



Dash



All



Articles



Videos



Problems



Quiz



Contest

<< Prev

Next >>

Frequencies of array elements

Given an array which may contain duplicates, print all elements and their frequencies.

Examples:

Input : arr[] = {10, 20, 20, 10, 10, 20, 5, 20}

Output : 10 3

20 4

5 1

Input : arr[] = {10, 20, 20}

Output : 10 1

20 2

A **simple solution** is to run two loops. For every item count number of times, it occurs. To avoid duplicate printing, keep track of processed items.

C++

Java

```
// Java program to count frequencies of array items
import java.util.Arrays;
```



Dash



All



Articles



Videos



Problems



Quiz



Contest



```
class GFG
{
    public static void countFreq(int arr[], int n)
    {
        boolean visited[] = new boolean[n];

        Arrays.fill(visited, false);

        // Traverse through array elements and
        // count frequencies
        for (int i = 0; i < n; i++) {

            // Skip this element if already processed
            if (visited[i] == true)
                continue;

            // Count frequency
            int count = 1;
            for (int j = i + 1; j < n; j++) {
                if (arr[i] == arr[j]) {
                    visited[j] = true;
                    count++;
                }
            }
            System.out.println(arr[i] + " " + count);
        }
    }
}
```





Dash



All



Articles



Videos



Problems



Quiz



Contest



```
// Driver code
public static void main(String []args)
{
    int arr[] = new int[]{ 10, 20, 20, 10, 10, 20, 5, 20 };
    int n = arr.length;
    countFreq(arr, n);
}
}
```

// This code contributed by Adarsh_Verma.



Output

```
10 3
20 4
5 1
```

Time Complexity : $O(n^2)$

Auxiliary Space : $O(n)$

An **efficient solution** is to use hashing.

C++

Java



Dash



All



Articles



Videos



Problems



Quiz



Contest



```
// Java program to count frequencies of array items
import java.util.*;

class GFG
{
    static void countFreq(int arr[], int n)
    {
        Map<Integer, Integer> mp = new HashMap<>();

        // Traverse through array elements and
        // count frequencies
        for (int i = 0; i < n; i++)
        {
            if (mp.containsKey(arr[i]))
            {
                mp.put(arr[i], mp.get(arr[i]) + 1);
            }
            else
            {
                mp.put(arr[i], 1);
            }
        }
        // Traverse through map and print frequencies
        for (Map.Entry<Integer, Integer> entry : mp.entrySet())
        {
```





Dash



All



Articles



Videos



Problems



Quiz



Contest



```
        System.out.println(entry.getKey() + " " + entry.getValue());
    }
}

// Driver code
public static void main(String args[])
{
    int arr[] = {10, 20, 20, 10, 10, 20, 5, 20};
    int n = arr.length;
    countFreq(arr, n);
}

// This code contributed by Rajput-Ji
```

Output

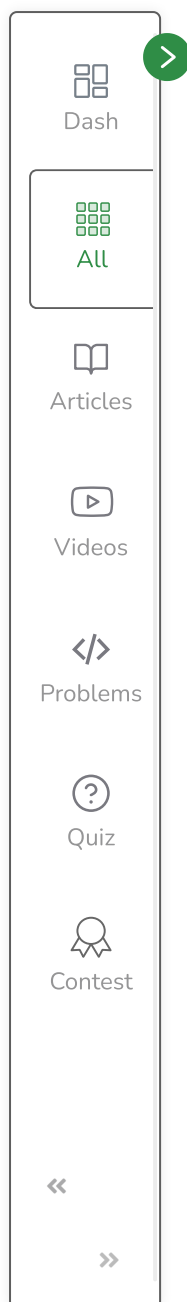
```
5 1
10 3
20 4
```

Time Complexity : $O(n)$

Auxiliary Space : $O(n)$



In above efficient solution, how to print elements in same order as they appear in input?



C++

Java

```
// Java program to count frequencies of array items
import java.util.*;

class GFG
{
    static void countFreq(int arr[], int n)
    {
        Map<Integer, Integer> mp = new HashMap<>();

        // Traverse through array elements and
        // count frequencies
        for (int i = 0; i < n; i++)
        {
            mp.put(arr[i], mp.get(arr[i]) == null ? 1 : mp.get(arr[i]) + 1);
        }

        // To print elements according to first
        // occurrence, traverse array one more time
        // print frequencies of elements and mark
        // frequencies as -1 so that same element
        // is not printed multiple times.
```





Dash



All



Articles



Videos



Problems



Quiz



Contest



```
for (int i = 0; i < n; i++)
{
    if (mp.get(arr[i]) != -1)
    {
        System.out.println(arr[i] + " " + mp.get(arr[i]));
        mp.put(arr[i], -1);
    }
}

// Driver code
public static void main(String[] args)
{
    int arr[] = {10, 20, 20, 10, 10, 20, 5, 20};
    int n = arr.length;
    countFreq(arr, n);
}
```

// This code contributed by Rajput-Ji

Output

```
10 3
20 4
```





Dash



All



Articles



Videos



Problems



Quiz



Contest

**Time Complexity :** $O(n)$ **Auxiliary Space :** $O(n)$

This problem can be solved in Java using Hashmap. Below is the program.

C++

Java

```
// Java program to count frequencies of
// integers in array using Hashmap
import java.io.*;
import java.util.*;
class OccurenceOfNumberInArray {
    static void frequencyNumber(int arr[], int size)
    {
        // Creating a HashMap containing integer
        // as a key and occurrences as a value
        HashMap<Integer, Integer> freqMap
            = new HashMap<Integer, Integer>();

        for (int i=0;i<size;i++) {
            if (freqMap.containsKey(arr[i])) {

                // If number is present in freqMap,
```





Dash



All



Articles

Courses

Tutorials

Jobs

Practice

Upcoming
Contests

Videos



Problems



Quiz



Contest



```
// incrementing it's count by 1
freqMap.put(arr[i], freqMap.get(arr[i]) + 1);
}
else {

// If integer is not present in freqMap,
// putting this integer to freqMap with 1 as it's value
freqMap.put(arr[i], 1);

}
}
```

```
System.out.println(entry.getKey() + " " + entry.getValue());
}
}

// Driver Code
public static void main(String[] args)
{
    int arr[] = {10, 20, 20, 10, 10, 20, 5, 20};
    int size = arr.length;
    frequencyNumber(arr, size);
}
}
```



Output

```
5 1
10 3
20 4
```



Mark as Read

 Report An Issue

If you are facing any issue on this page. Please let us know.



Dash



All



Articles



Videos



Problems



Quiz



Contest

