FindMissingInt

```java
public static int findMissingInt (int [] array) {
    if (array.length == 1) return (array[0] + 1);
    int missing = 0;
    int max = 0;
    boolean foundM =  false;
    for (int i = 0; i < array.length; i++) {
        for (int j = 0; j < array.length; j++) {
            if (array[i] + 1 == array[j]) {
                break;
            }
            max = Math.max(array[j], max);
            if (j == array.length - 1 && i != array.length - 1) {
                missing = array[i] + 1;
                foundM = true;
            }
        }
        if (max < missing) return 0;
        if (foundM) break;
    }
    return missing;
}
```

secondMaxValue

```java
public static int secondMaxValue(int [] array) {
        int max = array[0];
        int max2 = Math.min (array[0], array [1]);
        for (int i = 0; i < array.length; i++) {
            for (int j = 0; j < array.length; j++) {
                max = Math.max(max, Math.max(array[i], array[j]));
                if (array[j] < max) {
                    max2 = Math.max(max2, array[j]);
                }
                if (array[i] == array[j] && array[j] == max && i > 0 &&
                    i != j) return max;

            }
            if (max2 == max) {
                max2 = Math.min(max2, array[i]);
            }
        }
        return max2;
    }
```

containsTheSameElement

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

IsSorted

```java
public static boolean isSorted(int [] array) {
    int dir = 0; //is 0 if array is going down or 1 if array is
                 going up and 2 if a started with equal numbers
    for (int i = 0; i < array.length - 1; i++) {
        if ((i == 0 || dir == 2) && array[i] > array[i+1]) {
            dir = 0;
        } else if ((i == 0 || dir == 2) && array[i] < array[i+1])
        {
            dir = 1;
        } else if (i == 0 || dir == 2) {
            dir = 2;
        }
        if (array[i] > array[i+1] && dir == 1) {
            return false;
        } else if (array[i] < array[i+1] && dir == 0) {
            return false;
        }
    }
    return true;
}
```

CapVowelsLowRest

```
    public static String capVowelsLowRest (String string) {
        String s = "";
        for (int i = 0; i < string.length(); i++) {
            if (string.charAt(i) >= 65 && string.charAt(i) <= 90) {
                if (string.charAt(i) == 'A' || string.charAt(i) == 'I'
                    || string.charAt(i) == 'O' || string.charAt(i) ==
                    'E' || string.charAt(i) == 'U' ) {
                    s += string.charAt(i);
                } else {
                    s += (char)(string.charAt(i) + 32);
                }
            } else if (string.charAt(i) >= 97 && string.charAt(i) <=
                       122) {
                if (string.charAt(i) == 'a' || string.charAt(i) == 'i'
                    || string.charAt(i) == 'o' || string.charAt(i) ==
                    'e' || string.charAt(i) == 'u' ) {
                    s += (char)(string.charAt(i) - 32);
                } else {
                    s += string.charAt(i);
                }
            } else {
                s += string.charAt(i);
            }
        }
        return s;
    }
```

CamelCase

```
    public static String camelCase (String string) {
        String s = "";
        boolean first = true;
        for (int i = 0; i < string.length(); i++) {
            if (s == "") first = true;
            if (string.charAt(i) == ' ') {
                first = false;
            }
            if (first == true) {
                if (string.charAt(i) >= 65 && string.charAt(i) <= 90)
                    {
                    s += (char)(string.charAt(i) + 32);
                } else if (string.charAt(i) >= 97 && string.charAt(i)
                            <= 122) {
                    s += string.charAt(i);
                }
            } else {
                if (string.charAt(i - 1) == ' ') {
                    if (string.charAt(i) >= 65 && string.charAt(i) <=
                        90) {
                        s += string.charAt(i);
                    } else if (string.charAt(i) >= 97 &&
                                string.charAt(i) <= 122) {
                        s += (char)(string.charAt(i) - 32);
                    }
                } else if (string.charAt(i) >= 65 && string.charAt(i)
                            <= 90) {
```

```
                s += (char)(string.charAt(i) + 32);
            } else if (string.charAt(i) >= 97 && string.charAt(i)
                    <= 122) {
                s += string.charAt(i);
            }
        }
    }
    return s;
}
```

AllIndexOf

```java
public static int[] allIndexOf (String string, char chr) {
    int cnt = 0;
    int iCount = 0;
    for (int i = 0; i < string.length(); i++) {
        if (string.charAt(i) == chr) {
            cnt++;
        }
    }
    int[] indexOf = new int[cnt];s
    for (int i = 0; i < string.length(); i++) {
        if (string.charAt(i) == chr) {
            indexOf[iCount] = i;
            iCount++;
        }
    }
    return indexOf;
}
```