```
/* 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 <algorithm>
#include <map>
using namespace std;

/* function object to check the value of a map element
 */
template <class K, class V>
class value_equals {
  private:
    V value;
  public:
    // constructor (initialize value to compare with)
    value_equals (const V& v)
     : value(v) {
    }
    // comparison
    bool operator() (pair<const K, V> elem) {
        return elem.second == value;
    }
};

int main()
{
    typedef map<float,float> FloatFloatMap;
    FloatFloatMap coll;
    FloatFloatMap::iterator pos;

    // fill container
    coll[1]=7;
    coll[2]=4;
    coll[3]=2;
    coll[4]=3;
    coll[5]=6;
    coll[6]=1;
    coll[7]=3;

    // search an element with key 3.0
    pos = coll.find(3.0);                     // logarithmic complexity
    if (pos != coll.end()) {
        cout << pos->first << ": "
             << pos->second << endl;
    }

    // search an element with value 3.0
    pos = find_if(coll.begin(),coll.end(),    // linear complexity
                  value_equals<float,float>(3.0));
    if (pos != coll.end()) {
        cout << pos->first << ": "
             << pos->second << endl;
    }
}
```