Experiment - 3

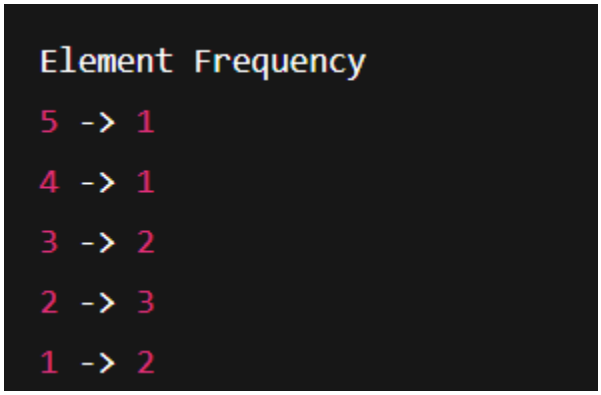
Student Name: Garvi Dabas
UID: 23BCS11346
Branch: BE-CSE
Section/Group: KRG-2B
Semester: 5th
Date of Performance: 30/7/25
Subject Name: DAA
Subject Code: 23CSH-301

1. Aim:
To write a C++ program that finds the frequency of each element in a given array in **O(n)** time complexity.

2. Procedure:
- Start the program and include necessary headers.
- Take an input array of size `n`.
- Use an auxiliary data structure (map or unordered_map) to store frequencies.
- Traverse the array once and update counts in the map for each element.
- Traverse the map to display each element along with its frequency.

3. Code:
```
#include <iostream>
#include <unordered_map>
using namespace std;

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

    unordered_map<int, int> freq;

    for (int i = 0; i < n; i++) {
        freq[arr[i]]++;
    }

    cout << "Element Frequency" << endl;
    for (auto it : freq) {
        cout << it.first << " -> " << it.second << endl;
    }

    return 0;
}
```

4. Output:



5. Learning Outcomes:

- Learnt how to compute frequencies of elements in linear **O(n)** time.
- Gained understanding of using **hash maps (unordered_map)** for efficient counting.
- Understood the difference between naive `O(n^2)` approaches and optimized `O(n)` solutions.
- Practised iterating over arrays and associative containers in C++.
- Developed confidence in applying hashing techniques to real-world problems.