Assignment-52: A Job Ready Bootcamp in c++,DSA and IOT

unordered_multimap

1. Write a C++ program to initialise the unordered_map and print it on the screen.
2. Write a C++ program to copy the data of one unordered_map into another and print it on the screen.
3. Write a C++ program to illustrate the reverse function in unordered_multimap.
4. Given a string of words, the task is to find the frequencies of the individual words using unordered_map.
   Example:
   Input: str = “Ineuron for Ineuron Ineuron quiz practice qa for”;
   Output: Frequencies of individual words are
   (qa, 1)
   (quiz, 1)
   (practice, 1)
   (Ineuron, 3)
   (for, 2)
5. Given an array arr[] of size N and three integers X, Y and K, the task is to count the number of pairs (i, j) where i < j such that (arr[i] * X + arr[j] * Y) = K complete the task using unordered_map.
   Example:
   Input: arr[] = {3, 1, 2, 3}, X = 4, Y = 2, K = 14
   Output: 2
   Explanation: The possible pairs are: (1, 2), (3, 4).
   For i = 1, j = 2, Value of the expression = 4 * 3 + 2 * 1 = 14.
   For i = 3, j = 4, Value of the expression = 4 * 2 + 2 * 3 = 14.
   Input: arr[] = [1, 3, 2], X = 1, Y = 3, K = 7
   Output: 1
   Explanation: The possible pairs are: (1, 2).
   For i = 1, j = 2, Value of the expression = 1 * 1 + 2 * 3 = 7.
6. Given an array arr[] consisting of N integers, the task is to find the maximum element with the minimum frequency using unordered_map.
7. Given an array arr[] consisting of N positive integers, the task is to find the count of unique pairs (i, j) such that the sum of arr[i] and the reverse(arr[j]) is the same as the sum of reverse(arr[i]) and arr[j] using unordered_map.
   Example:
   Input: arr[] = {2, 15, 11, 7}
   Output: 3
   Explanation:
   The pairs are (0, 2), (0, 3) and (2, 3).

   - (0, 2): arr[0] + reverse(arr[2]) (= 2 + 11 = 13) and reverse(arr[0]) + arr[2](= 2 + 11 = 13).
   - (0, 3): arr[0] + reverse(arr[3]) (= 2 + 7 = 9) and reverse(arr[0]) + arr[3](= 2 + 7 = 9).

- (2, 3): arr[2] + reverse(arr[3]) (= 11 + 7 = 18) and reverse(arr[2]) + arr[3](= 11 + 7 = 18).

Input: A[] = {22, 115, 7, 313, 17, 23, 22}
Output: 6

8. Given an array A[] consisting of positive integers, the task is to find the only array element with a single occurrence using unordered_map.
9. Given an array arr[], the task is to find the count of array elements whose squares are already present in the array using unordered_map.
10. Given string str, the task is to find the minimum count of characters that need to be deleted from the string such that the frequency of each character of the string is unique using unordered_map.
    Example:
    Input: str = “ceabaacb”
    Output: 2
    Explanation:
    The frequencies of each distinct character are as follows:
    c —> 2
    e —> 1
    a —> 3
    b —> 2
    Possible ways to make frequency of each character unique by minimum number of moves are:
    - Removing both occurrences of ‘c’ modifies str to “eabaab”
    - Removing an occurrence of ‘c’ and ‘e’ modifies str to “abaacb”

    Therefore, the minimum removals required is 2.