1 内置容器

1.1 vector

接口	示例
begin()	vector<int>::iterator it = myvector.begin();
end()	vector<int>::iterator it = myvector.end();
size()	myvector.size();
empty()	myvector.empty();
at(i)	myvector.at(i);
front()	myvector.front();
back()	myvector.back();
push_back()	myvector.push_back(num);
pop_back()	myvector.pop_back();
insert()	myvector.insert(myvector.begin(),value);
erase()	myvector.erase(myvector.begin()+3);
clear()	myvector.clear();

1.2 list

begin()	list<int>::iterator it = mylist.begin();
end()	list<int>::iterator it = mylist.end();
empty()	mylist.emprty();
size()	mylist.size();
front()	mylist.front();
back()	mylist.back();
push_front()	mylist.push_front();
pop_front()	mylist.pop_front();
push_back()	mylist.push_back();
pop_back()	mylist.pop_back();
insert()	mylist.insert(mylist.begin()+3,value);
erase()	mylist.erase(mylist.begin()+2);
clear()	mylist.clear();
remove()	mylist.remove(value);
sort()	mylist.sort();
unique()	mylist.unique();
merge()	mylist1.merge(mylist2);

1.3 stack

empty()	mystack.empty();
size()	mystack().size();
top()	mystack.top();
push()	mystack.push(value);
pop()	mystack.pop();

1.4 queue

empty()	myqueue.empty();
size()	myqueue.size();
front()	myqueue.front();
back()	myqueue.back();
push()	myqueue.push();
pop()	myqueue.pop();

1.5 set

begin()	myset.begin();
end()	myset.end();
empty()	myset.empty();
size()	myset.size();
insert()	myset.insert(value);
erase()	myset.erase(value);
clear()	myset.clear();
find()	set<int>::iterator it = myset.find(value);

1.6 map

construct	map<char,int> mymap;mymap['a']=1;
begin()	mymap.begin();
end()	mymap.end();
empty()	mymap.empty();
size()	mymap.size();
at()	mymap.at('a');
insert()	mymap.insert(std::pair<char,int>('a',3));
erase()	mymap.erase(key);
clear()	mymap.clear();
find()	mymap.find(key);