

江科大考研QQ：2962140400

100条经典C++笔试题目

江科大考研QQ：2962140400

■ 题目来源：

- 1、中兴、华为、慧通、英华达、微软亚洲技术中心等中外企业面试题目；
- 2、C++面试宝典(林锐《高质量编程第三版》)、C++ primer、Thinking in C++。

■ 说明：

- 1、部分C++语言面试题中可能会参杂部分和C相关的知识，为了保持题目的灵活性故保留，但选题最终还是会以C++语言题目为主体；
- 2、以上公司的面试题目已成为国内中小型企业公司出题模板；
- 3、由于本人的能力有限加上时间仓促，本课件肯定存在不足之处，恳请各位同学批评指正。

100条经典C++语言笔试题目

■ 题目类型

C++与C差异(1-18)

数据类型、关键字(19-37)

C++面向对象的特性(38-61)

程序阅读题(62-90)

编程练习(比C要难)(91-100)

江科大考研QQ : 2962140400

100条经典C++语言笔试题目

■ 1、C和C++中struct有什么区别？

【参考答案】

	Protection行为	能否定义函数
C	无	否，但可以有函数指针
C++	有，默认是private	可以

100条经典C++语言笔试题目

■ 2、C++中的struct和class有什么区别？

【参考答案】从语法上讲，class和struct做类型定义时只有两点区别：

（一）默认继承权限。如果不明确指定，来自class的继承按照private继承处理，来自struct的继承按照public继承处理；

（二）成员的默认访问权限。class的成员默认是private权限，struct默认是public权限。

除了这两点，class和struct基本就是一个东西。语法上没有任何其它区别。

100条经典C++语言笔试题目

- 3、如何判断一段程序是由C 编译程序还是由C++编译程序编译的？

【标准答案】

```
#ifdef __cplusplus
    cout<<"c++";
#else
    cout<<"c";
#endif
```

100条经典C++语言笔试题目

■ 4、C和C++有什么不同？

【参考答案】从机制上：c是面向过程的（但c也可以编写面向对象的程序）；c++是面向对象的，提供了类。但是，c++编写面向对象的程序比c容易。

从适用的方向：c适合要求代码体积小的，效率高的场合，如嵌入式；c++适合更上层的，复杂的；linux核心大部分是c写的，因为它是系统软件，效率要求极高。

从名称上也可以看出，c++比c多了+，说明c++是c的超集；那为什么不叫c+而叫c++呢，是因为c++比c来说扩充的东西太多了，所以就在c后面放上两个+；于是就成了c++。

C语言是结构化编程语言，C++是面向对象编程语言。

C++侧重于对象而不是过程，侧重于类的设计而不是逻辑的设计。

100条经典C++语言笔试题目

■ 5、“引用”与指针的区别是什么？

【参考答案】 指针通过某个指针变量指向一个对象后，对它所指向的变量间接操作。程序中使用指针，程序的可读性差；而引用本身就是目标变量的别名，对引用的操作就是对目标变量的操作。

100条经典C++语言笔试题目

■ 6、 class A

```
{  
    virtual void func1() ;  
    void func2();  
} ;  
class B: class A  
{  
    void func1(){cout << "fun1 in class B" << endl;}  
    virtual void func2(){cout << "fun2 in class B" << endl;}  
}
```

A, A中的func1和B中的func2都是虚函数

B, A中的func1和B中的func2都不是虚函数.

C, A中的func2是虚函数. , B中的func1不是虚函数.

D, A中的func2不是虚函数 , B中的func1是虚函数.

【标准答案】 A

100条经典C++语言笔试题目

- 7、`int id[sizeof(unsigned long)];`这个对吗？为什么？

【标准答案】 正确

**这个 `sizeof`是编译时运算符，编译时就确定了
可以看成和机器有关的常量。**

100条经典C++语言笔试题目

- 8、某文件中定义的静态全局变量(或称静态外部变量)其作用域是 ()?
- A.只限某个函数
B.本文件
C.跨文件
D.不限制作用域

【参考答案】B。静态全局变量则限制了其作用域，即只在定义该变量的源文件内有效，在同一源程序的其它源文件中不能使用它。由于静态全局变量的作用域局限于一个源文件内，只能为该源文件内的函数公用，因此可以避免在其它源文件中引起错误。

100条经典C++语言笔试题目

■ 9、C++函数中值的传递方式有哪几种？

【标准答案】 C++函数的三种传递方式为：值传递、指针传递和引用传递。

100条经典C++语言笔试题目

- 10、对于一个频繁使用的短小函数,在C语言中应用什么实现,在C++中应用什么实现?

