**Instructions for students:**

Complete the following methods on Arrays

You may use any language to complete the tasks.

All your methods must be written in one single .java or .py file. DO NOT CREATE separate files for each task.

If you are using JAVA, you must include the main method as well which should test your other methods and print the outputs according to the tasks.

If you are using PYTHON, then follow the coding templated shared in this folder.

## Linear Arrays

**NOTE: Each method carries 5 marks**

1. **Shift Left k Cells**

Consider an array named source. Write a method/function named shiftLeft( source, k)

that shifts all the elements of the source array to the left by 'k' positions. Finally, return the array.

source=[10,20,30,40,50,60]

shiftLeft(source,3)

After calling shiftLeft(source,3), printing the array should give the output as:

40, 50, 60, 0, 0, 0 ]

2. **Rotate Left k cells**

Consider an array named source. Write a method/function named rotateLeft( source, k) that rotates all the elements of the source array to the left by 'k' positions. Finally, return the array.

Example:source=[10,20,30,40,50,60]

rotateLeft(source,3)

After calling rotateLeft(source,3), printing the array should give the output as:

[ 40, 50, 60, 10, 20, 30]

3. **Shift Right k Cells**

Consider an array named source. Write a method/function named shifRight( source, k)

that shifts all the elements of the source array to the right by 'k' positions. Finally, return the array.

Example:

source=[10,20,30,40,50,60]

shiftRight(source,3)

After calling shiftRight(source,3), printing the array should give the output as:

0,0,0,10,20,30 ]

4. **Rotate Right k cells**

Consider an array named source. Write a method/function named rotateRight( source,

that rotates all the elements of the source array to the right by 'k' positions. Finally, return the array.

Example:

source=[10,20,30,40,50,60]

rotateRight(source,3)

After calling rotateRight(source,3), printing the array should give the output as:
[ 40, 50, 60, 10, 20, 30]

5. **Remove an element from an array**

Consider an array named source. Write a method/function named remove( source, idx) that removes the element in the index **idx** of the source array. Finally, return the array.

Example:

source=[10,20,30,40,50,0,0]

remove(source,2)

After calling remove(source,2) , printing the array should give the output as:

10,20,40,50,0,0,0]

6. **Remove all occurrences of a particular element from an array**

Consider an array named source. Write a method/function named removeAll( source, element) that removes all the occurrences of the given element in the source array. Finally, return the array.

Example:

source=[10,2,30,2,50,2,2,0,0]

removeAll(source,2)

After calling removeAll(source,2), all the occurrences of 2 must be removed. Printing the array afterward should give the output as:
10,30,50,0,0,0,0,0,0]

7. **Splitting an Array**

Suppose the elements of an array A containing positive integers, denote the weights in kilograms. And we have a beam balance. We want to put the weights on both pans of the balance in such a way that for some index 0 < i < A.length - 1, all values starting from A[0], A[1], upto A[ i - 1], should be on the left pan. And all values starting from A[ i ] upto A[ A.length - 1] should be on the right pan and the left and right pan should be balanced. If such i exists, return true. Else, return false.

Input: [1, 1, 1, 2, 1]  Output: true

Explanation: (summation of [1, 1, 1] = summation of [2,1])

Input: [2, 1, 1, 2, 1] Output: false

Input: [10, 3, 1, 2, 10] Output: true

Explanation: (summation of [10, 3] = summation of [1,2,10]))

8. **Max Bunch Count**

A "bunch" in an array is a consecutive chain of two or more adjacent elements of the same value. Write a method that returns the number of elements in the largest bunch found in the given array.

Input: [1, 2, 2, 3, 4, 4, 4]  Output: 3

Explanation: There are two bunches here {2,2} and {4,4,4}. The largest bunch is {4,4,4} containing 3 elements so 3 is returned.

Input: [1,1,2, 2, 1, 1,1,1] Output:4

Explanation: There are three bunches here {1,1} and {2,2} and {1,1,1,1}. The largest bunch is {1,1,1,1} containing 4 elements so 4 is returned.