1. （迭代器）是STL的核心技术，是通用算法的技术基础。
2. 三种STL容器适配器是（stack（栈））、（queue（队列））和（priority\_queue（优先级队列））。
3. 什么是函数对象？它通常用在什么地方？

答：函数对象是一个类，它重载了函数调用操作符（operator()）。该操作符封装了应该被实现为一个函数的操作。典型情况下，函数对象被作为实参传递给泛型算法，以完全解决类型依赖性的问题。和“引用”一样，“函数对象”很少独立使用。

函数对象与函数指针相比较有三个优点：第一，函数指针是间接引用，不能作为内联函数，而函数对象可以，这样速度更快；第二，类有数据域，函数对象可以拥有任意数量的额外数据，用这些数据可以用来缓冲当前数据和结果，提高运行质量；第三，编译时对函数对象作类型检查。

### 算法、迭代器与容器的关系？

### 答：每个容器都有相应的迭代器，容器通过迭代器共享某一具体算法，某一具体算法不依附于某一具体容器，迭代器起到一个中间媒介作用，把容器和算法连接起来。迭代器思维是编制通过泛型算法发展的必然结果，算法通过迭代器依次访问容器中的元素

1. 适配器是 （1） ，它依附于一个 （2） 容器上，它没有自己的 （3） 函数和 （4） 函数，而借用其实现类的对应函数。

三种STL容器适配器是 （5） 、 （6） 和 （7） 。

1. 下列的模板说明中,正确的是( )。
2. template<class T1,T2>;
3. template<typename T1,typename T2>;
4. template<class T1 ; class T2>;
5. template<typename T1,T2>;

1.为何map和set的插入删除效率比用其他序列容器高？

1. 三种SLT容器适配器是\_\_\_\_栈\_\_\_\_、\_\_\_队列\_\_\_\_\_和\_\_\_\_优先级队列\_\_\_\_.
2. 有两种容器适配器：\_\_\_第一类容器\_\_\_\_\_和\_\_\_\_近容器\_\_\_\_.STL不用new和delete,而用\_\_\_分配子\_\_\_\_\_实现各种控制内存分配和释放的方法。
3. 以下说法正确的是（A）。
4. list链表是一种双线性列表，只能顺序访问。
5. vector,list,queue都是序列性容器。
6. stack的特点是先进先出。
7. set是关联性容器，实现快速查找，允许重复值。

stl主要包括 容器，算法，迭代器三大部分，其中容器分为序列性容器，关联性容器和容器适配器三类。Stl容器包含了绝大多数数据结构，如：数组，链表，队列，堆，栈，树等。

1. 三种STL容器适配器是\_\_\_\_\_、\_\_\_\_\_\_、\_\_\_\_\_\_\_.

2、返回布尔值的函数对象称为\_\_\_\_\_\_,默认的是\_\_\_\_\_\_\_\_操作符。

3、如何向一个vector容器添加新元素？

答：1>.void push\_back(const T& x); //向量尾部增加一个元素x

2>.Iterator insert(iterator it, const T& x); //向量中某一元素前增加一个元素x

3>.Void insert(iterator it,int n.const T& x); //向量中某一元素前增加n个相同的元素x

4>.Void insert(iterator it,const\_iterator first,const\_iterator last);

1.成员函数end()得到容器（1）? ?的位置，而rend得到容器（2） ?的位置。算法通常返回??（3）? ?。?

2.三种STL容器适配器是??（1）? （2） ?和?（3）? ?。

3.适配器是?（1），它依附于一个（2）? ?容器上，它没有自己的?（3）? ?函数 和??（4）? ?函数，而借用其实现类的 对应函数。

4.返回布尔值的函数对象称为（1）?，默认的是?（2）? ?操作符。

5. hash\_map和map的区别在哪里?

6.vector和list的区别？

答案：1.（1）最后一个元素的后继位置（2）引用容器第一个元素的前导位置。实际上这是该容器前后反转之后end()? （3）迭代子?