【标准答案】 c用宏定义 , c++用inline

100条经典C++语言笔试题目

■ 11、引用与指针有什么区别？

【参考答案】

1) 引用必

须被初始化，指针不必。

2) 引用初始化以后不能被改变，指针可以改变所指的对象。

3) 不存在指向空值的引用，但是存在指向空值的指针。

100条经典C++语言笔试题目

■ 12、C++中virtual与inline的含义分别是什么？

【参考答案】在基类成员函数的声明前加上virtual关键字，意味着将该成员函数声明为虚函数。inline与函数的定义体放在一起，使该函数称为内联。inline是一种用于实现的关键字，而不是用于声明的关键字。

虚函数的特点；如果希望派生类能够重新定义基类的方法，则在基类中将该方法定义为虚方法，这样可以启用动态联编。

内联函数的特点；使用内联函数的目的是为了提高函数的运行效率。内联函数体的代码不能过长，因为内联函数省去调用函数的时间是以代码膨胀为代价的。内联函数不能包含循环语句，因为执行循环语句要比调用函数的开销大。

100条经典C++语言笔试题目

- 13、VC中，编译工具条内的Debug与Release选项是什么含义？
【参考答案】 Debug 通常称为调试版本，它包含调试信息，并且不作任何优化，便于程序员调试程序。Release 称为发布版本，它往往是进行了各种优化，使得程序在代码大小和运行速度上都是最优的，以便用户很好地使用。Debug带有大量的调试代码，运行时需要相应的运行库，发布模式程序紧凑不含有调试代码和信息，直接可以运行（如果不需要运行库）

100条经典C++语言笔试题目

■ 14、函数assert的用法？

【参考答案】断言assert是仅在debug版本起作用的宏，用于检查“不应该”发生的情况。程序员可以把assert看成一个在任何系统状态下都可以安全使用的无害测试手段。

100条经典C++语言笔试题目

■ 15、const 与 #define 的比较，const有什么优点？

【参考答案】（1）const 常量有数据类型，而宏常量没有数据类型。编译器可以对前者进行类型安全检查。而对后者只进行字符替换，没有类型安全检查，并且在字符替换可能会产生意料不到的错误（边际效应）。（2）有些集成化的调试工具可以对const 常量进行调试，但是不能对宏常量进行调试。

100条经典C++语言笔试题目

■ 16、请你谈谈引用和指针的区别。

【参考答案】（1）引用被创建的同时必须被初始化（指针则可以在任何时候被初始化）。

（2）不能有 NULL 引用，引用必须与合法的存储单元关联（指针则可以是 NULL）。

（3）一旦引用被初始化，就不能改变引用的关系（指针则可以随时改变所指的对象）。江科大考研QQ：2962140400

100条经典C++语言笔试题目

■ 17、有了 malloc/free 为什么还要 new/delete ？

【参考答案】 malloc 与 free 是 C++/C 语言的标准库函数，new/delete 是 C++的运算符。它们都可用于申请动态内存和释放内存。对于非内部数据类型的对象而言，光用 malloc/free 无法满足动态对象的要求。对象在创建的同时要自动执行构造函数，对象在消亡之前要自动执行析构函数。由于 malloc/free 是库函数而不是运算符，不在编译器控制权限之内，不能够把执行构造函数和析构函数的任务强加于 malloc/free。因此 C++语言需要一个能完成动态内存分配和初始化工作的运算符 new，以及一个能完成清理与释放内存工作的运算符 delete。注意 new/delete 不是库函数。

100条经典C++语言笔试题目

- 18、如果在申请动态内存时找不到足够大的内存块，malloc 和 new 将返回 NULL 指针，宣告内存申请失败。你是怎么处理内存耗尽的？

【参考答案】（1）判断指针是否为 NULL，如果是则马上用 return 语句终止本函数。（2）判断指针是否为 NULL，如果是则马上用 exit(1)终止整个程序的运行（3）为 new 和 malloc 设置异常处理函数。例如 Visual C++可以用 _set_new_handler 函数为 new 设置用户自己定义的异常处理函数，也可以让 malloc 享用与 new 相同的异常处理函数。

100条经典C++语言笔试题目

■ 19、C++是不是类型安全的？

【参考答案】不是。两个不同类型的指针之间可以强制转换（用 `reinterpret cast`）。

100条经典C++语言笔试题目

■ 20、 const 符号常量；

(1) const char *p

(2) char const *p

(3) char * const p

说明上面三种描述的区别；

【参考答案】 (1)p是一个指向const char的指针，p是可以改变指向的，但是p指向的值是不能改变的； (2)p指向的恰好是一个指向const的char的普通指针； (3)p是一个指针，这个指针是指向char的const指针。 (1)和(2)的定义是一样的。

100条经典C++语言笔试题目

- 21、用C++写个程序，如何判断一个操作系统是16位还是32位的？

【标准答案】 定义一个指针p，打印出sizeof(p),如果结果是4，则表示该操作系统是32位，打印结果是2，表示是16位。

100条经典C++语言笔试题目

- 22、.用C++写个程序，如何判断一个操作系统是16位还是32位的？不能用sizeof()函数。

【参考答案】

```
int a = ~0;
if( a>65536 )
{
    cout<<"32 bit"<<endl;
}
else
{
    cout<<"16 bit"<<endl;
}
```

