```cpp
#include "List.h"
#include <iostream>
#include <cassert>

int main()
{
    List<int> l;

    assert(l.size() == 0);
    assert(l.empty());

    l.push_front(44);          // l = 44
    assert(!l.empty());
    assert(l.front() == 44);
    assert(l.back() == 44);

    l.push_front(33);          // l = 33, 44
    assert(l.size() == 2);
    assert(l.front() == 33);
    assert(l.back() == 44);

    l.push_front(22);          // l = 22, 33, 44
    List<int>::iterator it = l.begin();
    l.insert(it, 11);          // l = 11, 22, 33, 44
    it = l.begin();
    assert(l.front() == 11);
    assert(*it == 11);
    assert(*++it == 22);
    assert(*++it == 33);
    assert(*++it == 44);

    it = l.begin();
    ++it;
    ++it;
    ++it;
    l.insert(it, 38);          // l = 11, 22, 33, 38, 44
    List<int>::iterator it2 = l.begin();
    assert(*it2 == 11);
    assert(*++it2 == 22);
    assert(*++it2 == 33);
    assert(*++it2 == 38);
    assert(*++it2 == 44);

    l.pop_front();             // l = 22, 33, 38, 44
    it2 = l.begin();
    assert(*it2 == 22);
    assert(*++it2 == 33);
    assert(*++it2 == 38);
    assert(*++it2 == 44);
```

```
    l.pop_back();              //    l = 22, 33, 38
    List<int> copy = l;        // copy = 22, 33, 38
    assert(copy.size() == 3);
    List<int>::iterator it3 = copy.begin();
    assert(*it3 == 22);
    assert(*++it3 == 33);

    copy.erase(it3);           // copy = 22, 38
    assert(copy.size() == 2);
    it3 = copy.begin();
    assert(*it3 == 22);
    assert(*++it3 == 38);


    return 0;
}
```