```cpp
//============================================================================
// Name        : ICE.cpp
// Author      : Paul Hrycewicz
// Version     :
// Copyright   : Your copyright notice
// Description : Hello World in C++, Ansi-style
//============================================================================

#include <iostream>
#include <list>
using namespace std;

bool equalto2 (int s)
{
    if (s == 2) return true;
    return false;
}

int main() {
    list <int> LL;
    LL.push_front(2);
    LL.push_back(3);
    LL.push_front(2);
    LL.push_back(3);
    LL.push_front(2);
    LL.push_back(3);
    cout << LL.size() << endl;
    LL.remove(2);       // remove all the 2's
    cout << LL.size() << endl;

    LL.pop_front();     // removes only the front
    cout << LL.size() << endl;

    int i;
    i = LL.front();     // retrieves (does not remove) value in front node
    i = LL.back();      // back node

    LL.unique();        // removes consecutive dups,
                        // but not necessarily all

    LL.sort();          // unless we sort first

    LL.push_back(2);
    LL.push_back(3);
    LL.push_back(4);
    LL.push_back(5);

    list<int>::iterator li;     // define list iterator for
                                // a list of ints
    li = LL.begin();    // point iterator at the 0th node
    li++;               // now node 1
    li++;               // and node 2
    cout << *li<<endl;  // display the value that's in node 2
                        // by dereferencing the iterator

// iterate through the list using the iterator
// remember that the list starts at LL.begin() and ends at
// LL.end().  LL.end() points to the imaginary element
// that's just past the actual last element
    for (li=LL.begin(); li != LL.end(); li ++)
    {
        cout << "list entry " << *li << endl;
    }

    LL.remove_if (equalto2);    // uses the removal function equalto2
                                // see the function above

    return 0;
}

```