EXPERIMENT 1.3

## AIM: Evaluate the complexity of the developed program to find frequency of elements in a given array. (Using hash maps)

**Objectives: To understand the concept of Hash maps and Arrays.**

**Tools/Resource used:** VS Code

# Procedure/Algorithm:

1. Define a function named findFrequency that takes an array (arr[]) and its size (n) as parameters.
   * Create an unordered\_map named mp to store element frequencies.
2. Iterate through the array using a loop with index i ranging from 0 to n-1:
   * Increment the frequency of arr[i] in the mp map.
3. Iterate through the elements in the mp map:
   * Print the element and its corresponding frequency.
4. End of the findFrequency function.
5. In the main function:
   * Declare an integer array named arr and initialize it with elements.
   * Calculate the size of the array (n) using sizeof(arr) / sizeof(arr[0]).
   * Call the findFrequency function with arr[] and n as arguments.
6. End of the program.

**Program Code:** #include <bits/stdc++.h> using namespace std;

void findFrequency(int arr[], int n)

{

unordered\_map<int, int> mp; for (int i = 0; i < n; i++) {

mp[arr[i]]++;

}

for (auto i : mp) {

cout << "Element " << i.first << " occurs "

<< i.second << " times" << endl;

}

}

int main()

{

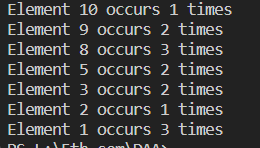
cout << "Shivam Kumar\nUID - 21BCS2124\n"; int arr[] = { 1, 1, 1, 2, 3, 3, 5, 5, 8, 8, 8, 9, 9, 10 };

int n = sizeof(arr) / sizeof(arr[0]); findFrequency(arr, n);

return 0;

}

# Observations/Outcome:



**Time Complexity is O(n)**