100条经典C++语言笔试题目

- 23、`void * (* (*fp1)(int))[10];`
`float (*(* fp2)(int,int,int))(int);`
`int (* (* fp3)())[10]();`
分别表示什么意思？

【标准答案】

`(*fp1)(int))[10];`

fp1是一个指针，指向一个函数，这个函数的参数为int型，函数的返回值是一个指针，这个指针指向一个数组，这个数组有10个元素，每个元素是一个void*型指针。

1.`void * (*`

100条经典C++语言笔试题目

■ 2.float (*(* fp2)(int,int,int))(int);

fp2是一个指针，指向一个函数，这个函数的参数为3个int型，函数的返回值是一个指针，这个指针指向一个函数，这个函数的参数为int型，函数的返回值是float型。

100条经典C++语言笔试题目

■ 3.int (* (* fp3)())[10]();

fp3是一个指针，指向一个函数，这个函数的参数为空，函数的返回值是一个指针，这个指针指向一个数组，这个数组有10个元素，每个元素是一个指针，指向一个函数，这个函数的参数为空，函数的返回值是int型。

100条经典C++语言笔试题目

- 24、多态类中的虚函数表是Compile-Time，还是Run-Time时建立的？

【标准答案】 虚拟函数表是在编译期就建立了,各个虚拟函数这时被组织成了一个虚拟函数的入口地址的数组.而对象的隐藏成员--虚拟函数表指针是在运行期--也就是构造函数被调用时进行初始化的,这是实现多态的关键。

100条经典C++语言笔试题目

■ 25、错误的转义字符是 ()

A. '\091'

B. '\\'

C. '\0'

D. '\'

【标准答案】 A

100条经典C++语言笔试题目

- 26、若数组名作实参而指针变量作形参，函数调用实参传给形参的是 ()
- A.数组的长度 B.数组第一个元素的值 C.数组所有元素的值 D.数组第一个元素的地址

【标准答案】 D江科大考研QQ：2962140400

100条经典C++语言笔试题目

- 27、变量的指针含意是指变量的 ()
A.值 B.地址 C.存储 D.名字

【标准答案】 B

100条经典C++语言笔试题目

■ 28、内存的分配方式有几种？

【参考答案】一、从静态存储区域分配。内存在程序编译的时候就已经分配好，这块内存在程序的整个运行期间都存在。例如全局变量。二、在栈上创建。在执行函数时，函数内局部变量的存储单元都可以在栈上创建，函数执行结束时这些存储单元自动被释放。栈内存分配运算内置于处理器的指令集中，效率很高，但是分配的内存容量有限。

三、从堆上分配，亦称动态内存分配。程序在运行的时候用 `malloc` 或 `new` 申请任意多少的内存，程序员自己负责在何时用 `free` 或 `delete` 释放内存。动态内存的生存期由我们决定，使用非常灵活，但问题也最多。

100条经典C++语言笔试题目

- 29、float a,b,c ,问等式 $(a+b)+c==(b+a)+c$ 和 $(a+b)+c==(a+c)+b$ 能否成立？

【参考答案】两者都不行。在比较float或double时，不能简单地比较。由于计算误差，相等的概率很低。应判断两数之差是否落在区间 $(-e,e)$ 内。这个e应比浮点数的精度大一个数量级。

100条经典C++语言笔试题目

- 30、全局变量和局部变量有什么区别？是怎么实现的？操作系统和编译器是怎么知道的？

【参考答案】生命周期不同：全局变量随主程序创建和创建，随主程序销毁而销毁；局部变量在局部函数内部，甚至局部循环体等内部存在，退出就不存在；内存中分配在全局数据区。

使用方式不同：通过声明后全局变量程序的各个部分都可以用到；局部变量只能在局部使用；分配在栈区。操作系统和编译器通过内存分配的位置来知道的，全局变量分配在全局数据段并且在程序开始运行的时候被加载。局部变量则分配在堆栈里面。

100条经典C++语言笔试题目

■ 31、Heap与stack的差别

【参考答案】 Heap是堆，stack是栈。

Stack的空间由操作系统自动分配/释放，Heap上的空间手动分配/释放。

Stack空间有限，Heap是很大的自由存储区

C中的malloc函数分配的内存空间即在堆上,C++中对应的是new操作符。

程序在编译期对变量和函数分配内存都在栈上进行,且程序运行过程中函数调用时参数的传递也在栈上进行

■ 江科大考研QQ：2962140400

100条经典C++语言笔试题目

- 32、In C++, what does "explicit" mean? what does "protected" mean?

