**Find duplicates in an array: -**

Easy Accuracy: 18.95% Submissions: 525K+ Points: 2

Given an array **a** of size **N** which contains elements from **0** to **N-1**, you need to find all the elements occurring more than once in the given array. Return the answer in ascending order. If no such element is found, return list containing **[-1]**.

**Note:** The extra space is only for the array to be returned. Try and perform all operations within the provided array.

**Example 1:**

**Input:**

N = 4

a[] = {0,3,1,2}

**Output:**-1

**Explanation:**There is no repeating element in the array. Therefore output is -1.

**Example 2:**

**Input:**

N = 5

a[] = {2,3,1,2,3}

**Output:**2 3

**Explanation:**2 and 3 occur more than once in the given array.

**Your Task:**  
Complete the function **duplicates()** which takes array a[] and n as input as parameters and returns a list of elements that occur more than once in the given array in a sorted manner.

**Expected Time Complexity:** O(n).  
**Expected Auxiliary Space:** O(n).

**Constraints:**  
1 <= N <= 105  
0 <= A[i] <= N-1, for each valid i

**Code: -**

//{ Driver Code Starts

#include <bits/stdc++.h>

using namespace std;

// } Driver Code Ends

class Solution{

public:

vector<int> duplicates(int arr[], int n) {

// code here

vector<int> ans;

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

arr[ (arr[i] % n) ] += n;

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

if(arr[i] / n > 1)

ans.push\_back(i);

}

if(ans.empty())

return {-1};

return ans;

}

};

//{ Driver Code Starts.

int main() {

int t;

cin >> t;

while (t-- > 0) {

int n;

cin >> n;

int a[n];

for (int i = 0; i < n; i++) cin >> a[i];

Solution obj;

vector<int> ans = obj.duplicates(a, n);

for (int i : ans) cout << i << ' ';

cout << endl;

}

return 0;

}

// } Driver Code Ends

**T.C: - O(N)**

**S.C; - O(1) excluding answer storage**