

```

#include<iostream>
#include<algorithm>
#include<list>

/**
push_front push_back insert:插入元素
指针:begin end 元素:front back
empty:判断是否为空
sort:默认升序排序(可以创建函数进行改变)
reverse:倒置
iterator:获取迭代器
find(使用迭代器),返回的是一个迭代器指针
assign:集合中元素替换为几个什么info元素
pop_front pop_back:删除开头结尾元素
clear:清除全部元素
erase:迭代遍历删除元素,下一个元素的指针移动到这个位置
*/

using namespace std;

int compare(int a,int b){
    return a>b;
}

int main(){
    list<int> list1;
    //插入元素(开头或者结尾)
    list1.push_front(12);
    list1.push_front(45);
    list1.push_back(30);
    //进行遍历
    cout<<"遍历:";
    list<int>::iterator it=list1.begin();
    while(it!=list1.end()){
        cout<<*it<<" ";
        it++;
    }
    cout<<endl;
    //在指定位置插入元素(运用指针)
    list1.insert(++list1.begin(),1,54);
    //list1.insert(list1.begin(),2,65);
    it=list1.begin();
    cout<<"插入后遍历";
    while(it!=list1.end()){
        cout<<*it<<" ";
    }
}

```

```
        it++;
    }
    cout<<endl;
    cout<<"开始元素:"<<list1.front()<<endl;
    cout<<"最后元素:"<<list1.back()<<endl;
    cout<<"集合是否为空:"<<list1.empty()<<endl;
```

```
//实现排序
list1.sort(compare);
//实现集合的倒转
list1.reverse();
//查找元素
list<int>::iterator itfind=find(list1.begin(),list1.end(),12);
cout<<"实现排序倒转遍历并查找:";
it=list1.begin();
while(it!=list1.end()){
    if(it==itfind){
        cout<<"查找成功:";
    }
    cout<<*it<<" ";

    it++;
}
cout<<endl;
```

```
//将集合中的元素全部换为5个10
list1.assign(5,10);
//删除开始元素
list1.pop_front();
//删除末尾元素
list1.pop_back();
it=list1.begin();
cout<<"替换删除后的遍历:";
while(it!=list1.end()){
    cout<<*it<<" ";
    //一个个删除
    it=list1.erase(it);
}
cout<<endl;
cout<<"遍历删除后的集合是否为空:"<<list1.empty()<<endl;
```

```
//删除全部元素
list1.clear();
```

}