2.（1）stack（栈）?（2）queue（队列）? （3）priority\_queue（优先级队列）

3.（1）不独立的? （2）顺序?（3）构造函数?（4）析构函数

4.（1）谓词（predicate）2.小于比较操作符“<”

5.构造函数。hash\_map需要hash函数，等于函数;map只需要比较函数(小于函数).存储结构。

hash\_map采用hash表存储，map一般采用 红黑树(RB Tree) 实现。因此其memory数据结构是不一样的。

6.vector和数组类似，拥有连续的内存空间，支持随机的存取。

list是由双向链表实现的，只能通过数组指针来进行数据访问。

**编程题**

1.补全代码，使用迭代器遍历

Vector<int>::iterator int int\_vec\_iter=int\_vec.begin();

While(\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_)

{cout<<\_\_\_\_\_\_\_\_\_\_<<”\t”;

\_\_\_\_\_\_\_\_\_\_;

}

2.利用字符串流，如何将整型数据50，转换成,50”？

Int（intargc,char \*argv）

{

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

//补全代码

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Return 0;

}

**填空题**

1.STL中三种容器适配器包括：\_\_\_\_\_\_\_,\_\_\_\_\_\_\_\_,\_\_\_\_\_\_\_\_\_。

2.STL主要包含容器，算法和\_\_\_\_\_\_\_。

3.成员函数rbegin()得到容器\_\_\_\_\_\_\_的位置，而rend得到容器\_\_\_\_\_\_的位置

4.STL中大量使用的是\_\_\_\_\_\_,因为\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_。

5.对下面的代码用函数对象，找出成绩范围在70 ~ 90之间的学生

struct Student {

string sn;

string name;

int grade;

};

vector<Student>v;

int n = count\_if( v.begin(), v.end(),);

6.返回布尔值的函数对象称为\_\_\_\_。

**选择题**

1.下列不是实现萃取技术的步骤为（）

A. 定义基本模板类B.模板特化

C定义基本容器类D.统一模板调用类编制

2.vector<int>v(6);

v.pop\_front(3)<<endl; //换为其他函数也行

结果输出为（）

A.3 B.4 C.5 D.2

**简答题**

1.STL容器分为几类？有什么区别？

2.什么是函数对象？用在什么地方？

1. traits技术依靠（ 显示模板特殊化 ）来把代码中因类型不同二发生变化的片段拖出来，用统一的接口来包装

2.在list容器中，操作函数void splice(iterator it , list& x)的功能是：（ A ）

A. 队列 x 所有元素插入迭代器指针 it 前，x 变成空队列

B. 队列 x 所有元素插入迭代器指针 it 后，x 变成空队列

C. 在迭代器指针 it 后插入元素x ，返回 x 迭代器指针

D. 在迭代器指针 it 后插入元素x ，返回尾元素之后的迭代器指针

STL的主要内容是什么？

1. 简述STL中迭代子与C++指针的关系与异同点。
2. 关联容器有哪四种？简单介绍它们是怎样组成的？各有什么特点？
3. 编程测试以双向队列（deque）为基础的容器适配器队列（queue）的主要功能和使用方法。

1、STL包含( )、（ )、( )三类。

1. 一个动态模板类，把与内存管理的部分进一步抽象，编译成代码，实现了内存的（）（）（）。
2. STL标准模板库编程的基本步骤：形成（）、取出（）、调用（）。
3. 五种主要迭代子类型为输入、输出、正向、双向、随机访问。
4. 2.三种STL容器适配器是栈、队列、优先级队列。

.三种STL容器适配器是­\_\_\_\_\_\_\_,\_\_\_\_\_\_\_,\_\_\_\_\_\_\_\_。

STL标准模板库编程的基本步骤\_\_\_\_\_\_,\_\_\_\_\_\_\_,\_\_\_\_\_\_。

选择题：

class Student {

char name[20];

int grade;

public:

Student ( char name[ ], int grade ) {

strcpy( this->name, name );

this->grade = grade;

}

( )

};