【标准答案】c++中的explicit关键字用来修饰类的构造函数，表明该构造函数是显式的，在某些情况下，我们要求类的使用者必须显示调用类的构造函数时就需要使用explicit,反之默认类型转换可能会造成无法预期的问题。protected控制的是一个函数对一个类的成员（包括成员变量及成员方法）的访问权限。protected成员只有该类的成员函数及其派生类的成员函数可以访问。

100条经典C++语言笔试题目

- 33、重复多次fclose一个打开过一次的FILE *fp指针会有什么结果，并请解释。

【参考答案】考察点：导致文件描述符结构中指针指向的内存被重复释放，进而导致一些不可预期的异常。

100条经典C++语言笔试题目

- 34、为什么数组名作为参数，会改变数组的内容，而其它类型如int却不会改变变量的值？

【参考答案】当数组名作为参数时，传递的实际上是地址。而其他类型如int作为参数时，由于函数参数值实质上是实参的一份拷贝，被调函数内部对形参的改变并不影响实参的值。

100条经典C++语言笔试题目

- 35、你觉得如果不使用常量，直接在程序中填写数字或字符串，将会有什么麻烦？

【参考答案】（1）程序的可读性（可理解性）变差。程序员自己会忘记那些数字或字符串是什么意思，用户则更加不知它们从何处来、表示什么。

（2）在程序的很多地方输入同样的数字或字符串，难保不发生书写错误。

（3）如果要修改数字或字符串，则会在很多地方改动，既麻烦又容易出错。

100条经典C++语言笔试题目

- 36、为什么需要使用堆，使用堆空间的原因？

【参考答案】直到运行时才知道一个对象需要多少内存空间；不知道对象的生存期到底有多长。

100条经典C++语言笔试题目

■ 37、const关键字？有哪些作用

【参考答案】const关键字至少有下列n个作用：

- (1) 欲阻止一个变量被改变，可以使用const关键字。在定义该const变量时，通常需要对它进行初始化，因为以后就没有机会再去改变它了；**
- (2) 对指针来说，可以指定指针本身为const，也可以指定指针所指的数为const，或二者同时指定为const；**
- (3) 在一个函数声明中，const可以修饰形参，表明它是一个输入参数，在函数内部不能改变其值；**
- (4) 对于类的成员函数，若指定其为const类型，则表明其是一个常函数，不能修改类的成员变量；**
- (5) 对于类的成员函数，有时候必须指定其返回值为const类型，以使得其返回值不为“左值”。**

剖析：这个题可以考查面试者对程序设计知识的掌握程度是初级、中级还是比较深入，没有一定的知识广度和深度，不可能对这个问题给出全面的解答。大多数人只能回答出static和const关键字的部分功能。

100条经典C++语言笔试题目

- 38、是不是一个父类写了一个virtual 函数，如果子类覆盖它的函数不加virtual ,也能实现多态？

【参考答案】virtual修饰符会被隐形继承的。virtual可加可不加。子类的空间里有父类的所有变量(static除外)。同一个函数只存在一个实体(inline除外)。子类覆盖它的函数不加virtual ,也能实现多态。在子类的空间里，有父类的私有变量。私有变量不能直接访问。

100条经典C++语言笔试题目

■ 39、面向对象的三个基本特征，并简单叙述之？

【参考答案】

1. 封装：将

客观事物抽象成类，每个类对自身的数据和方法实行

protection(private,protected,public)

2. 继承：广义的

继承有三种实现形式：实现继承（指使用基类的属性和方法而无需额外编码的能力）、可视继承（子窗体使用父窗体的外观和实现代码）、接口继承（仅使用属性和方法，实现滞后到子类实现）。前两种（类继承）和后一种（对象组合=>接口继承以及纯虚函数）构成了功能复用的两种方式。

3. 多态：是将父对

象设置成为和一个或更多的与他的子对象相等的技术，赋值之后，父对象就可以根据当前赋值给它的子对象的特性以不同的方式运作。简单的说，就是一句话：允许将子类类型的指针赋值给父类类型的指针。

100条经典C++语言笔试题目

- 40、重载 (overload)、重写(override , 有的书也叫做“覆盖”)、重定义 (redefinition) 的区别？

【标准答案】

名称	名字空间	区别
重载	同一名字空间	是指允许存在多个同名函数，而这些函数的参数表不同。
重定义/隐藏	不同名字空间	用于继承，派生类与基类的函数同名，屏蔽基类的函数
重写/覆盖	不同名字空间	用于继承，子类重新定义父类虚函数的方法

100条经典C++语言笔试题目

■ 41、多态的作用？

【参考答案】主要是两个：1. 隐藏实现细节，使得代码能够模块化；扩展代码模块，实现代码重用；2. 接口重用：为了类在继承和派生的时候，保证使用家族中任一类的实例的某一属性时的正确调用。

