**Assignment-51: A Job Ready Bootcamp in c++,DSA and IOT**

**unordered\_multiset**

1. Write a C++ program to initialise the unordered\_multiset and print it on the screen.

Sol – 1.

#include<bits/stdc++.h>

using namespace std;

void printUset(unordered\_multiset<int> ums)

{

unordered\_multiset<int>::iterator it;

it=ums.begin();

for(;it!=ums.end();it++)

cout<<\*it<<" ";

cout<<endl;

}

int main()

{

unordered\_multiset<int>ums1;

unordered\_multiset<int>ums2({1,3,1,4,5,9,8,7});

ums1={5,6,9,4,1,3,5};

if(ums1.empty())

cout<<"Unordered multiset 1 is empty\n";

else

cout<<"Unordered multiest 1 is not empty\n";

cout<<"The size of unordered multiset 2 is : "<<ums2.size()<<endl;

printUset(ums1);

ums1.insert(7);

printUset(ums2);

return 0;

}

2. Write a C++ program to delete all copies from an unordered\_multiset.

Example:

Input - 6 4 2 7 3 3 1 1 1

Output - 6 4 2 7 3 1

Sol – 2.

#include<bits/stdc++.h>

using namespace std;

void printUset(unordered\_multiset<int> ums)

{

unordered\_multiset<int>::iterator it;

it=ums.begin();

for(;it!=ums.end();it++)

cout<<\*it<<" ";

cout<<endl;

}

void uniqelement(unordered\_multiset<int>&ums)

{

set<int>s;

for(auto it=ums.begin();it!=ums.end();it++)

{

s.insert(\*it);

}

ums.clear();

for(auto it=s.begin();it!=s.end();it++)

{

ums.insert(\*it);

}

}

int main()

{

unordered\_multiset<int>ums1;

ums1={5,6,9,9,4,1,3,5};

uniqelement(ums1);

printUset(ums1);

return 0;

}

3. Given an array arr[] of N integer elements, the task is to change the minimum

number of elements of this array such that it contains first N terms of the Catalan

Sequence. Thus, find the minimum changes required using unordered\_multiset.

First few Catalan numbers are 1, 1, 2, 5, 14, 42, 132, 429, 1430, 4862, .....

Examples:

Input: arr[] = {4, 1, 2, 33, 213, 5}

Output: 3

We have to replace 4, 33, 213 with 1, 14, 42 to make the first 6 terms of Catalan

sequence.

Input: arr[] = {1, 1, 2, 5, 41}

Output: 1

Simply change 41 with 14

Sol – 3.

#include<bits/stdc++.h>

using namespace std;

#define MAX 100000

#define ll long long int

ll catlan[MAX];

void catlanDP(ll n)

{

catlan[0]=catlan[1]=1;

for(int i=2;i<n;i++)

{

catlan[i]=0;

for(int j=0;j<i;j++)

catlan[i]=catlan[i]+catlan[j]\*catlan[i-j-1];

}

}

int CatlanSequence(int arr[],int n)

{

catlanDP(n);

unordered\_multiset<int>s;

int a=1,b=1;

int c;

s.insert(a);

if(n>=2)

s.insert(b);

for(int i=2;i<n;i++)

{

s.insert(catlan[i]);

}

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

{

auto it=s.find(arr[i]);

if(it!=s.end())

s.erase(it);

}

return s.size();

}

int main()

{

int arr[]={1,1,2,5,41};

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

cout<<CatlanSequence(arr,n);

return 0;

}

4. Write a C++ program to illustrate the swapping of data between two

unordered\_multiset.

Sol – 4.

#include<iostream>

#include<string>

#include<unordered\_set>

using namespace std;

int main()

{

unordered\_multiset<string>first={"Gurudev"},second={"Singla"};

cout<<"Before swap : "<<endl;

cout<<"1st container : "<<endl;

for(const string& x:first)

{

cout<<x<<endl;

}

cout<<"2nd container : "<<endl;

for(const string& x:second)

{

cout<<x<<endl;

}

first.swap(second);

cout<<"After swap : "<<endl;

for(const string& x:first)

{

cout<<x<<endl;

}

cout<<"2nd container : "<<endl;

for(const string& x:second)

{

cout<<x<<endl;

}

return 0;

}

5. Write a C++ program to count the frequency of elements in unordered\_multiset.

Sol – 5.

#include<iostream>

#include<unordered\_set>

using namespace std;

int main()

{

unordered\_multiset<int>s;

s.insert(23);

s.insert(1);

s.insert(54);

s.insert(21);

s.insert(89);

s.insert(8);

cout<<"Element\tFrequency"<<endl;

for(auto it=s.begin();it!=s.end();it++)

{

cout<<\*it<<"\t"<<s.count(\*it)<<endl;

}

return 0;

}

6. Write a C++ program to illustrate the emplace() function in unordered\_multiset.

Sol – 6.

#include<iostream>

#include<unordered\_set>

using namespace std;

int main()

{

unordered\_multiset<int>s;

s.insert(23);

s.insert(1);

s.insert(54);

s.insert(21);

s.insert(89);

s.insert(8);

s.emplace(45);

for(auto it:s)

cout<<it<<" ";

return 0;

}

7. Write a C++ program to illustrate the find() function in unordered\_multiset.

Sol – 7.

#include<iostream>

#include<unordered\_set>

using namespace std;

int main()

{

unordered\_multiset<int>s;

s.insert(23);

s.insert(1);

s.insert(54);

s.insert(21);

s.insert(89);

s.insert(8);

s.emplace(45);

if(s.find(89)!=s.end())

{

cout<<"Found!"<<endl;

}

return 0;

}

8. Write a C++ program to illustrate the bucket\_count() function in unordered\_multiset.

Sol – 8.

#include<iostream>

#include<unordered\_set>

using namespace std;

int main()

{

unordered\_multiset<int>s;

s.insert(23);

s.insert(1);

s.insert(54);

s.insert(21);

s.insert(89);

s.insert(89);

s.emplace(45);

cout<<"Bucket Count : "<<s.bucket\_count()<<endl;

for(int i=0;i<s.bucket\_count();i++)

{

cout<<"\nBucket "<<i<<" : ";

if(s.bucket\_size(i)==0)

cout<<"empty";

for(auto it=s.cbegin(i);it!=s.cend(i);it++)

cout<<\*it<<" ";

}

return 0;

}

9. Write a C++ program to illustrate the load\_factor() function in unordered\_multiset.

Sol – 9.

#include<iostream>

#include<unordered\_set>

using namespace std;

int main()

{

unordered\_multiset<int>s;

s.insert(23);

s.insert(1);

cout<<"Size : "<<s.size()<<endl<<"Bucket count : "<<s.bucket\_count()<<endl;

cout<<"Load Factor : "<<s.load\_factor()<<endl;

s.insert(54);

s.insert(21);

s.insert(89);

cout<<"Size : "<<s.size()<<endl<<"Bucket count : "<<s.bucket\_count()<<endl;

cout<<"Load Factor : "<<s.load\_factor()<<endl;

s.insert(89);

s.emplace(45);

cout<<"Size : "<<s.size()<<endl<<"Bucket count : "<<s.bucket\_count()<<endl;

cout<<"Load Factor : "<<s.load\_factor()<<endl;

return 0;

}

10. Write a C++ program to illustrate the reverse() function in unordered\_multiset.

Sol – 10.

#include<iostream>

#include<unordered\_set>

using namespace std;

int main()

{

unordered\_multiset<int>s;

s.reserve(5);

s.insert(5);

s.insert(6);

s.insert(7);

for(auto &it:s)

cout<<it<<" ";

return 0;

}