**Assignment - 45 A Job Ready Bootcamp in C++, DSA and IOT**

**multiset**

1. Create a c++ program using multiset and returns an iterator to the first element in

the multiset –> O(1)

Sol – 1.

#include<bits/stdc++.h>

using namespace std;

int main()

{

int arr[]={14,10,15,11,10};

multiset<int>ms(arr,arr+5);

cout<<"First element is : "<<\*ms.begin()<<endl;

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

{

cout<<\*it<<" ";

}

cout<<endl;

return 0;

}

2. Create a c++ program using multiset and returns an iterator to the theoretical

element that follows the last element in the multiset –> O(1)

Sol – 2.

#include<bits/stdc++.h>

using namespace std;

int main()

{

int arr[]={14,10,15,11,10};

multiset<int>ms(arr,arr+5);

cout<<"First element is : "<<\*ms.begin()<<endl;

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

{

cout<<\*it<<" ";

}

cout<<endl;

cout<<"Last element is : "<<\*--ms.end()<<endl;

return 0;

}

3. Create a c++ program using multiset and returns the number of elements in the

multiset –> O(1)

Sol – 3.

#include<bits/stdc++.h>

using namespace std;

int main()

{

int arr[]={14,10,15,11,10};

multiset<int>ms(arr,arr+5);

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

{

cout<<\*it<<" ";

}

cout<<endl;

cout<<"Size : "<<ms.size()<<endl;

return 0;

}

4. Create a c++ program using multiset and returns the maximum number of elements

that the multiset can hold –> O(1)

Sol – 4.

#include<bits/stdc++.h>

using namespace std;

int main()

{

int arr[]={14,10,15,11,10};

multiset<int>ms(arr,arr+5);

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

{

cout<<\*it<<" ";

}

cout<<endl;

cout<<"Max Size : "<<ms.max\_size()<<endl;

return 0;

}

5. Create a c++ program using multiset and returns whether the multiset is empty –>

O(1)

Sol – 5.

#include<bits/stdc++.h>

using namespace std;

int main()

{

int arr[]={14,10,15,11,10};

multiset<int>ms(arr,arr+5);

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

{

cout<<\*it<<" ";

}

cout<<endl;

if(ms.empty())

cout<<"Multiset is empty"<<endl;

else

cout<<"Multiset is not empty"<<endl;

return 0;

}

6. Create a c++ program using multiset and inserts the element x in the multiset –>

O(log n)

Sol – 6.

#include<bits/stdc++.h>

using namespace std;

int main()

{

multiset<int,greater<int>>ms;

ms.insert(10);

ms.insert(13);

ms.insert(50);

ms.insert(45);

ms.insert(33);

ms.insert(10);

cout<<"Elements in set in decreasing order : ";

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

{

cout<<\*it<<" ";

}

cout<<endl;

return 0;

}

7. Create a c++ program using multiset and removes all the elements from the multiset

–> O(n)

Sol – 7.

#include<bits/stdc++.h>

using namespace std;

int main()

{

multiset<int,greater<int>>ms;

ms.insert(10);

ms.insert(13);

ms.insert(50);

ms.insert(45);

ms.insert(33);

ms.insert(10);

cout<<"Elements in set in decreasing order : ";

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

{

cout<<\*it<<" ";

}

cout<<endl;

ms.erase(++ms.begin(),--ms.end());

cout<<"Size : "<<ms.size()<<endl;

ms.clear();

cout<<"Size : "<<ms.size()<<endl;

return 0;

}

8. Create a c++ program using multiset and removes all the occurrences of x –> O(log

n)

Sol – 8.

#include<bits/stdc++.h>

using namespace std;

int main()

{

multiset<int,greater<int>>ms;

ms.insert(10);

ms.insert(13);

ms.insert(50);

ms.insert(45);

ms.insert(33);

ms.insert(10);

cout<<"Elements in set in decreasing order : ";

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

{

cout<<\*it<<" ";

}

cout<<endl;

ms.erase(10);

cout<<"After removing 10 : ";

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

{

cout<<\*it<<" ";

}

cout<<endl;

return 0;

}

9. Create a c++ program using multiset and remove only one instance of element from

multiset having same value

Sol – 9.

#include<bits/stdc++.h>

using namespace std;

int main()

{

multiset<int,greater<int>>ms;

ms.insert(10);

ms.insert(13);

ms.insert(50);

ms.insert(45);

ms.insert(33);

ms.insert(10);

cout<<"Elements in set in decreasing order : ";

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

{

cout<<\*it<<" ";

}

cout<<endl;

multiset<int,greater<int>>::iterator it;

it=ms.find(10);

if(it!=ms.end())

{

ms.erase(it);

}

cout<<"After removing 10 : ";

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

{

cout<<\*it<<" ";

}

cout<<endl;

return 0;

}

10. Unlike a set, a multiset may contain multiple occurrences of same number. The

multiset equivalence problem states to check if two given multisets are equal or not.

For example let A = {1, 2, 3} and B = {1, 1, 2, 3}. Here A is set but B is not (1 occurs

twice in B), whereas A and B are both multisets. More formally, “Are the sets of pairs

defined as A' ={ (a, frequency(a)) | a belongs to A} equal for the two given

multisets?” Given two multisets A and B, write a program to check if the two multisets

are equal.

Sol – 10.

#include<bits/stdc++.h>

using namespace std;

bool areSame(multiset<int>&a,multiset<int>&b)

{

return a==b;

}

int main()

{

multiset<int>a({7,7,5}),b({7,5,7});

if(areSame(a,b))

{

cout<<"Yes"<<endl;

}

else

cout<<"No"<<endl;

return 0;

}