100条经典C++语言笔试题目

- 42、当一个类A 中没有声明任何成员变量与成员函数,这时 sizeof(A)的值是多少,如果不是零,请解释一下编译器为什么没有让它为零。

【标准答案】 sizeof(A)= 1 ;

100条经典C++语言笔试题目

- 43、如果ClassA中定义并实现虚函数`int func(void)`，ClassB中也实现该函数，那么上述变量`a->func()`将调用哪个类里面的函数？如果`int func(void)`不是虚函数，情况又如何？为什么？

【参考答案】第一问调用的是B的。第二问调用A的。虚函数的一个典型应用，虚函数只能借助于指针或者引用来达到多态的效果。

100条经典C++语言笔试题目

- 44、 C++里面是不是所有的动作都是main()引起的？如果不是，请举例。

【参考答案】 比如全局变量的初始化，就不是由main函数引起的。

举例：

```
class A{};  
A a; //a的构造函数限执行  
int main() {}
```

100条经典C++语言笔试题目

■ 45、 内联函数在编译时是否做参数类型检查

【参考答案】内联函数要做参数类型检查，这是内联函数跟宏相比的优势。

100条经典C++语言笔试题目

■ 46、请讲一讲析构函数和虚函数的用法和作用？

【参考答案】析构函数是特殊的类成员函数，它没有返回类型，没有参数，不能随意调用，也没有重载，只有在类对象的生命期结束的时候，由系统自动调用。有适放内存空间的作用。虚函数是C++多态的一种表现，使用虚函数，我们可以灵活的进行动态绑定，当然是以一定的开销为代价。

100条经典C++语言笔试题目

■ 47、“new”in c++ is a:

- A. library function like malloc in c B. key word C. operator
D. none of the above

【参考答案】C。malloc是库函数，不在编译器控制范围之内；
new是运算符，在编译器控制范围之内。

调用malloc时，从堆中申请内存；调用new时，从堆中申请内存
并为内存调用构造函数。

100条经典C++语言笔试题目

- 48、对于C++中类(class) 与结构(struct)的描述正确的为:
- A,类中的成员默认是private的,当是可以声明public,private 和 protected,结构中定义的成员默认的都是public;
 - B,结构中不允许定义成员函数,当是类中可以定义成员函数;
 - C,结构实例使用malloc() 动态创建,类对象使用new 操作符动态分配内存;
 - D,结构和类对象都必须使用new 创建;
 - E,结构中不可以定义虚函数,当是类中可以定义虚函数.
 - F,结构不可以存在继承关系,当是类可以存在继承关系.

【标准答案】 A,D

100条经典C++语言笔试题目

■ 49、两个互相独立的类:ClassA 和 ClassB,都各自定义了非静态的公有成员函数 PublicFunc() 和非静态的私有成员函数 PrivateFunc();现在要在 ClassA 中增加定义一个成员函数ClassA::AdditionalPunction(ClassA a,ClassB b);则可以在AdditionalPunction(ClassAx,ClassB y)的实现部分(函数功能体内部)出现的合法的表达是最全的是:

A,x.PrivateFunc();x.PublicFunc();y.PrivateFunc();y.PublicFunc();

B,x.PrivateFunc();x.PublicFunc();y.PublicFunc();

C,x.PrivateFunc();y.PrivateFunc();y.PublicFunc();

D,x.PublicFunc();y.PublicFunc();

【标准答案】 B

100条经典C++语言笔试题目

■ 50、C++程序下列说法正确的有:

A,对调用的虚函数和模板类都进行迟后编译.

B,基类与子类中函数如果要构成虚函数,除了要求在基 类中用 virtual 声名,而且必须名字相同且参数类型相同返回类型相同。

C,重载的类成员函数都必须要有:或者返回类型不同,或者参数数目不同,或者参数序列的类型不同.

D,静态成员函数和内联函数不能是虚函数,友员函数和构造函数也不能是虚函数,但是析构函数可以是虚函数.

【标准答案】 A

100条经典C++语言笔试题目

■ 51、在C++中有没有纯虚构造函数？

【标准答案】构造函数不能是虚的。只能有虚的析构函数。

100条经典C++语言笔试题目

■ 52、下面的 throw表达式哪些是错误的？

(a) `class exceptionType { };`

`throw exceptionType { };`

(b) `enum mathErr { overflow, underflow, zeroDivide };`

`throw zeroDivide();`

【标准答案】

(a) `class exceptionType { };`

`throw exceptionType();`

(b) `enum mathErr { overflow, underflow, zeroDivide };`

`throw zeroDivide;`

100条经典C++语言笔试题目

■ 53、谈谈你是怎么认识c++中的模板的？

【参考答案】模板使程序员能够快速建立具有类型安全的类库集合和函数集合，它的实现，方便了大规模的软件开发。（结合stl更好）

100条经典C++语言笔试题目

- 54、在c++的一个类中声明一个static成员变量有没有用？

【参考答案】在C++类的成员变量被声明为static（称为静态成员变量），意味着它为该类的所有实例所共享，也就是说当某个类的实例修改了该静态成员变量，也就是说不管创建多少对象，static修饰的变量只占有一块内存。其修改值为该类的其它所有实例所见；而类的静态成员函数也只能访问静态成员（变量或函数）。static是加了访问控制的全局变量，不被继承。

