

```

public class ArrayOps {

    public static void main(String[] args) {

    }

    public static int findMissingInt (int [] array) {

        //the program gets an array of natural numbers from 0-n, with a missing number in the
        sequence, and returns it

        for (int i = 0; i < array.length; i++) {
            if (!isInTheArr(array, i)) {
                return i;
            }
        }

        return (array.length);
    }

    public static boolean isInTheArr (int [] array, int n) {

        // the program gets an int n, and an array of ints. Return true if n exist in the array and
        false otherwise

        for (int i = 0; i < array.length; i++) {
            if (array [i] == n) {
                return true;
            }
        }

        return false;
    }

    public static int secondMaxValue(int [] array) {

        // the program gets an array of ints and return the second biggest int

        int max = Math.max(array[0] , array[1]);

        int indexMax = 0;

        int secMax = Math.min(array[0] , array[1]);
    }

```

```

for (int i = 0; i < array.length; i++) {
    if ( array [i] >= max) {
        secMax = max;
        max = array[i];
    }
}
return secMax;
}

```

```

public static boolean containsTheSameElements(int [] array1,int [] array2) {
    for (int i = 0; i < array2.length; i++) { // A loop that checks if array2 is contained in array1
        if (!isInTheArr(array1, array2[i])) {
            return false;
        }
    }
    for (int i = 0; i < array1.length; i++) { // A loop that checks if array1 is contained in array2
        if (!isInTheArr(array2, array1[i])) {
            return false;
        }
    }
    return true;
}

```

```

public static boolean isSorted(int [] array) {
    boolean increase = false;
    boolean decrease = false;
    for (int i = 1; i < array.length; i++) { // checks if the array is sorted decreasingly
        if (array [i - 1] >= array [i]) {
            decrease = true;
        }
    }
}

```

```

        else {
            decrease = false;
            break;
        }
    }

    for (int i = 1; i < array.length; i++) { // checks if the array is sorted increasingly
        if (array [i - 1] <= array [i]) {
            increase = true;
        }
        else {
            increase = false;
            break;
        }
    }

    return (increase || decrease);
}
}

```

```
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) {

// A program that gets a string and returns it with lower letters, and vowels as upper
letters

String capV= "";

for (int i = 0; i < string.length(); i++) {

    if (isVowel(string.charAt(i))) {

        capV += lowToUpper(string.charAt(i));

    }

    else if (isVowelUpperCase(string.charAt(i))){

        capV += string.charAt(i);

    }

    else if ((string.charAt(i) >= 'A') && (string.charAt(i) <= 'Z')) {

        capV += upToLower(string.charAt(i));

    }

    else {

        capV += string.charAt(i);

    }

}
}

```

```

    }
}
return capV;
}

```

```

public static boolean isVowel (char c) { // check if the char is a vowel

    String vowels = "aouie";

    for (int i = 0; i < vowels.length(); i++) {

        if ( c == vowels.charAt(i)) {

            return true;

        }

    }

    return false;

}

```

```

public static boolean isVowelUpperCase (char c) { // check if the char is a vowel in upper
case

    String vowelsUpp= "AOUIE";

    for (int i = 0; i < vowelsUpp.length(); i++) {

        if ( c == vowelsUpp.charAt(i)) {

            return true;

        }

    }

    return false;

}

```

```

public static char lowToUpper (char c) { // gets a lower letter and switch it to upper

    char a = (char)((int)(c) - 32);

    return a;

}

```

```

public static char upToLower (char c) { // gets a capital letter and switch it to lower case

    char a = (char)((int)(c) + 32);

    return a;

}

```

```

public static String camelCase (String string) {

/* A program that gets a sentence and returns it connects,
when the first word in lower letters, and the rest of the words starts with a capital
letter and continue with lower letters */

    boolean first = true ;

    String camel = "" ;

    char last = string.charAt(0);

    for (int i = 0; i < string.length(); i++) {

        if (string.charAt(i) == ' ') {

            last = ' ';

        }

        else if (first) {

            // checks if its the first word, and if so add the whole in lower case to 'camel'

            if ((string.charAt(i) >= 'A') && (string.charAt(i) <= 'Z'))

            {

                camel += upToLower(string.charAt(i));

                first = false;

                last = string.charAt(i);

            }

            else {

                camel += string.charAt(i);

                first = false;

                last = string.charAt(i);

            }

        }

        else if (last == ' ') {

```

// checks if the last char was space, and if so turns the next letter to capital letter and add it to 'camel'

```
        if ((string.charAt(i) >= 'A') && (string.charAt(i) <= 'Z'))
        {
            camel += string.charAt(i);
            last = string.charAt(i);
        }
        else
        {
            camel += lowToUpper(string.charAt(i));
            last = string.charAt(i);
        }
    }
    // add the other letters to 'camel' as lower case
    else if ((string.charAt(i) >= 'A') && (string.charAt(i) <= 'Z'))
    {
        camel += upToLower(string.charAt(i));
        last = string.charAt(i);
    }
    else {
        camel += string.charAt(i);
        last = string.charAt(i);
    }
}
return camel;
}
```

```
public static String correctWord (String word) {
    boolean first = true;
    String correct = "";
    for (int i = 0; i < word.length(); i++) {
```

```

    if (first) {
        if ((word.charAt(i) >= 'A') && (word.charAt(i) <= 'Z')) {
            correct += word.charAt(i);
        }
        else {
            correct += lowToUpper(word.charAt(i));
        }
        first = false;
    }
    else {
        if ((word.charAt(i) >= 'A') && (word.charAt(i) <= 'Z')) {
            correct += upToLower(word.charAt(i));
        }
        else {
            correct += word.charAt(i);
        }
    }
}
return correct;
}

```

```

public static String lowerLettersWord (String word) {
    String correct = "";
    for (int i = 0; i < word.length(); i++) {
        if ((word.charAt(i) >= 'A') && (word.charAt(i) <= 'Z')) {
            correct += upToLower(word.charAt(i));
        }
        else {
            correct += word.charAt(i);
        }
    }
}

```



```

        return correct;
    }

    public static int[] allIndexOf(String string, char chr)
    {
        int count= 0;
        for (int i= 0; i < string.length(); i++) {
            if (string.charAt(i) == chr) {
                count++;
            }
        }
        int [] index = new int [count];
        int arr= 0;
        for ( int i = 0; i < string.length(); i++ ) {
            if (string.charAt(i) == chr) {
                index [arr] = i;
                arr++;
            }
        }
        return index;
    }
}

```