Testing and JUnit

Assignment 3

**Note :**

* Please write proper specifications for all of the methods covered in this assignment.
* Write both positive and negative testcases for these methods.
* Write parameterized test cases wherever applicable.

**Question**

Design a class ArrOperation which should support following methods. Assume each method will receive an array of positive integers.

1. **Return the size of the largest mirror section found in the input array.**

Mirror section in an array is a group of contiguous elements such that somewhere in the array, the same group appears in reverse order. For example

maxMirror([1, 2, 3, 8, 9, 3, 2, 1]) → 3

maxMirror([7, 1, 4, 9, 7, 4, 1]) → 2

maxMirror([1, 2, 1, 4]) → 3

maxMirror\*([1, 4, 5, 3, 5, 4, 1]) → 7

Throw assertion error in case array is empty.

**2**. **Return the number of clumps in the input array.**

Clump in an array is a series of 2 or more adjacent elements of the same value. For example

countClumps([1, 2, 2, 3, 4, 4]) → 2

countClumps([1, 1, 2, 1, 1]) → 2

countClumps([1, 1, 1, 1, 1]) → 1

Throw assertion error in case array is empty.

**3.** **Solve fixXY problem**

Return an array that contains exactly the same numbers as the input array, but rearranged so that every X is immediately followed by a Y. Do not move X within array, but every other number may move. For ex:

Let’s say value of X is 4 and Y is 5. Then

fixXY([5, 4, 9, 4, 9, 5]) → [9, 4, 5, 4, 5, 9]

fixXY([1, 4, 1, 5]) → [1, 4, 5, 1]

fixXY([1, 4, 1, 5, 5, 4, 1]) → [1, 4, 5, 1, 1, 4, 5]

Assume we are receiving an array of integers along with value of X and Y as a method parameter.

Throw assertion in following cases :

* If array is empty
* If there are unequal numbers of X and Y in input array.
* If two adjacents X values are there.
* If X occurs at the last index of array.

**4.** **splitArray**

Return the index if there is a place to split the input array so that the sum of the numbers on one side is equal to the sum of the numbers on the other side else return -1. For ex

splitArray([1, 1, 1, 2, 1]) → 3

splitArray([2, 1, 1, 2, 1]) → -1

splitArray([10, 10]) → 1

Throw assertion error in case array is empty.

**Add-on Exercise**

Write Junit test cases for all the previous assignments done so far.

* Include both positive and negative test cases for each problem using JUnit concepts.
* Also use concept of parameterized test cases wherever applicable.