

```

/* The following code example is taken from the book
 * "The C++ Standard Library – A Tutorial and Reference"
 * by Nicolai M. Josuttis, Addison-Wesley, 1999
 *
 * (C) Copyright Nicolai M. Josuttis 1999.
 * Permission to copy, use, modify, sell and distribute this software
 * is granted provided this copyright notice appears in all copies.
 * This software is provided "as is" without express or implied
 * warranty, and with no claim as to its suitability for any purpose.
 */
#include <iostream>
#include <set>
#include <algorithm>
#include <iterator>
using namespace std;

int main()
{
    /* type of the collection:
     * - duplicates allowed
     * - elements are integral values
     * - descending order
     */
    typedef multiset<int, greater<int> > IntSet;

    IntSet coll1;          // empty multiset container

    // insert elements in random order
    coll1.insert(4);
    coll1.insert(3);
    coll1.insert(5);
    coll1.insert(1);
    coll1.insert(6);
    coll1.insert(2);
    coll1.insert(5);

    // iterate over all elements and print them
    IntSet::iterator pos;
    for (pos = coll1.begin(); pos != coll1.end(); ++pos) {
        cout << *pos << ' ';
    }
    cout << endl;

    // insert 4 again and process return value
    IntSet::iterator ipos = coll1.insert(4);
    cout << "4 inserted as element "
         << distance(coll1.begin(), ipos) + 1 << endl;

    // assign elements to another multiset with ascending order
    multiset<int> coll2(coll1.begin(),
                       coll1.end());

    // print all elements of the copy
    copy (coll2.begin(), coll2.end(),
          ostream_iterator<int>(cout, " "));
    cout << endl;
}

```

```
// remove all elements up to element with value 3
coll2.erase (coll2.begin(), coll2.find(3));

// remove all elements with value 5
int num;
num = coll2.erase (5);
cout << num << " element(s) removed" << endl;

// print all elements
copy (coll2.begin(), coll2.end(),
      ostream_iterator<int>(cout, " "));
cout << endl;
}
```