```
/*
 * Modify Elements in a Container
 */

vector<int> vec = {1,2,3,4,5};
vec[2] = 9;   // vec: {1,2,9,4,5}


list<int> mylist = {1,2,3,4,5};
list<int>::iterator itr = mylist.find(3);
if (itr != mylist.end())
      *itr = 9;   // mylist: {1,2,9,4,5}


// How about modifying a set?
set<int> myset = {1,2,3,4,5};
set<int>::iterator itr = myset.find(3);
if (itr != myset.end()) {
      *itr = 9;     // Many STL implementation won't compile
      const_cast<int&>(*itr) = 9;  // {1,2,9,4,5} ???


// What about map
map<char,int> m;
m.insert ( make_pair('a',100) );
m.insert ( make_pair('b',200) );
m.insert ( make_pair('c',300) );
...
map<char,int>::iterator itr = m.find('b');
if (itr != m.end()) {
      itr->second = 900;   // OK
      itr->first = 'd';    // Error
}


// Same thing for multimap, multiset, unordered set/multiset, unordered
map/multimap


/*
 * How to modify an element of associative container or unordered
container?
 */
map<char,int> m;
m.insert ( make_pair('a',100) );
m.insert ( make_pair('b',200) );
m.insert ( make_pair('c',300) );
...
map<char,int>::iterator itr = m.find('b');
if (itr != m.end()) {
      pair<char,int> orig(*itr);
      orig.first = 'd';
      m.insert(orig);
}
```