# C program to find the frequency of each element in the array

## Frequency of Element in C

Here, in this page we will discuss the program to find the frequency of element in C programming language. We are given with an array and need to print the frequency of each given element.

### Example

**Input** : arr[6] = [1, 2, 2, 3, 1, 2]
**Output** : 1 occurs 2 times
2 occurs 3 times
3 occurs 1 times

## Method Discussed :

- Method 1 : Using Extra Space
- Method 2 : Naive approach without extra space.
- Method 3 : Using Sorting Technique.

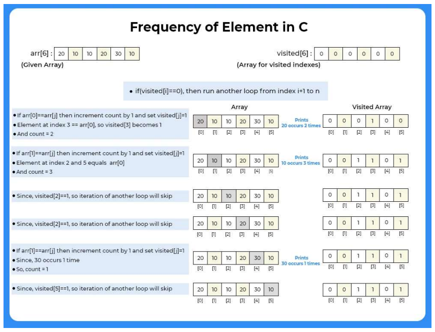
## Method 1:

In this method we will count the frequency of each elements using two for loops.

- To check the status of visited elements create a array of size n.
- Run a loop from index 0 to n and check if (visited[i]==1) then skip that element.
- Otherwise create a variable count = 1 to keep the count of frequency.
- Run a loop from index i+1 to n
- Check if(arr[i]==arr[j]), then increment the count by 1 and set visited[j]=1.
- After complete iteration of for loop print element along with value of count.

## Time and Space Complexity :

- Time Complexity : **O($n^2$)**
- Space Complexity : **O(n)**



- if(visited[i]==0), then run another loop from index i+1 to n

- If arr[0]==arr[j] then increment count by 1 and set visited[j]=1
- Element at index 3 == arr[0], so visited[3] becomes 1
- And count = 2

- If arr[1]==arr[j] then increment count by 1 and set visited[j]=1
- Element at index 2 and 5 equals arr[0]
- And count = 3

- Since, visited[2]==1, so iteration of another loop will skip

- Since, visited[2]==1, so iteration of another loop will skip

- If arr[1]==arr[j] then increment count by 1 and set visited[j]=1
- Since, 30 occurs 1 time
- So, count = 1

- Since, visited[5]==1, so iteration of another loop will skip

### Code in C

Run

```
#include<stdio.h>

// Main function to run the program
int main()
{
    int arr[] = {10, 30, 10, 20, 10, 20, 30, 10};
    int n = sizeof(arr)/sizeof(arr[0]);

    int visited[n];

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

      if(visited[i]==0){
        int count = 1;
        for(int j=i+1; j<n; j++){
          if(arr[i]==arr[j]){
            count++;
            visited[j]=1;
          }
        }

        printf("%d occurs %d times\n", arr[i],
count);
      }
    }

   return 0;
```

Related Pages

Sort first half in ascending order and second half in descending

Sort the elements of an array

Sorting elements of an array by frequency

Finding the Longest Palindrome in an Array

Counting Distinct Elements in an Array

Prime Course Trailer

Related Banners

Output

```
10 occurs 4 times
30 occurs 2 times
20 occurs 2 times
```

## Method 2 :

In this method we will use the naive way to find the frequency of elements in the given integer array without using any extra space.

Method 2 : Code in C

Run

```
#include<stdio.h>

void countFrequency(int *arr, int size){

    for (int i = 0; i < size; i++){
        int flag = 0;
        int count = 0;

        // Counting of any element has to be delayed
to its last occurrence
        for (int j = i+1; j < size; j++){
            if (arr[i] == arr[j]){
                flag = 1;
                break;
            }
        }

        // The continue keyword is used to end the
current iteration
        // in a for loop (or a while loop), and
continues to the next iteration
        if (flag == 1)
            continue;

        for(int j = 0;j<=i;j++){
            if(arr[i]==arr[j])
                count +=1;
        }

        printf("%d : %d\n", arr[i], count);
    }
}

int main()
{
    int arr[] = {5, 8, 5, 7, 8, 10};
    int size = sizeof(arr)/sizeof(arr[0]);

    countFrequency(arr, size);

    return 0;
}
```

Output

```
5 : 2
7 : 1
8 : 2
10 : 1
```

## Method 3 :

In this method we will sort the array then, count the frequency of the elements.