以下选项中能实现正确比较Student对象成绩大小的函数是（）。

1. bool operator>( int&value ) const { return grade > value; }
2. bool operator>( const int &value ) const { return grade > value; }
3. bool operator>( int&value ) { return grade > value; }
4. bool operator>( const int &value ) { return grade > value; }

-----B

Stl常用的数据结构不包过D

1. 栈 B.列表 C.队列 D.图

以下哪一项不属于序列性容器C

A.vector B.deque C.stack D.llist

2.简述STL中迭代子与C++指针的关系与异同点。

1. .实现traits技术的主要步骤有定义基本模板类、模板特化、统一模板调用类编制。
2. 什么是一元函数。什么是二元函数，它们的区别是什么？

3、补充完整代码，要求利用copy函数将v中的元素全部显示到屏幕上，每一个元素中间用“v”间隔。

int main()

{

vector<int> v;

for(int i=0;i<10;i++)

v.insert(i);

copy(\_\_\_\_,\_\_\_\_\_\_,\_\_\_\_\_);

return 0;

}

4、容器的成员函数rbegin()返回的反向迭代器指向的元素为（ ）

A．第一个元素 B.最后一个元素 C.最后一个元素下一个 D.5第一个元素前一个

(单选题)以下哪一个方法无法向Vector容器中插入数据？

A. insert() B.push\_front() C.push\_back() D.选项ABC均可插入数据

答案：B

(单选题)一下那种容器存储在内存的地址是完全连续的？

A. vector B. set C.deque D. list

答案：A

(单选题)以下关于deque的说法哪一个是正确的？

1. 当deque的size大于capacity的时候，deque会进行扩容
2. deque没有push\_front()方法
3. deque扩容的时候会申请一个新的空间，然后把数据全部迁移过去，再将原空间销毁
4. deque的内存可以是连续的，也可以是不连续的

答案：D

(填空题)请写出stl中的其中3种容器\_\_\_、\_\_\_、\_\_\_。

参考答案：vector，deque，list，set，multiset，map，multimap，stack，queue等（写出其中三个即可）

(填空题)容器有3类，分别是\_\_\_、\_\_\_、\_\_\_。

答案：序列性容器，关联式容器，容器适配器

(判断题)STL主要包含容器和迭代器两大部分

答案：错误，STL主要包含容器、算法和迭代器三大部分

(判断题)“using namespace std”属于名字空间域的运用

答案：正确

1. 实现traits技术的主要步骤不包括（）。
2. 定义基本模板类
3. 模板特化
4. 定义基本容器类
5. 统一模板调用类编制

STL提供的主要输出迭代器是。

1. 下列哪个是错误的?

int ia[ 7 ] = { 0, 1, 1, 2, 3, 5, 8 };

(a) vector< vector< int > > ivec;

(b) vector< int > ivec = { 0, 1, 1, 2, 3, 5, 8 };

(c) vector< int > ivec( ia, ia+7 );

(d) vector< string > svec = ivec;

(e) vector< string > svec( 10, string( "null" ));

2. 为何map和set的插入删除效率比用其他序列容器高？

1、适配器是 （1） ，它依附于一个 （2） 容器上，它没有自己的 （ 3） 函数和 （4） 函数，而借用其实现类的对应函数。

答案：（1）不独立的（2）顺序（3）构造函数（4）析构函数

2、 三种 STL 容器适配器是 （1） 、 （2） 和 （3） 。

答案：（1）stack（栈）（2）queue（队列）（3）priority\_queue （优先级队列）

3、什么是函数对象？它通常用在什么地方？

答：函数对象是一个类，它重载了函数调用操作符（ operator()）。该操作符封装了应该被实现为一个函数的操作。 典型情况下， 函数对象被作为实参传递给泛型算法， 以完全解决类型依赖性的问题。和“引用”一样， “函数对象”很少独立使用。函数对象与函数指针相比较有三个优点： 第一， 函数指针是间接引用， 不能作为内联函数，而函数对象可以，这样速度更快；第二，类有数据域，函数对象可以拥有任意数量的额外数据，用这些数据可以用来缓冲当前数据和结果，提高运行质量；第三，编译时对函数对象作类型检查。

1. 返回布尔值的函数对象称为\_\_\_\_\_\_, 默认的是\_\_\_\_\_\_\_\_操作符。

答案：谓词小于比较符号“<”

1. 五种主要迭代类型为\_\_\_\_\_，\_\_\_\_\_\_，\_\_\_\_\_\_，\_\_\_\_\_\_\_和\_\_\_\_\_\_\_.

