```
/* 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 <string>
#include <deque>
#include <set>
#include <algorithm>
using namespace std;


/* class Person
 */
class Person {
  private:
    string fn;    // first name
    string ln;    // last name
  public:
    Person() {
    }
    Person(const string& f, const string& n)
     : fn(f), ln(n) {
    }
    string firstname() const;
    string lastname() const;
    // ...
};

inline string Person::firstname() const {
    return fn;
}

inline string Person::lastname() const {
    return ln;
}

ostream& operator<< (ostream& s, const Person& p)
{
    s << "[" << p.firstname() << " " << p.lastname() << "]";
    return s;
}


/* binary function predicate:
 * - returns whether a person is less than another person
 */
bool personSortCriterion (const Person& p1, const Person& p2)
{
    /* a person is less than another person
```

```
     * - if the last name is less
     * - if the last name is equal and the first name is less
     */
    return p1.lastname()<p2.lastname() ||
           (p1.lastname()==p2.lastname() &&
            p1.firstname()<p2.firstname());
}

int main()
{
    // create some persons
    Person p1("nicolai","josuttis");
    Person p2("ulli","josuttis");
    Person p3("anica","josuttis");
    Person p4("lucas","josuttis");
    Person p5("lucas","otto");
    Person p6("lucas","arm");
    Person p7("anica","holle");

    // insert person into collection coll
    deque<Person> coll;
    coll.push_back(p1);
    coll.push_back(p2);
    coll.push_back(p3);
    coll.push_back(p4);
    coll.push_back(p5);
    coll.push_back(p6);
    coll.push_back(p7);

    // print elements
    cout << "deque before sort():" << endl;
    deque<Person>::iterator pos;
    for (pos = coll.begin(); pos != coll.end(); ++pos) {
        cout << *pos << endl;
    }

    // sort elements
    sort(coll.begin(),coll.end(),    // range
         personSortCriterion);       // sort criterion

    // print elements
    cout << "deque after sort():" << endl;
    for (pos = coll.begin(); pos != coll.end(); ++pos) {
        cout << *pos << endl;
    }
}
```