### Time and Space Complexity :

- Time Complexity : **O(nlogn)**
- Space Complexity : **O(1)**

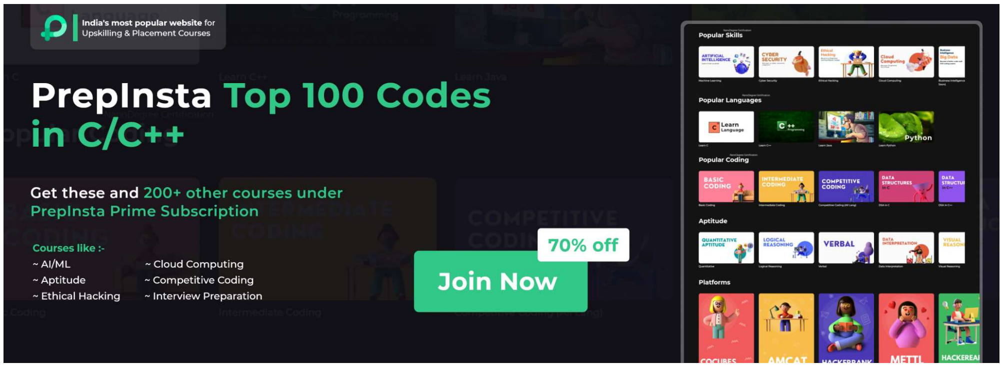
### Method 3 : Code in C

```
#include<stdio.h>

void countDistinct(int arr[], int n)
{
    //Sorting of the array
    for(int i=0; i<n; i++){
        for(int j=i+1; j<n; j++){ if(arr[i]>arr[j]){
                int temp = arr[i];
                arr[i] = arr[j];
                arr[j] = temp;
            }
        }
    }

    // Traverse the sorted array
    for (int i = 0; i < n; i++){
        int count = 1;

        // Move the index ahead whenever
        // you encounter duplicates
        while (i < n - 1 && arr[i] == arr[i + 1]){
            i++;
            count++;
        }

        printf("%d : %d\n", arr[i], count);
    }

}

// Driver program to test above function
int main()
{
    int arr[] = {5, 8, 5, 7, 8, 10};
    int n = sizeof(arr) / sizeof(arr[0]);
    countDistinct(arr, n);
    return 0;
}
```

Run

### Output

```
5 : 2
7 : 1
8 : 2
10 : 1
```


India's most popular website for Upskilling & Placement Courses
PrepInsta Top 100 Codes in C/C++
Get these and 200+ other courses under PrepInsta Prime Subscription
Courses like :-
~ AI/ML
~ Aptitude
~ Ethical Hacking
~ Cloud Computing
~ Competitive Coding
~ Interview Preparation
70% off
Join Now

**Login/Signup** to comment

Isshita Ghosh_CSE_023 In method 1, when I am giving (10,20,30,20,50,50,40) The output is :

20 occurs 2 times

50 occurs 2 times

0 Log in to Reply

PrepInsta

PrepInsta.com

No.1 and most visited website for Placements in India.

We help students to prepare for placements with the best study material, online classes, Sectional Statistics for better focus and Success stories & tips by Toppers on PrepInsta.

© 2022 Prep Insta

**Support**

Contact Us
About Us
Refund Policy
Privacy Policy
Services
Disclaimer
Terms and Conditions

**Companies**

Accenture
Cognizant
MindTree
VMware
CapGemini
Deloitte
Wipro
Microsoft
TCS
Infosys
Oracle
HCL
TCS Ninja
IBM

**All Exams Dashboards**

CoCubes Dashboard
eLitmus Dashboard
HirePro Dashboard
MeritTrac Dashboard
Mettl Dashboard
DevSquare Dashboard

**Get In Touch**

Instagram
Linkedin
Youtube
Telegram
facebook
Twitter

**Get In Touch**

support@prepinsta.com
+91-8448440710
Text us on
Whatsapp/Instagram