答案：输入输出正向双向随机访问

2.STL中容器分哪两类？两类有何区别？

答：序列性容器，关联性容器。

序列性容器：按照线性排列储存某类型的集合，每个元素都有自己特定的位置

关联性容器：更注重快速和高效的检索数据的能力。这些容器根据键值来检索数据，键可以是值也可以是容器中某一成员。这一类中的成员在初始化后都是按一定顺序排好序的。

(1)迭代器最大的好处是\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

答案：可以使容器和算法分离；

(2)trsits技术在解决输入，输出参数类型不同时的步骤：

答案：（1）定义基本模板类（2）模板特化（3）统一模板调用类编制

2.迭代器提供了统一\_\_\_\_\_\_\_的方法，是通用算法坚实的技术基础。 访问容器元素

3.已知有数列｛1,3,5,7,9,8,6,4,2,0｝,则not1(bind1st(less<int>(),4))的执行结果有几个值？

A.2个 B.3个 C.4个 D.5个 D

一、为何每次insert之后，以前保存的iterator不会失效?

答:iterator这里就相当于指向节点的指针，内存没有变，指向内存的指针怎么会失效呢(当然被删除的那个元素本身已经失效了)。相对于vector来说，每一次删除和插入，指针都有可能失效，调用push\_back在尾部插入也是如此。因为为了保证内部数据的连续存放，iterator指向的那块内存在删除和插入过程中可能已经被其他内存覆盖或者内存已经被释放了。即使时push\_back的时候，容器内部空间可能不够，需要一块新的更大的内存，只有把以前的内存释放，申请新的更大的内存，复制已有的数据元素到新的内存，最后把需要插入的元素放到最后，那么以前的内存指针自然就不可用了。特别时在和find等算法在一起使用的时候，牢记这个原则:不要使用过期的iterator。

二、List的功能方法

实际上有两种List: 一种是基本的ArrayList,其优点在于随机访问元素，另一种是更强大的LinkedList,它并不是为快速随机访问设计的，而是具有一套更通用的方法。

List : 次序是List最重要的特点:它保证维护元素特定的顺序。List为Collection添加了许多方法，使得能够向List中间插入与移除元素(这只推荐LinkedList使用。)一个List可以生成ListIterator,使用它可以从两个方向遍历List,也可以从List中间插入和移除元素。

ArrayList : 由数组实现的List。允许对元素进行快速随机访问，但是向List中间插入与移除元素的速度很慢。ListIterator只应该用来由后向前遍历ArrayList,而不是用来插入和移除元素。因为那比LinkedList开销要大很多。

LinkedList : 对顺序访问进行了优化，向List中间插入与删除的开销并不大。随机访问则相对较慢。(使用ArrayList代替。)还具有下列方法:addFirst(), addLast(), getFirst(),getLast(), removeFirst() 和 removeLast(), 这些方法 (没有在任何接口或基类中定义过)使得LinkedList可以当作堆栈、队列和双向队列使用

2 vector插入删除和list有什么区别？

vector插入和删除数据，需要对现有数据进行复制移动，如果vector存储的对象很大或者构造函数很复杂，则开销较大，如果是简单的小数据，效率优于list。

容器算法迭代器分别是什么

容器是用来管理某一类对象的集合。C++ 提供了各种不同类型的容器，比如 deque、list、vector、map 等。

算法作用于容器。它们提供了执行各种操作的方式，包括对容器内容执行初始化、排序、搜索和转换等操作。

迭代器用于遍历对象集合的元素。这些集合可能是容器，也可能是容器的子集。