C++ 单向链表

作者：heyang5188

原文链接：https://ld246.com/article/1502955115962

来源网站：链滴

```
int  main()
{
    forward_list<int> list{ 934,34,3,1,4,14,2,2,52,23,24,25,23,23,52,5,2,32,42,232,2,52,23 };
    auto prev = list.before_begin(); //前驱
    auto begin = list.begin();//后继
    while (begin != list.end())
    {
        if (*begin % 2 == 0) {
            begin = list.erase_after(prev);
        }
        else {
            begin = list.insert_after(begin,*begin);
            prev = begin;
            begin++;

        }
    }
    for (auto &elem : list)
        cout << elem << endl;
    system("pause");
    return 0;
}
```

函数只使用迭代器和insert 和 erase

```
string function(string &s, const string &oldval, const string &newVal)
{
    auto l = oldval.size(); //定义旧的元素
    auto iter = s.begin();  //起点迭代器
    while (iter != s.end())
    {
        auto iter1 = iter;  //保留 起点迭代器，复制一个iter1 迭代器用来检验每个字符
        auto iter2 = oldval.begin();
        while (iter2 != oldval.end() && *iter1 == *iter2) { //当且仅当字符相等的时候，两个string迭代器加1
            iter1++; iter2++;
        }
        if (iter2 == oldval.end()){ // 小字符的迭代器到了微端，表示两个string 有相同的地方
            iter = s.erase(iter, iter1); //删除 起点跌代器到标识位
            if (newVal.size()) {
                iter2 = newVal.end();
                do {                //倒着插入
                    iter2--;
                    iter = s.insert(iter, *iter2);
                } while (iter2 != newVal.begin());
            }
            iter += newVal.size(); //完成替换重置迭代器
        }
        else iter++;
    }
    return s;
}
```

原文链接：C++ 单向链表