```
/* The following code example is taken from the book
 * "The C++ Standard Library - A Tutorial and Reference, 2nd Edition"
 * by Nicolai M. Josuttis, Addison-Wesley, 2012
 *
 * (C) Copyright Nicolai M. Josuttis 2012.
 * 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 <forward_list>
#include <iostream>
#include <algorithm>
#include <iterator>
#include <string>
using namespace std;

void printLists (const string& s, const forward_list<int>& l1,
                                  const forward_list<int>& l2)
{
    cout << s << endl;
    cout << " list1: ";
    copy (l1.cbegin(), l1.cend(), ostream_iterator<int>(cout," "));
    cout << endl << " list2: ";
    copy (l2.cbegin(), l2.cend(), ostream_iterator<int>(cout," "));
    cout << endl;
}

int main()
{
    // create two forward lists
    forward_list<int> list1 = { 1, 2, 3, 4 };
    forward_list<int> list2 = { 77, 88, 99 };
    printLists ("initial:", list1, list2);

    // insert six new element at the beginning of list2
    list2.insert_after(list2.before_begin(),99);
    list2.push_front(10);
    list2.insert_after(list2.before_begin(), {10,11,12,13} );
    printLists ("6 new elems:", list1, list2);

    // insert all elements of list2 at the beginning of list1
    list1.insert_after(list1.before_begin(),
                       list2.begin(),list2.end());
    printLists ("list2 into list1:", list1, list2);

    // delete second element and elements after element with value 99
    list2.erase_after(list2.begin());
    list2.erase_after(find(list2.begin(),list2.end(),
                           99),
                      list2.end());
    printLists ("delete 2nd and after 99:", list1, list2);

    // sort list1, assign it to list2, and remove duplicates
    list1.sort();
    list2 = list1;
```

```
    list2.unique();
    printLists ("sorted and unique:", list1, list2);

    // merge both sorted lists into list1
    list1.merge(list2);
    printLists ("merged:", list1, list2);
}
```