# 422. Find Duplicates

## Description

Given an integer array nums of length n where all the integers of nums are in the range [1, n] and each integer appears at most twice, return an array of all the integers that appears twice.

You must write an algorithm that runs in O(n) time and uses only constant auxiliary space, excluding the space needed to store the output

## Code

import java.util.Arrays;  
  
class FindDuplicates {  
 public static void main(String args[]) {  
 int arr[] = {1, 1, 2};  
  
 FindDuplicates fd = new FindDuplicates();  
 System.out.println(Arrays.toString(fd.findDuplicates(arr)));  
 }  
  
 int[] findDuplicates(int[] arr) {  
 int n = arr.length;  
 int max = 0;  
 int result[] = new int[n];  
 int ri = 0;  
  
 for (int i=0; i<n; ++i) {  
 if (arr[i] > max)   
 max = arr[i];  
 }  
  
 int map[] = new int[max+1];  
  
 for (int i=0; i<n; ++i) {  
 map[arr[i]]++;  
 }  
   
 for (int i=0; i<n; ++i) {  
 if (map[i] == 2) {  
 result[ri++] = i;  
 }  
 }  
 return Arrays.copyOfRange(result, 0, ri);  
 }  
}

## Output

[1]