**SET STL**

//the main drawback of vector is its sorting takes O(nlogn) times and if a number is added in vector then we need to sort it again for performing some operation again

//this can be overcome by using set datastructure- most of the operation can be done in O(n) times.

#include <iostream>

#include<set>

#include<algorithm>

using namespace std;

int main()

{

set<int> S;

S.insert(1);

S.insert(2);

S.insert(-1);

S.insert(-10);

// Set is a data structure which automatically arrange elemnts in ascending order so sorting takes O(n) times.

set<int>::iterator itr;

for(itr=S.begin();itr!=S.end();itr++)

{

cout<<\*itr<<" ";

}

/\*the output can also be shown by doing following step

for(int x : S)

{

cout<<x<<" "; //Similar to that as above

}\*/

cout<<endl;

itr=S.find(-1);

if(itr==S.end()) // If the element is not present in the set then itr is assigned to end

cout<<"Element is not present"<<endl;

else

cout<<"It's present"<<endl;

auto it1=S.upper\_bound(-1); // in the vector we need to pass begin and end as parameter but in set do not pass and can directly give the value to which >= or > value we wants

//auto is a keyword which automatically identifies that it1 is an iterator, C++ is smart enough..

auto it2=S.upper\_bound(0);

auto it3=S.upper\_bound(2); //since no value >2 is present so it3 points to end

cout<<\*it1<<" "<<\*it2<<" "<<\*it3<<endl;

}