```
public class ArrayOps {
    public static void main(String[] args) {
        //findMissingInt(new int[] {3, 0, 1});
        //System.out.print(secondMaxValue(new int[] {-202, 48, 13, 7, 8}));
        //System.out.print(isSorted(new int[] {1, 1, 500}));
        //System.out.print(containsTheSameElements(new int[] {2,3}, new int[] {2,3,5}));
    }

    public static int findMissingInt (int [] array) {
        for(int i =0 ; i<=array.length ; i++){
            boolean isInArray = false;
            for(int j = 0 ; j<array.length ;j++){
                if(array[j] == i){
                    isInArray = true;
                    break;
                }
            }
            if(!isInArray){
                return i;
            }
        }
        return -1;
    }

    public static int secondMaxValue(int [] array) {
        int max = Math.max(array[0], array[1]);
        int secondMax = Math.min(array[0], array[1]);
        if(array.length == 2){
            return secondMax;
        }
        for(int i = 2 ;i<array.length; i++){
            if(array[i] >= max){
                secondMax = max;
                max = array[i];
```

```
        }
        if(array[i] < max && array[i] > secondMax){
            secondMax = array[i];
        }

    }
    return secondMax;
}

public static boolean containsTheSameElements(int [] array1,int [] array2) {
    for (int i = 0 ; i< array1.length;i++){
        boolean isContainedInArray2 = false;
        for (int j = 0 ; j<array2.length; j++){
            if(array2[j] == array1[i]){
                isContainedInArray2 = true;
            }
        }
        if(!isContainedInArray2){
            return false;
        }
    }
    for (int i = 0 ; i< array2.length;i++){
        boolean isContainedInArray1 = false;
        for (int j = 0 ; j<array1.length; j++){
            if(array1[j] == array2[i]){
                isContainedInArray1 = true;
            }
        }
        if(!isContainedInArray1){
            return false;
        }
    }
    return true;
}
```

```
    public static boolean isSorted(int [] array) {
        int numberBefore = array[0];
        boolean increasing = false;
        boolean decreasing = false;
        for (int i = 1; i< array.length ; i++){
            if(!increasing && !decreasing){
                if(numberBefore > array[i]){
                    decreasing = true;
                }
                if(numberBefore< array[i]){
                    increasing = true;
                }
            }
            else{
                if(increasing && (numberBefore > array[i])){
                    return false;
                }
                if(decreasing && (numberBefore < array[i])){
                    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) {
    System.out.print(capVowelsLowRest("One two tHRee world"));
  }

  public static String capVowelsLowRest (String string) {
    // Write your code here:
    String newString = "";
    char[] vowels = new char[]{'a','e','i','o','u'};
    for(int i=0 ; i<string.length(); i++){
      boolean isVowel= false;
      for (int j= 0 ; j<vowels.length;j++){
```

```
            if(string.charAt(i) == vowels[j] || string.charAt(i) == (char) (vowels[j]-32)){
                isVowel = true;
            }
        }
        //checks if the char is upper case
        if(string.charAt(i) <=90 && string.charAt(i) >= 65){
            if(isVowel == false){
                newString += (char) (string.charAt(i)+32);
            }
            else{
                newString += string.charAt(i);
            }
        }
        //checks if the char is lower case
        else if(string.charAt(i) <=122 && string.charAt(i) >= 97){
            if(isVowel== true){
                newString += (char) (string.charAt(i)-32);
            }
            else{
                newString += string.charAt(i);
            }
        }
        else {
            newString += string.charAt(i);
        }
    }
    return newString;
}

public static String camelCase (String string) {
    // Write your code here:
    String newString = "";
    boolean didSeeFirstLetter = false;
    boolean nextCharCapital = false;
```

```
    for (int i = 0 ; i<string.length(); i++){
        if(string.charAt(i) == 32){
            if(didSeeFirstLetter){
                nextCharCapital = true;
            }
        }
        else{
            if(!didSeeFirstLetter){
                didSeeFirstLetter = true;
            }
            if(nextCharCapital == true){
                if(string.charAt(i) <=122 && string.charAt(i) >= 97){
                    newString += (char) (string.charAt(i)-32);
                }
                else{
                    newString += string.charAt(i);
                }
            }
            else{
                if(string.charAt(i) <=90 && string.charAt(i) >= 65){
                    newString += (char) (string.charAt(i)+32);
                }
                else{
                    newString += string.charAt(i);
                }
            }
            nextCharCapital= false;
        }
    }
    return newString;
}

public static int[] allIndexOf (String string, char chr) {
    // Write your code here:
```

```
        int numberOfTimesChar= 0;
        for(int i = 0; i<string.length(); i++){
            if(string.charAt(i) == chr){
                numberOfTimesChar++;
            }
        }
        int[] indexesOfChar = new int[numberOfTimesChar];
        int indexOfArray = 0;
        for(int i = 0; i<string.length(); i++){
            if(string.charAt(i) == chr){
                indexesOfChar[indexOfArray] = i;
                indexOfArray++;
            }
        }
        return indexesOfChar;
    }
}
```