToUpperCase(char character) {
    if (character >= 97 && character <= 122) {
        return (char)(character - 32);
    }
    return character;
}
private static char characterToLowerCase(char character) {
    if (character >= 65 && character <= 90) {
        return (char)(character + 32);
    }
    return character;
}

public static String camelCase(String string) {
    String answer = " ";

    for (int i= 0; i<string.length(); i++){
        while(string.charAt(i) == ' '){
            i++;
            }
            if(answer!= "" && string.charAt(i - 1) == ' '){
                answer = answer + characterToUpperCase(string.charAt(i));
            }
            else{
                answer  = answer + characterToLowerCase(string.charAt(i));
            }
        }
        return answer;
    }

    public static int[] allIndexOf(String string, char chr) {
        int counter =0;
        // we go over the string characters and count them do decide the size of the  array
        for (int i= 0; i<string.length(); i++){
            if(string.charAt(i)== chr){
                counter++;
            }


        }
        int[] array = new int[counter]; //the size of the arrayn is the amount of times the character repeats  itself
        int indexInArray = 0;
        //loop that fills up the array with the inex value of the character in the given string
        for (int i= 0; i<string.length(); i++){
            if(string.charAt(i)== chr){
                array[indexInArray]= i;
                indexInArray++;
            }
        }

        return array;
    }
}
```