100条经典C++语言笔试题目

■ 55、C++中为什么用模板类。

【参考答案】(1)可用来创建动态增长和减小的数据结构 (2)它是类型无关的，因此具有很高的可复用性。(3)它在编译时而不是运行时检查数据类型，保证了类型安全 (4)它是平台无关的，可移植性 (5)可用于基本数据类型

100条经典C++语言笔试题目

■ 56、函数模板与类模板有什么区别？

【参考答案】函数模板的实例化是由编译程序在处理函数调用时自动完成的，而类模板的实例化必须由程序员在程序中显式地指定。

100条经典C++语言笔试题目

- 58、请你谈谈你在类中如何使用const的。

【参考答案】有时我们希望某些常量只在类中有效。由于#define 定义的宏常量是全局的，不能达到目的，于是想当然地觉得应该用const 修饰数据成员来实现。const 数据成员的确是存在的，但其含义却不是我们所期望的。const 数据成员只在某个对象生存期内是常量，而对于整个类而言却是可变的，因为类可以创建多个对象，不同的对象其 const 数据成员的值可以不同。不能在类声明中初始化const 数据成员。

const 数据成员的初始化只能在类构造函数的初始化表中进行。

100条经典C++语言笔试题目

- 59、函数重载，我们靠什么来区分调用的那个函数？靠返回值判断可以不可以？

【参考答案】如果同名函数的参数不同（包括类型、顺序不同），那么容易区别出它们是不同的。如果同名函数仅仅是返回值类型不同，有时可以区分，有时却不能。例如：

```
void Function(void);
```

```
int Function (void);
```

上述两个函数，第一个没有返回值，第二个的返回值是 int 类型。如果这样调用函数：

```
int x = Function ();
```

则可以判断出 Function 是第二个函数。问题是在 C++/C 程序中，我们可以忽略函数的返回值。在这种情况下，编译器和程序员都不知道哪个 Function 函数被调用。所以只能靠参数而不能靠返回值类型的不同来区分重载函数。

100条经典C++语言笔试题目

■ 60、所有的运算符都能重载吗？

【参考答案】不能被重载的运算符

在 C++运算符集合中，有一些运算符是不允许被重载的。这种限制是出于安全方面的考虑，可防止错误和混乱。（1）不能改变 C++内部数据类型（如 int,float 等）的运算符。（2）不能重载 `'.'`，因为 `'.'`在类中对任何成员都有意义，已经成为标准用法。（3）不能重载目前 C++运算符集合中没有的符号，如`#`,`@`,`$`等。原因有两点，一是难以理解，二是难以确定优先级。

（4）对已经存在的运算符进行重载时，不能改变优先级规则，否则将引起混乱。

100条经典C++语言笔试题目

- 61、基类的析构函数不是虚函数，会带来什么问题？

【参考答案】派生类的析构函数用不上，会造成资源的泄漏。

100条经典C++语言笔试题目

■ 62、main 函数执行以前，还会执行什么代码？

【参考答案】全局对象的构造函数会在main 函数之前执行。

100条经典C++语言笔试题目

- 63、 There are two int variables: a and b, don't use "if", "? :", "switch" or other judgement statements, find out the biggest one of the two numbers.

【标准答案】 $((a + b) + \text{abs}(a - b)) / 2$

100条经典C++语言笔试题目

- 64、如何打印出当前源文件的文件名以及源文件的当前行号？

【标准答案】

```
cout << __FILE__;
```

```
cout<<__LINE__;
```

__FILE__和__LINE__是系统预定义宏，这种宏并不是在某个文件中定义的，而是由编译器定义的。（c也有）

100条经典C++语言笔试题目

- 65、下面两种if语句判断方式。请问哪种写法更好？为什么？

int n;

if (n == 10) // 第一种判断方式

if (10 == n) // 第二种判断方式

【参考答案】这是一个风格问题，第二种方式如果少了个=号，编译时就会报错，减少了出错的可能行，可以检测出是否少了=。

100条经典C++语言笔试题目

■ 66、写出运行结果:

```
{// test1
    char str[] = "world"; cout << sizeof(str) << ": ";
    char *p = str; cout << sizeof(p) << ": ";
    char i = 10; cout << sizeof(i) << ": ";
    void *pp = malloc(10); cout << sizeof(p) << endl;
}
```

