//list反转和排序

#include<iostream>

#include<list>

#include<string>

#include<algorithm>

using namespace std;

void print(list<int>& l) {

for (list<int>::iterator it = l.begin(); it != l.end(); it++) {

cout << \*it << " ";

}cout << endl;

}

void test01() {

//反转

list<int>l1;

l1.push\_back(20);

l1.push\_back(50);

l1.push\_back(40);

l1.push\_back(30);

l1.push\_back(10);

cout << "反转前" << endl;

print(l1);

//反转

cout << "反转后" << endl;

l1.reverse();

print(l1);

}

bool mc(int v1,int v2) {

//降序，就让第一个数大于第二个数

return v1 > v2;

}

//排序

void test02() {

list<int>l1;

l1.push\_back(30);

l1.push\_back(40);

l1.push\_back(20);

l1.push\_back(50);

l1.push\_back(10);

//排序

cout << "排序前" << endl;

print(l1);

//所以不支持随机访问的迭代器的容器不可以用标准算法

//不支持随机访问的迭代器的容器，内部会提供算法

//sort(l1.begin(), l1.end());

l1.sort();

cout << "排序后" << endl;

print(l1);

l1.sort(mc);

print(l1);

}

int main(){

//test01();

test02();

system("pause");

return 0;

}