### Remove duplicate elements from sorted Array 7 tiimes asked

Given a sorted array**A[]** of size **N**, delete all the duplicated elements from A[]. Modify the array such that if there are X distinct elements in it then the first X positions of the array should be filled with them in increasing order and return the number of distinct elements in the array.  
  
**Note:**  
1.Don't use set or HashMap to solve the problem.  
2. You must return the number of distinct elements(X) in the array, the generated output will print all the elements of the modified array from index 0 to X-1.  
  
**Example 1:**

**Input:**

N = 5

Array = {2, 2, 2, 2, 2}

**Output:** 1

**Explanation:** After removing all the duplicates

only one instance of 2 will remain i.e. {2}.

**Example 2:**

**Input:**

N = 3

Array = {1, 2, 2}

**Output:** 2  
**Explation:**After removing all duplicates modify array will be {1,2} and lenght is 2

### Java code

//{ Driver Code Starts

//Initial Template for Java

import java.io.\*;

import java.util.\*;

public class CodingMaxima {

public static void main(String[] args) throws Exception {

Scanner sc = new Scanner(System.in);

int T = sc.nextInt();

while(T>0)

{

int N = sc.nextInt();

int a[] = new int[N];

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

a[i] = sc.nextInt();

Solution g = new Solution();

int n = g.remove\_duplicate(a,N);

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

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

System.out.println();

T--;

}

}

}

// } Driver Code Ends

class Solution {

int remove\_duplicate(int A[],int n){

// code here

int j = 0;

for (int i = 0; i < n - 1; i++) {

if (A[i] != A[i + 1]) {

A[j++] = A[i];

}

}

A[j++] = A[n - 1];

return j;

}

}