ArrayOps

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

   }

public static int findMissingInt(int [] array) {

      int[] arr1 = new int[array.length + 1];
      int a = 0;

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

         arr1[array[i]] = array[i];

      }

      for(int j = 1; j < arr1.length; j++){

         if (arr1[j] == 0) a = j;

      }

      return a;
   }
```

```
public static int secondMaxValue(int [] array) {

    int max = array[0];
    int maxIndex = 0;
    int secondMax = array[0];

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

        if (array[i] > max) {

            max = array[i];
            maxIndex = i;

        }

        if (i == array.length-1) {

            array[maxIndex] = 0;

            for (int j = 0; j < array.length; j++){

                if (array[j] > secondMax) {

                    secondMax = array[j];

                }

            }

        }
    }

    return secondMax;
}
```

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

        int greaterArr = Math.max(array1.length, array2.length);
        int lowerArr = Math.min(array1.length, array2.length);
        int num = 0;

        for (int i = 0; i < greaterArr; i++){

            for (int j = 0; j < lowerArr; j++){

                if(array1.length == greaterArr && array1[i] == array2[j]){

                    num++;
                    break;

                } else if (array2.length == greaterArr && array2[i] == array1[j]) {

                    num++;
                    break;

                }

            }

        }

        if (num == greaterArr) {

            return true;

        }

    return false;
}
```

```
public static boolean isSorted(int [] array) {
        if (array[0] < array [1]){
            for (int i = 2; i < array.length; i++){
                if (array[i] < array[i-1]){
                    return false;
                }
            }
        }

        if (array[0] > array[1]){
            for (int i = 2; i < array.length; i++){
                if (array[i] > array[i-1]){
                    return false;
                }
            }
        }
      return true;
   }
}
```

StringOps

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

   }

   public static String capVowelsLowRest (String string) {

      String newS = "";
      char C;

      for (int i = 0; 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){

            C = string.charAt(i);
            C -= 32;
            newS += C;

         } else if (string.charAt(i) >= 98 && string.charAt(i) <= 122 || string.charAt(i) == 32
                  || string.charAt(i) == 65 || string.charAt(i) == 69 || string.charAt(i) == 73
                  || string.charAt(i) == 79 || string.charAt(i) == 85){

               newS += string.charAt(i);

            } else {

               C = string.charAt(i);
               C += 32;
               newS += C;

            }

      }

      return newS;
   }
```

```
public static String camelCase (String string) {

    int indexLetter = 0;
    String substr = "";
    String newS = "";
    char C;

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

        if (string.charAt(i) >= 65){

            indexLetter = string.indexOf(string.charAt(i));
            break;
        }
    }

    substr = string.substring(indexLetter);

    if(substr.charAt(0) < 97){

        C = substr.charAt(0);
        C += 32;
        newS += C;

        } else {

            newS += substr.charAt(0);

        }

    for(int j = 1; j < substr.length(); j++){

        C = substr.charAt(j);

        if ((substr.charAt(j-1) == 32) && (substr.charAt(j) >= 97)){

            C = substr.charAt(j);
            C -= 32;
            newS += C;
```

```
        } else if ((substr.charAt(j) < 97) && (substr.charAt(j-1) != 32) && (C != 32)){

            C = substr.charAt(j);
            C += 32;
            newS += C;

        } else if (C != 32){

            newS += C;

            }
    }

    return newS;
}
```

```
    public static int[] allIndexOf (String string, char chr) {

        int indexCounter = 0;
        int arrayIndexCounter = 0;

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

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

                indexCounter++;

            }
        }

        int[] arr1 = new int[indexCounter];

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

            for (int i = arrayIndexCounter; i < arr1.length; i++){

              if (string.charAt(j) == chr){

                    arr1[i] = j;
                    arrayIndexCounter++;
                    break;

                }
            }
        }

        return arr1;
    }
}
```