"Experiment 1.3"

Student Name: **SUMIT KUMAR**
Branch: **CSE**
Semester: **5**
Subject Name: **Design and Analysis of Algorithms Lab**

UID: **20BCS8226**
Section/Group: **808-A**
Date of Performance: **18-08-22**
Subject Code: **20CSP-312**

1. Aim/Overview of the practical:

In O(n) time complexity, find the frequency of elements in a given array.

2. Algorithm/Flowchart (For programming based labs):

Step 1: Input the number of elements of an array.

Step 2: Input the array elements.

Step 3: Create another array to store the frequency of elements.

Step 4: Traverse the input array and update the count of the elements in the frequency array.

Step 5: Print the frequency array which displays the frequency of all the elements of the array.

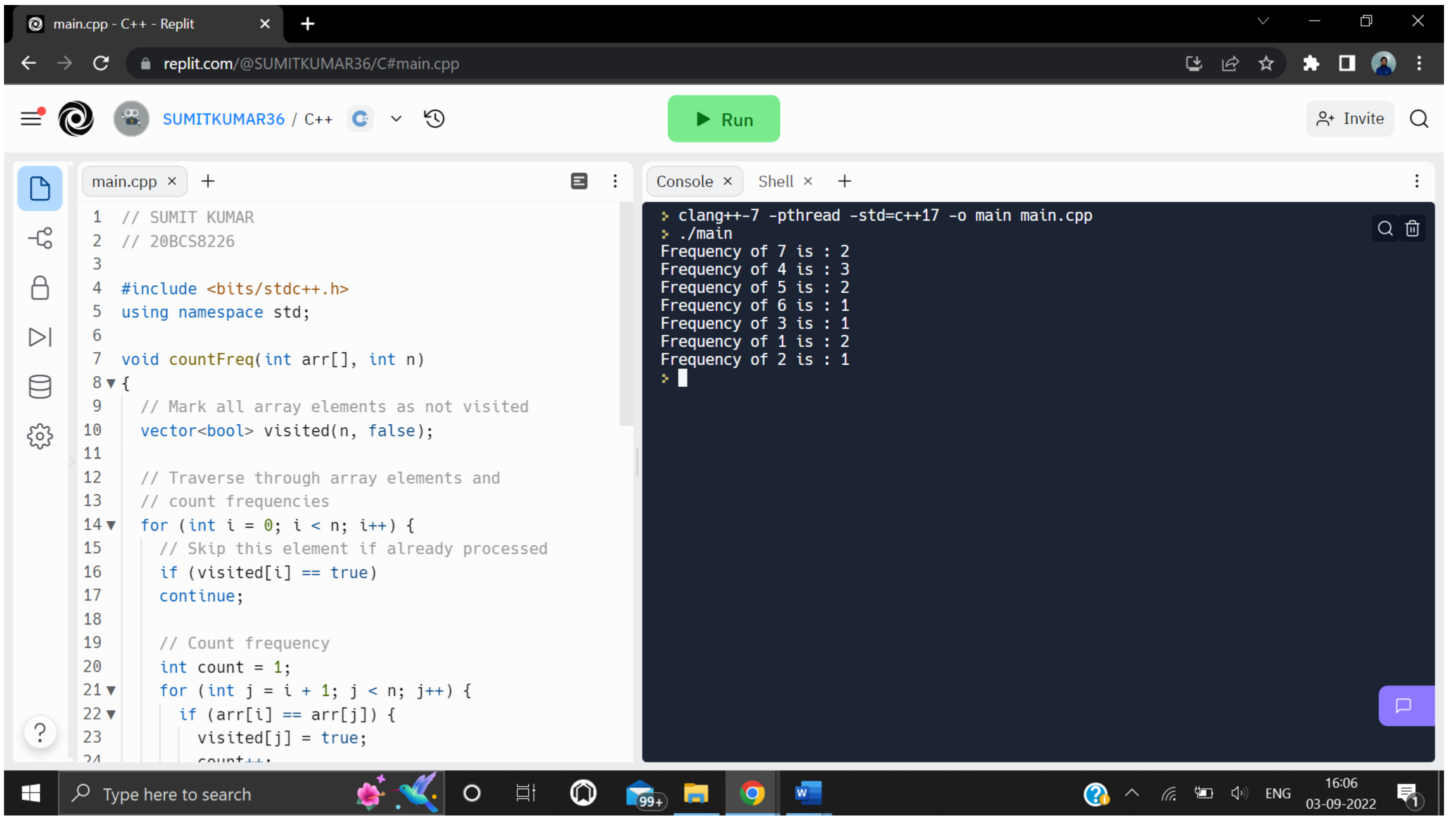
3. Steps for experiment/practical/Code:

```
// SUMIT KUMAR
// 20BCS8226

#include <bits/stdc++.h>
using namespace std;

void countFreq(int arr[], int n)
{
  // Mark all array elements as not visited
  vector<bool> visited(n, 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++;
      }
    }
    cout<<"Frequency of "<<arr[i]<< " is : "<<count<<endl;
  }
}

int main()
{
  int arr[] = { 7, 7, 4, 5, 6, 5, 4, 4, 3, 1, 2, 1};
  int n = sizeof(arr) / sizeof(arr[0]);
  countFreq(arr, n);

  return 0;
}
```

4. Observations/Discussions/ Complexity Analysis:

This approach will result in linear complexity, i.e., O(n) time complexity. Here, n is the number of elements present in the given array.

5. Result/Output/Writing Summary:



```cpp
// SUMIT KUMAR
// 20BCS8226

#include <bits/stdc++.h>
using namespace std;

void countFreq(int arr[], int n)
{
  // Mark all array elements as not visited
  vector<bool> visited(n, 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;
```

```
> clang++-7 -pthread -std=c++17 -o main main.cpp
> ./main
Frequency of 7 is : 2
Frequency of 4 is : 3
Frequency of 5 is : 2
Frequency of 6 is : 1
Frequency of 3 is : 1
Frequency of 1 is : 2
Frequency of 2 is : 1
>
```

Learning outcomes (What I have learnt):

1. Learnt about a way of calculating frequency of each element.

2. Learnt how to implement know frequency in linear time complexity.

3. Learnt faster method of calculating frequency of elements in an array.

Evaluation Grid (To be created as per the SOP and Assessment guidelines by the faculty):

Sr. No.	Parameters	Marks Obtained	Maximum Marks
1.			
2.			
3.			