【标准答案】 6 : 4 : 1 : 4

100条经典C++语言笔试题目

- 67、在不用第三方参数的情况下，交换两个参数的值。

【参考答案】

```
a = a + b;
```

```
b = a - b;
```

```
a = a - b;
```

100条经典C++语言笔试题目

- 68、以下代码如果有错，请改正，并写出输出结果？
- ```
void
main(void)
{
 int nArrLength(400), i = 546;
 for (int i = 0; i < 999999999999; i++); cout <<
 nArrLength << endl;
 cout << i << endl;
}
```



# 100条经典C++语言笔试题目

## 【标准答案】

void

```
main(void)
```

```
{
```

```
 int nArrLength(400), i = 546; /*主要是考看对C++的基础知识是否
```

了解这里的int nArrLength(400)是对整数的定义，当然，明名上有问题，这里是故意这样的，但是，最好是变量名改为 ....[还是您自己看着办了]\*/

```
 for (int i = 0; i < 999999999999; i++); /*这里是考对变量越界理解，同
```

时....，所以，999...应该改为 ~((int)0),也就是整数中0取反

```
nArrLength << endl; // 这里输出 400
```

```
 cout << i << endl; // 这里输出 546
```

```
}
```

# 100条经典C++语言笔试题目

- 69、`int i = 5, b = 7;`  
`cout << (i+++b) << endl;`  
不用调试，请说出，以上代码在gcc编译过后的执行结果！  
【标准答案】结果是12。

# 100条经典C++语言笔试题目

- 70、写一个能做左值的函数(方法有很多)。如：`max(x, y) += 2874 + 55;`

`drwline(x, y)++;`

**【参考答案】**

```
int &max(int & x, int & y)
```

```
{
```

```
 return x > y? x : y;
```

```
}
```

```
int x = 55, y = 77;
```

```
max(x, y) += 12 + 11; // 此时 y = 92;
```

```
cout << "x = " << x << " "; y = " << y << endl; // 输出 x = 55; y = 92;
```

# 100条经典C++语言笔试题目

■ 71、

```
class human
{
public:
 ~human()
 {
 cout << "human over..... " << endl;
 }
 void Disp()
 {
 cout << "human disp " << endl;
 }
};
class man : public human
{
public:
 ~man()
 {
 cout << "man over....." << endl;
 }
 void Disp()
 {
 cout << "man disp " << endl;
 }
};
int main()
{
 human* p = new man;
 p->Disp();
 delete p;
 return 0;
}
```

【标准答案】

human over .....

# 100条经典C++语言笔试题目

- 72、下面的函数实现在一个固定的数上加上一个数，有什么错误，改正：

```
int add_n(int n)
{
 static int i=100;
 i+=n;
 return i;
}
```

**【标准答案】** 因为static使得i的值会保留上次的值。以后的i会一直更新，使得第二次调用出现 出现错误，去掉static就可了

# 100条经典C++语言笔试题目

## ■ 73、写出打印结果

```
unsigned short array[]={1,2,3,4,5,6,7};
```

```
int i = 3;
```

```
*(array + i) =
```

**【标准答案】 4**

# 100条经典C++语言笔试题目

- 74、写一个函数计算当参数为n(n很大)时的值  $1-2+3-4+5-6+7+\dots+n$ 。

**【参考答案】**

fn(long n)

{

if(n <= 0)

{

printf("error: n must > 0);

exit(1);

}

if(0 == n % 2)

return (n / 2) \* (-1);

else

return (n / 2) \* (-1) + n;

}

long

# 100条经典C++语言笔试题目

- 75、字符指针、浮点数指针、以及函数指针这三种类型的变量哪个占用的内存最大？为什么？

**【参考答案】**指针变量也占用内存单元，而且所有指针变量占用内存单元的数量都是相同的。就是说，不管是指向何种对象的指针变量，它们占用内存的字节数都是一样的，并且要足够把程序中所能用到的最大地址表示出来（通常是一个机器字长）。



# 100条经典C++语言笔试题目

## ■ 76、以下三条输出语句分别输出什么？

```
char str1[] = "abc";
```

```
char str2[] = "abc";
```

```
const char str3[] = "abc";
```

```
const char str4[] = "abc";
```

```
const char* str5 = "abc";
```

```
const char* str6 = "abc";
```

```
cout << boolalpha << (str1==str2) << endl; // 输出什么？
```

```
cout << boolalpha << (str3==str4) << endl; // 输出什么？
```

```
cout << boolalpha << (str5==str6) << endl; // 输出什么？
```

**【参考答案】**分别输出false,false,true。str1和str2都是字符数组，每个都有其自己的存储区，它们的值则是各存储区首地址，不等；str3和str4同上，只是按const语义，它们所指向的数据区不能修改。str5和str6并非数组而是字符指针，并不分配存储区，其后的“abc”以常量形式存于静态数据区，而它们自己仅是指向该区首地址的指针，相等。

