***Move Zeros to End***

Given an array of **n** numbers. The problem is to move all the 0’s to the end of the array while maintaining the order of the other elements. Only single traversal of the array is required.  
Examples: 

Input : arr[] = {1, 2, 0, 0, 0, 3, 6}

Output : 1 2 3 6 0 0 0

Input: arr[] = {0, 1, 9, 8, 4, 0, 0, 2, 7, 0, 6, 0, 9}

Output: 1 9 8 4 2 7 6 9 0 0 0 0 0

**Algorithm:** 

moveZerosToEnd(arr, n)

Initialize count = 0

for i = 0 to n-1

if (arr[i] != 0) then

arr[count++]=arr[i]

for i = count to n-1

arr[i] = 0

C++Java

// Java implementation to move

// all zeroes at the end of array

import java.io.\*;

class GFG {

// function to move all zeroes at

// the end of array

static void moveZerosToEnd(int arr[], int n) {

// Count of non-zero elements

int count = 0;

// Traverse the array. If arr[i] is non-zero, then

// update the value of arr at index count to arr[i]

for (int i = 0; i < n; i++)

if (arr[i] != 0)

arr[count++] = arr[i];

// Update all elements at index >=count with value 0

for (int i = count; i<n;i++)

arr[i]=0;

}

// function to print the array elements

static void printArray(int arr[], int n) {

for (int i = 0; i < n; i++)

System.out.print(arr[i] + " ");

}

// Driver program to test above

public static void main(String args[]) {

int arr[] = {0, 1, 9, 8, 4, 0, 0, 2,

7, 0, 6, 0, 9};

int n = arr.length;

System.out.print("Original array: ");

printArray(arr, n);

moveZerosToEnd(arr, n);

System.out.print("\nModified array: ");

printArray(arr, n);

}

}

**Output**

Original array: 0 1 9 8 4 0 0 2 7 0 6 0 9

Modified array: 1 9 8 4 2 7 6 9 0 0 0 0 0

Time Complexity: O(n).   
Auxiliary Space: O(1).