1.Vectors

Removes the element present at position.

Ex: v.erase(v.begin()+4); (erases the fifth element of the vector v)

Removes the elements in the range from start to end inclusive of the start and exclusive of the end.

Ex:v.erase(v.begin()+2,v.begin()+5);(erases all the elements from the third element to the fifth element.)

2.Lower Bound Stl

Indexaddress=lower\_bound(start,end,query element);

Returns the address of query element if found(binary search) or if not found then it returns the address of the next higher element ..if query element is not present and also there is no higher number present then it returns the address of (end-1)th address..

3.Sets -STL

set<int>s; //Creates a set of integers.

int length=s.size(); //Gives the size of the set.

s.insert(x); //Inserts an integer x into the set s.

s.erase(val); //Erases an integer val from the set s.

finding an element in a set:

set<int>::iterator itr=s.find(val); //Gives the iterator to the element val if it is found otherwise returns s.end() .

Ex: set<int>::iterator itr=s.find(100); //If 100 is not present then it==s.end().

4.Maps

map<string,int>m; //Creates a map m where key\_type is of type string and data\_type is of type int.

int length=m.size(); //Gives the size of the map.

m.insert(make\_pair("hello",9)); //Here the pair is inserted into the map where the key is "hello" and the value associated with it is 9.

m.erase(val); //Erases the pair from the map where the key\_type is val.

Finding an element:

map<string,int>::iterator itr=m.find(val); //Gives the iterator to the element val if it is found otherwise returns m.end() .

Ex: map<string,int>::iterator itr=m.find("Maps"); //If Maps is not present as the key value then itr==m.end().

Accessing the value stored in a key

To get the value stored of the key "MAPS" we can do m["MAPS"] or we can get the iterator using the find function and then by itr->second we can access the value.

5.Manipulators(used for structuting the output)(very helpful)

<http://www.cplusplus.com/reference/library/manipulators/>

**Independent flags (switch on)**:

[**boolalpha**](http://www.cplusplus.com/reference/ios/boolalpha/)

Alphanumerical bool values (function )

[**showbase**](http://www.cplusplus.com/reference/ios/showbase/)

Show numerical base prefixes (function )

[**showpoint**](http://www.cplusplus.com/reference/ios/showpoint/)

Show decimal point (function )

[**showpos**](http://www.cplusplus.com/reference/ios/showpos/)

Show positive signs (function )

[**skipws**](http://www.cplusplus.com/reference/ios/skipws/)

Skip whitespaces (function )

[**unitbuf**](http://www.cplusplus.com/reference/ios/unitbuf/)

Flush buffer after insertions (function )

[**uppercase**](http://www.cplusplus.com/reference/ios/uppercase/)

Generate upper-case letters (function )

**Independent flags (switch off)**:

[**noboolalpha**](http://www.cplusplus.com/reference/ios/noboolalpha/)

No alphanumerical bool values (function )

[**noshowbase**](http://www.cplusplus.com/reference/ios/noshowbase/)

Do not show numerical base prefixes (function )

[**noshowpoint**](http://www.cplusplus.com/reference/ios/noshowpoint/)

Do not show decimal point (function )

[**noshowpos**](http://www.cplusplus.com/reference/ios/noshowpos/)

Do not show positive signs (function )

[**noskipws**](http://www.cplusplus.com/reference/ios/noskipws/)

Do not skip whitespaces (function )

[**nounitbuf**](http://www.cplusplus.com/reference/ios/nounitbuf/)

Do not force flushes after insertions (function )

[**nouppercase**](http://www.cplusplus.com/reference/ios/nouppercase/)

Do not generate upper case letters (function )

**Numerical base format flags ("basefield" flags)**:

[**dec**](http://www.cplusplus.com/reference/ios/dec/)

Use decimal base (function )

[**hex**](http://www.cplusplus.com/reference/ios/hex/)

Use hexadecimal base (function )

[**oct**](http://www.cplusplus.com/reference/ios/oct/)

Use octal base (function )

**Floating-point format flags ("floatfield" flags)**:

[**fixed**](http://www.cplusplus.com/reference/ios/fixed/)

Use fixed floating-point notation (function )

[**scientific**](http://www.cplusplus.com/reference/ios/scientific/)

Use scientific floating-point notation (function )

**Adustment format flags ("adjustfield" flags)**:

[**internal**](http://www.cplusplus.com/reference/ios/internal/)

Adjust field by inserting characters at an internal position (function )

[**left**](http://www.cplusplus.com/reference/ios/left/)

Adjust output to the left (function )

[**right**](http://www.cplusplus.com/reference/ios/right/)

Adjust output to the right (function )

**Input manipulators**

[**ws**](http://www.cplusplus.com/reference/istream/ws/)

Extract whitespaces (function )

**Output manipulators**

[**endl**](http://www.cplusplus.com/reference/ostream/endl/)

Insert newline and flush (function )

[**ends**](http://www.cplusplus.com/reference/ostream/ends/)

Insert null character (function )

[**flush**](http://www.cplusplus.com/reference/ostream/flush-free/)

Flush stream buffer (function )