```java
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) {
        int[] newarray = allIndexOf("hello world",'l');
    for (int i = 0; i < newarray.length; i++) {
        System.out.print(newarray[i]);
    }


    }

    public static String capVowelsLowRest (String string) {
        String s = "";
        int i = 0;
        for (string.charAt (i); i < string.length(); i++) {
            if (string.charAt(i) == 97 || string.charAt(i) == 101 ||
string.charAt(i) == 105 || string.charAt(i) == 111 || string.charAt(i) == 117)
{
                s = s + (char)(string.charAt (i) - 32);
            } else if (string.charAt(i) < 91 && string.charAt(i) > 65 &&
string.charAt(i) != 69 && string.charAt(i) != 73 && string.charAt(i) != 79 &&
string.charAt(i) != 85) {
                s = s + (char)(string.charAt (i) + 32);
            } else {
                s = s + string.charAt(i);
            }
        }
        return s;
    }

    public static String camelCase (String string) {
        string = deleteSpacebeginning (string);
        string = lowerCase(string);
        string = upperNew(string);
        string = deleteSpace(string);
        return string;
    }

    public static String lowerCase (String string) {
        String s = "";
        for (int i = 0; i < string.length(); i++) {
            if (string.charAt(i) > 64 && string.charAt(i) < 91) {
                s = s + (char)(string.charAt(i) + 32);
            } else {
              s = s + string.charAt(i);
            }
        }
        return s;
    }

    public static String upperNew (String string) {
        String s = "" + string.charAt(0);
        for (int i = 1; i < string.length(); i++) {
            if (string.charAt(i) == 32) {
                s = s + string.charAt(i);
            }else if (string.charAt(i-1) == 32) {
                s = s + (char)(string.charAt(i) - 32);
            } else {
              s = s + string.charAt(i);
            }
        }
        return s;
    }

    public static String deleteSpace (String string) {
        String s = "";
        for (int i = 0; i < string.length(); i++) {
            if (string.charAt(i) != ' ') {
              s += string.charAt(i);
            }
        }
        return s;
    }

    public static String deleteSpacebeginning (String string) {
        String s = "";
        for (int i = 0; i < string.length(); i++) {
            if (string.charAt(i) != ' ') {
              s = string.substring(i); break;
            }
        }
        return s;
    }

    public static int[] allIndexOf (String string, char chr) {
        int counter = 0;
        for (int i = 0; i < string.length(); i++) {
            if (string.charAt(i) == chr) {
                counter++;
             }
        }
        int[] array = new int[counter];
        for (int r = 0; r < array.length; r++) {
        for (int j = 0; j < string.length(); j++) {
            if (string.charAt(j) == chr) {
            array [r] = j;
            r++;
            }
        }
    }
        return array;
    }
}
```

```java
public class ArrayOps {
    public static void main(String[] args) {
        System.out.println(findMissingInt(new int[]{1,0,3}));
        System.out.println(secondMaxValue(new int[] {6, 9, 4, 7, 3, 4}));
        System.out.println(containsTheSameElements(new int[] {1, 4, 1, 1, 2},
new int[] {2, 1, 4}));
        System.out.println(isSorted(new int[] {7, 5, 4, 3, -12}));
        System.out.println(isSorted(new int[] {1, -2, 3}));
        System.out.println(isSorted(new int[] {1,2,3}));
    }

    public static int findMissingInt (int [] array) {
        boolean inArray;

        for (int index = 0; index < array.length; index++) {
            inArray = false;
           for (int p = 0; p < array.length; p++) {
            if (array[p] == index) {
                inArray = true;
            }
           }
            if (inArray == false) {
                return index;

            }

        }
        return array.length;
    }

    public static int secondMaxValue(int [] array) {
        int max = array[0];
        for (int index = 0; index < array.length; index++) {
            if (array[index] >= max) {
                max = array[index];
            }
        }
        int check = 0;
        for (int index = 0; index < array.length; index++) {
            if (array[index] == max) {
                check++;
            }
        }
        if (check > 1) {
            return max;
        }
        int max2 = array[0];
        for (int i = 0; i < array.length; i++) {
             if (array[i] > max2 && array[i]!=max) {
                max2 = array[i];
             }
            }

             return max2;
        }


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

    public static boolean isSorted(int [] array) {
        boolean same = false;
        for (int i = 0; i < array.length - 1; i++) {
            if (array[i] < array[i+1]) {
                same = true;
            } else {
            same = false;
            break;
        }
        if (same == true) {
            return same;
        }

    }
        for (int j = 0; j < array.length - 1; j++) {
            if (array[j] > array[j+1]) {
                same = true;
            } else {
            same = false;
            break;
            }
        }
    return same;
    }
}
```