```
public class ArrayOps {
    public static void main(String[] args) {
        int[] array = {1,-2,3,-4,5};
        System.out.println(findMissingInt(new int[] {3, 0, 1}));
        System.out.println(secondMaxValue(new int[] {6, 9, 4, 7, 3, 4}));
        System.out.println(containsTheSameElements(new int[] {3, -4, 1, 2, 5}, new int[]
{1, 3, -4, 5}));
        System.out.println(isSorted(new int[] {7, 5, 4, 3, -12}));
    }
    public static int findMissingInt(int[] array) {
        int n = array.length;
        int expectedSum = (n * (n + 1)) / 2;
        int actualSum = 0;
        int missing;
        if (array.length >= 1) {
            for (int i = 0; i < n; i++) {
                actualSum = actualSum + array[i];
            }
            missing = expectedSum - actualSum;
            return missing;
        }
        return -1;
    }
    public static int secondMaxValue(int[] array) {
        int firstMax = Math.max(array[0], array[1]);
        int secondMax = Math.min(array[0], array[1]);
        for (int i = 2; i < array.length; i++) {
            if (array[i] > firstMax) { //we found another max.We update the second max to
be the first
                // max as now we have a greater value and the current value to be the new
max
                secondMax = firstMax;
                firstMax = array[i];
            } else if (array[i] > secondMax) {
                secondMax = array[i]; //as it's not greater than first max but greater than
secondmax,
                // we found a new second max and change its value now
            }
        }
        return secondMax;
    }
    public static boolean containsTheSameElements(int [] array1,int [] array2) {
        boolean result1 = true;
        boolean result2 = true;
```

```
        int count;
        for (int i = 0; i <array1.length; i++) {
            count=0;
            for (int j = 0; j <array2.length; j++) {
                if(array1[i] != array2[j]){
                    count++;
                }
            }
            if(count == array2.length){
                result1 = false;
                break;
            }
        }
        for (int i = 0; i <array2.length; i++) {
            count=0;
            for (int j = 0; j <array1.length; j++) {
                if(array2[i] != array1[j]){
                    count++;
                }
            }
            if(count == array1.length){
                result2 = false;
                break;
            }
        }
        return result1 && result2;
    }
    public static boolean isSorted(int[] array) {
        boolean increase = true;
        boolean decrease = true;
        for (int i = 1; i < array.length; i++) {
            if (array[i - 1] > array[i]) { //we found an element that is greater than the
following one
                increase = false;
                break;
            }
        }
        for (int i = 1; i < array.length; i++) {
            if (array[i - 1] < array[i]) { //we found an element that is smaller than the
following one
                decrease = false;
                break;
            }
        }
```

```
        return increase || decrease;
    }
}
```

```
public class StringOps {
   public static void main(String[] args) {
      System.out.println(capVowelsLowRest("Hello World"));
      System.out.println(capVowelsLowRest("One two tHRee world"));
      System.out.println(capVowelsLowRest("vowels are fun"));
      System.out.println(capVowelsLowRest("intro"));
      System.out.println(capVowelsLowRest("yellow"));
      System.out.println(camelCase("Hello World"));
      System.out.println();
      System.out.println(camelCase(" two     words"));
      System.out.println();
      System.out.println(camelCase("world"));
      System.out.println();
      System.out.println(camelCase("   Intro to coMPUter sCIEncE   "));
      System.out.println(allIndexOf("Hello world",'l'));
      System.out.println(allIndexOf("Hello worLd",'l'));
      System.out.println(allIndexOf("Hello world",'0'));
      System.out.println(allIndexOf("Hello world",' '));
      System.out.println(allIndexOf("MMMM",'M'));
   }
   public static String capVowelsLowRest(String string) {
      String str = "";
      for (int i = 0; i < string.length(); i++) {
         if (string.charAt(i) == 'a' || string.charAt(i) == 'e' || string.charAt(i) == 'i' ||
string.charAt(i) == 'o' || string.charAt(i) == 'u') {
            char chr = (char) (string.charAt(i) - 32);
            str = str + chr;
         } else if ((string.charAt(i) > 'A' && string.charAt(i) <= 'Z') && (string.charAt(i) !=
'A' && string.charAt(i) != 'E' && string.charAt(i) != 'I' && string.charAt(i) != 'O' &&
string.charAt(i) != 'U')) {
            char chr = (char) (string.charAt(i) + 32);
            str = str + chr;
         } else {
            str = str + string.charAt(i);
         }
      }
      return str;
   }
  public static String camelCase(String string) {
      String str = "";
      int indexSpace = 0;
      int numberofLetters = string.length();
      int index=0;
      for (int i = 0; i <string.length()-1 ; i++) {
```

```
            if((int)string.charAt(i)!=32 && (int)string.charAt(i+1)==32 ){
                indexSpace=i+1;
                break;
            }
        }
        for (int j = 0; j < indexSpace; j++) {
            if (string.charAt(j) == ' ') {
                continue;
            }
            if ((int) string.charAt(j) >= 65 && (int) string.charAt(j) <= 90) {
                char chr = (char) ((int) string.charAt(j) + 32);
                str = str + chr;
            } else {
                str = str + string.charAt(j);
            }
        }
        for (int i = indexSpace; i < numberofLetters - 1; i++) {
            if (string.charAt(i) ==' ' && string.charAt(i + 1) != ' ') {
                if ((int) string.charAt(i + 1) >= 65 && (int) string.charAt(i + 1) <= 90) {
                    char chr = (char) ((int) string.charAt(i + 1));
                    str = str + chr;
                }
                else{
                    char chr = (char) ((int) string.charAt(i + 1)-32);
                    str = str + chr;
                }
            }
            if (string.charAt(i) != ' ' && string.charAt(i + 1) != ' ') {
                if ((int) string.charAt(i + 1) >= 65 && (int) string.charAt(i + 1) <= 90) {
                    //     indexSpaceStart = i + 1; //önceki boş sonraki doluysa yeni indeximiz
i+1 dir
                    char chr = (char) ((int) string.charAt(i + 1) + 32);
                    str = str + chr;
                } else {
                    char chr = (char) ((int) string.charAt(i + 1));
                    str = str + chr;
                }
            }
        }
        return str;
    }
    public static int[] allIndexOf(String word, char chr) {
            int count = 0;
            for (int i = 0; i < word.length(); i++) {
```

```
            if (word.charAt(i) == chr) {
                count++;
            }
        }
        int[] array = new int[count];
        int index = 0;
        for (int i = 0; i < word.length(); i++) {
            if (word.charAt(i) == chr) {
                array[index] = i;
                index++;
            }
        }
        return array;
    }
}
```