P2

STL六大部件：容器(Container)、算法(Algorithm)、迭代器(Iterators)、仿函数(Functor)、分配器(Allocator)和适配器(Adapter)。

容器有默认的分配器，可以不写，也可以写，如：

vector<int, allocator<int>> vi;//模板里的int应该一样

前闭后开区间，c.begin()指向容器第一个数据，c.end()指向容器最后一个数据后面空间，如：

*Container<T>* c;

*Container<T>*::iterator ite = c.begin(); //ite是一个指针

for(; ite != c.end(); ++ite)

{…}

//C++11的range-based for, 逐个取容器里的元素

//eg1

for(int i:{2,3,5,7,9}){

std::cout<<i<<std::endl;}

//eg2

std::vector<double> vec;

…

for(auto elem:vec){

std::cout<<elem<<std::endl;}

for(auto& elem:vec){

elem \*=3;}

P3测试array

容器分为两类：Sequence Container和Associative Container。Sequence Container主要包括Array，Vector，Deque，List，Forward-List（10分钟处的示意图）；Associative Container包括Set/Multiset, Map/Multimap, Hash Table Separate Chaining等。

P4测试vector

测试技巧：每个测试定义一个namespace，在namespace前include所以需要的模块。

vector安装2倍增长分配空间。如，现在一个vector有两个元素，当pushback一个新元素时，会将size增大到4；当放入第5个元素时，size增大到8，以此类推。每次扩容都要找到一段足够大的内存，再将原来的数据搬过去。vector是连续分配的。其中vector.size()和vector.capacity()分别表示vector的大小和容量。

异常抓取：

try{

...

}

catch(exception& p){

cout<<p.what()<<endl;//打印错误信息

abort(); //重要

}

P5

deque是分段连续，让使用者以为是连续的。参考20:00。

P6