Q1. Find the total number of elements of the set container.

```
#include <iostream>
#include <set>
using namespace std;

int main() {
    set <int> s = {10, 20, 30, 40, 50};

    cout<<s.size();
}
```

Q2. Using inbuilt function and insert an element to the set container.

```
#include <iostream>
#include <set>
using namespace std;

int main()
{
   set <int> q = {10, 20, 30, 40};

   q.insert(5);

   for(int x: q) cout<<x<<" ";

}
```

Q3. Erase an element from a set.

```
#include <iostream>
#include <set>

using namespace std;

int main()
{
   set <int> q = {10, 20, 30, 70};

   q.erase(30);

   for(int x : q) cout<<x<<" ";
}
```

Q4. Checks whether the set is empty or not.if it is empty insert 5 elements in the set

```
#include <iostream>
#include <set>
using namespace std;

int main()
{
   set <int> s;

   if(s.empty())
   {
      cout<<"set is empty";
      for(int i = 1; i < 6; i++)
      {
         s.insert(i * 10);
      }
   }
   else
   {
         cout<<"set is not empty";
   }
}
```

Q5. Make a c++ program to insert unique element in set

```
#include <iostream>
#include <set>
using namespace std;

int main()
{
   set <int> s;

   int n = 0, x = 0;

   cout<<"How many elements you want to enter = ";
   cin>>n;

   for(int i = 0; i < n; i++)
   {
      cout<<"Enter element = ";
      cin>>x;

      set <int>::iterator it = s.find(x);
      if(it == s.end())
      {
         s.insert(x);
         cout<<"element inserted"<<endl;
      }
      else
      {
         i = i - 1;
         cout<<"element already inserted"<<endl;
      }
   }
}
```

Q6. How to find the lower bound of any desired element from the set.

```
#include <iostream>
#include <set>
using namespace std;

int main()
{
   set <int> s = {10, 20, 30, 40, 50};

   set <int>::iterator it;

   int x;
   cout<<"enter number = ";
   cin>>x;

      it = s.find(x);
      if(it == s.end())
      {
         for (int i : s)
         {
            if(x < i)
            {
               cout<<"lower bound = "<<i;
               break;
            }
         }
      }
      else if((*it) == x)
      {
         cout<<"lower bound = "<<x;
      }
      else
      {
         cout<<"there is no loer bound";
      }

}
```

Q7. how to find the upper bound of any desired element from the set.

```
#include <iostream>
#include <set>
using namespace std;

int main()
{
   set <int> s = {10, 20, 30, 40, 50};

   set <int>::iterator it;

   int x;
   cout<<"enter number = ";
   cin>>x;

      it = s.find(x);
      if(it == s.end())
      {
         for (int i : s)
         {
            if(x < i)
            {
               cout<<"upper bound = "<<i;
               break;
            }
         }
      }
      else if((*it) == x)
      {
         it++;
         cout<<"upper bound = "<<*it;
      }
      else
      {
         cout<<"there is no upper bound";
      }

}
```

Q8. Create a function to search the element in the set.

```
#include <iostream>
#include <set>
using namespace std;

int main()
{
   set <int> s = {10, 20, 30, 40, 60, 50};
   set <int>::iterator it = s.begin();

   int x;
   int flag = 0;

   cout<<"Enter element = ";
   cin>>x;

   for(; it != s.end(); it++)
   {
      if((*it) == x)
      {
         cout<<"element found";
         flag = 1;
      }
   }

   if(!flag)
   {
      cout<<"element not found";
   }

}
```

Q9. Converting String into Set in C++ STL

```
#include <iostream>
#include <string>
#include <set>
using namespace std;

int main() {
    string s = "hello world";
    set<char> charSet(s.begin(), s.end());


    for (char c : charSet) {
        cout << c << " ";
    }
    cout <<endl;

    return 0;
}
```

Q10. You have a set of integer s, which originally contains all the numbers from 1 to n. Unfortunately, due to some error, one of the numbers in s got duplicated to another number in the set, which results in repetition of one number and loss of another number.You are given an integer array nums representing the data status of this set after the error.Find the number that occurs twice and the number that is missing and return them in the form of an array.You have a set of integer s, which originally contains all the numbers from 1 to n.
Unfortunately, due to some error, one of the numbers in s got duplicated to another number in the set, which results in repetition of one number and loss of another number.You are given an integer array nums representing the data status of this set after the error.Find the number that occurs twice and the number that is missing and return them in the form of an array.

```
#include <iostream>
#include <set>
#include <array>
using namespace std;

int main() {

    int n = 0;
    int err = 0;

    cout<<"Enter value of n = ";
    cin>>n;


    cout<<"Enter duplicate number = ";
    cin>>err;

    int nums[n];

    for(int i = 0; i < n; i++)//enter data and duplicate value in to array
    {
        if((i + 1) == err)
        {
            nums[i] = i + 1;
            i = i + 1;
            nums[i] = i;
        }
        else
        {
             nums[i] = i + 1;
        }
    }

   set <int> s;
   for(int i = 0; i < n; i++)//enter data in set
    {
        s.insert(nums[i]);
    }
    set <int>::iterator it = s.begin();

    array <int, 2> ans;
    for(int i = 0; i < n; it++, i++)//finding ans
    {
```

```
        if(nums[i] == (*it) && it != s.end())
        {
            continue;
        }
        else
        {
            cout<<"else";
            ans[0] = nums[i];
            ans[1] = nums[i] + 1;
            break;
        }
    }

    cout<<"[ "<<ans[0]<<", "<<ans[1]<<" ]";
}
```