ArrayOps.java

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

{

} public static int findMissingInt (int [] array)
;int size = array.length
;int sumTotal = 0
;int sumArray = 0

}for(int i = 1 ; i <= size; i++)
;sumTotal = sumTotal + i
{

}for(int i = 0; i < size ; i++)
;sumArray = sumArray + array[i]
{

;return sumTotal - sumArray
{

} public static int secondMaxValue(int [] array)
;int max
;int secondMax

}if(array[0] > array[1])
;[0]max = array
;[1]secondMax = array
```

```
{
}else
;[1]max = array
;[0]secondMax = array
{

}for(int i = 2 ; i < array.length; i++)
}if(array[i] > max)
;secondMax = max
;max = array[i]
{
}else if(array[i] > secondMax)
;secondMax = array[i]
{
{

;return secondMax
{

} public static boolean containsTheSameElements(int [] array1,int [] array2)
return containsOneWay(array1,array2) &&
;containsOneWay(array2,array1)
{
} public static boolean containsOneWay(int [] array1,int [] array2)
;boolean sameElements = false

}for(int i = 0; i < array1.length; i++)
;sameElements = false
}for(int j = 0; j < array2.length; j++)
}if(array1[i] == array2[j])
;sameElements = true
```

```
;break
{
{
}if (sameElements == false)
;return  false
{
{

;return true
{

} public static boolean isSorted(int [] array)
;return isSortedUp(array) || isSortedDown(array)
{
} public static boolean isSortedUp(int [] array)

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

{
;return true
{

} public static boolean isSortedDown(int [] array)

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

```
{

{
;return true
{

{
```

StringOps

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

{



} public static String capVowelsLowRest (String string)
;"" = String newString
;"String eitan = "aeiou
}for (int i = 0; i < string.length(); i++)
;char letter = string.charAt(i)
} if (eitan.indexOf(toLower(letter)) != -1)
;newString += toUpper(letter)
{
} else
;newString += toLower(letter)
{
{
;return newString
{
;"private static String lowerCaseLetters = "abcdefghijklmnopqrstuvwxyz
private static String upperCaseLetters =
;""ABCDEFGHIJKLMNOPQRSTUVWXYZ

} private static char toUpper (char ch)
;int index = lowerCaseLetters.indexOf(ch)
}if (index == -1)
;return ch
{
} else
```

```
;return upperCaseLetters.charAt(index)
{

{
} private static char toLower (char ch)
;int index = upperCaseLetters.indexOf(ch)
}if (index == -1)
;return ch
{
} else
;return lowerCaseLetters.charAt(index)
{

{

} public static String camelCase (String string)
;"" = String newString
;boolean afterSpace = false
;int i = 0
}while (i < string.length() && string.charAt(i) == ' ')
;++i
{
}for (;i < string.length(); i++)
;char letter = string.charAt(i)
} if (letter == ' ')
;afterSpace = true
{
}else
}if (afterSpace)
;newString += toUpper(letter)
```

```
{
} else
;newString += toLower(letter)
{
;afterSpace = false
{

{
;return newString
{

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