Assignment 01

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. You must execute the method by passing an array and number of cells to be shifted. After calling the method, print the array to show whether the elements have been shifted properly.

Example:

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. You must execute the method by passing an array and number of cells to be shifted. After calling the method, print the array to show whether the elements have been shifted properly.

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. Remove an element from an array

Consider an array named source. Write a method/function named remove (source, size, idx) that removes the element in index idx of the source array. You must execute the method by passing an array, its size and the idx (that is the index of the element to be removed). After calling the method, print the array to show whether the element of that particular index has been removed properly.

Example:

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

remove(source,5,2)

After calling remove(source,5,2), printing the array should give the output as:

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

4. Remove all occurrences of a particular element from an array

Consider an array named source. Write a method/function named removeAll(source, size, element) that removes all the occurrences of the given element in the source array. You must execute the method by passing an array, its size and the element to be removed. After calling the method, print the array to show whether all the occurrences of the element have been removed properly.

Example:

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

removeAll(source,7,2)

After calling removeAll(source,7,2), all the occurrences of 2 must be removed. Printing the array afterwards should give the output as:

[10,30,50,0,0,0,0,0,0]

5. 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], up to A [i - 1], should be on the left pan. And all values starting from A [i] up to A [a. length - 1] should be on the right pan and the left and right pan should be balanced. If such an 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]))

6. Array series

Write a method that takes an integer value n as a parameter. Inside the method, you should create an array of length n squared (n*n) and fill the array with the following pattern. Return the array at the end and print it.

If,

n=2: {0,1, 2,1} (spaces have been added to show two distinct groups).

n=3: {0, 0, 1, 0, 2, 1, 3, 2, 1} ((spaces have been added to show three distinct groups).

n=4: {0, 0, 0, 1, 0, 0, 2, 1, 0, 3, 2, 1, 4, 3, 2, 1} (spaces have been added to show four distinct groups).

7. 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.

8. Repetition

Write a method that takes in an array as a parameter and counts the repetition of each element. That is, if an element has appeared in the array more than once, then its 'repetition' is its number of occurrences. The method returns true if there are at least two elements with the same number of 'repetitions. Otherwise, return false.

Input: {4,5,6,6,4,3,6,4} Output: True

Explanation: Two numbers repeat in this array: 4 and 6. 4 has a repetition of 3, 6 has a repetition of 3. Since two numbers have the same repetition output is True.

Input: {3,4,6,3,4,7,4,6,8,6,6} Output: False

Explanation: Three numbers repeat in this array:3,4 and 6 .3 has a repetition of 2, 4 has a repetition of 3, 6 has a repetition of 4. Since no two numbers have the same repetition output is False.

Circular Array

9. Palindrome

Write a method/function that takes in a circular array, its size and start index and finds whether the elements in the array form a palindrome or not. Return true if the elements form a palindrome, otherwise, return false.

Example:

Input: [20,10,0,0,0,10,20,30] (start =5, size=5) Output: True.

Input: [10,20,0,0,0,10,20,30] (start =5, size=5) Output: False

10. Intersection

Write a method/function that takes two circular arrays containing distinct integer values, their sizes and start indexes and returns a linear array containing the common elements between the two circular arrays.

Input:

Circular array 1: [40,50,0,0,0,10,20,30] (start_1 =5, size_1 =5)

Circular array 2: [10,20,5,0,0,0,0,0,55,40,15,25] (start_2 =8, size_2 =7)

Output: [10,20,40]