# 100条经典C++语言笔试题目

■ 77、以下代码有什么问题？

```
(true?1:"1") << endl;
```

```
cout <<
```

**【参考答案】**三元表达式 “?:” 问号后面的两个操作数必须为同一类型。

# 100条经典C++语言笔试题目

- 78、以下代码能够编译通过吗，为什么？

```
unsigned int const size1 = 2;
```

```
char str1[size1];
```

```
unsigned int temp = 0;
```

```
cin >> temp;
```

```
unsigned int const size2 = temp;
```

```
char str2[size2];
```

**【标准答案】** str2定义出错，size2非编译器期间常量，而数组定义要求长度必须为编译期常量。

# 100条经典C++语言笔试题目

- 79、以下代码中的输出语句输出0吗，为什么？

```
struct CLS
{
 int m_i;
 CLS(int i) : m_i(i) {}
 CLS()
 {
 CLS(0);
 }
};

CLS obj;

cout << obj.m_i << endl;
```

**【标准答案】**不能。在默认构造函数内部再调用带参的构造函数属用户行为而非编译器行为，亦即仅执行函数调用，而不会执行其后的初始化表达式。只有在生成对象时，初始化表达式才会随相应的构造函数一起调用。

# 100条经典C++语言笔试题目

■ 80、 How do you code an infinite loop in Cplus plus ?

**【参考答案】** while(1){} or for(;;)

# 100条经典C++语言笔试题目

- 81、What are the values of a, b, and c after the following instructions:

```
int a=5, b=7, c;
```

```
c = a+++b;
```

**【标准答案】** a=6,b=7,c=12

# 100条经典C++语言笔试题目

- 82、在排序方法中，关键码比较次数与记录地初始排列无关的是（ ）
- A. Shell排序  
直接插入排序
- B. 归并排序
- C.
- D. 选择排序

**【标准答案】 D**

# 100条经典C++语言笔试题目

## ■ 83、代码

```
void func()
```

```
{
```

```
 static int val;
```

```
 ...
```

**} 中，变量val的内存地址位于：**

**A. 已初始化数据段    B. 未初始化数据段**

**C. 堆                  D. 栈**

**【标准答案】 A**



# 100条经典C++语言笔试题目

- 84、一个栈的入栈序列是A , B , C , D , E , 则栈的不可能的输出序列是 ( )
- A、 EDCBA ;    B、 DECBA ;  
C、 DCEAB ;    D、 ABCDE
- 【标准答案】 C**

# 100条经典C++语言笔试题目

- 85、写出判断ABCD四个表达式的是否正确, 若正确, 写出经过表达式中 a的值。

`int a = 4;`

(A) `a += (a++);` (B) `a += (++a);`

(C) `(a++) += a;` (D) `(++a) += (a++);`

a = ?

**【参考答案】** C错误, 左侧不是一个有效变量, 不能赋值, 可改为 `(++a) += a;` 改后答案依次为9,10,10,11

# 100条经典C++语言笔试题目

- 86、请你谈谈你是如何使用return语句的。

**【参考答案】**（1）return 语句不可返回指向“栈内存”的“指针”或者“引用”，因为该内存在函数体结束时被自动销毁。

（2）要搞清楚返回的究竟是“值”、“指针”还是“引用”。

（3）如果函数返回值是一个对象，要考虑 return 语句的效率。

# 100条经典C++语言笔试题目

- 87、①return String(s1 + s2); 和②String temp(s1 + s2);return temp; 一样吗？

**【参考答案】** ①这是临时对象的语法，表示“创建一个临时对象并返回它”。②将发生三件事。首先，temp 对象被创建，同时完成初始化；然后拷贝构造函数把 temp 拷贝到保存返回值的外部存储单元中；最后，temp 在函数结束时被销毁（调用析构函数）。然而“创建一个临时对象并返回它”的过程是不同的，编译器直接把临时对象创建并初始化在外部存储单元中，省去了拷贝和析构的化费，提高了效率。

# 100条经典C++语言笔试题目

## ■ 88、下列程序的运行结果是

```
#include <iostream>
```

```
const char *str = "vermeer";
```

```
int main()
```

```
{
```

```
 const char *pstr = str;
```

```
 cout << "The address of pstr is: " << pstr << endl;
```

```
}
```

**【标准答案】** The address of pstr is: vermeer

# 100条经典C++语言笔试题目

## ■ 89、下列程序输出结果是

```
inline void max_out(int val1, int val2)
{
 cout << (val1 > val2) ? val1 : val2;
}
int main()
{
 int ix = 10, jx = 20;
 cout << "The larger of " << ix;
 cout << ", " << jx << " is ";
 max_out(ix, jx);
 cout << endl;
}
```

**【标准答案】** The larger of 10, 20 is 0 , 为什么不是20呢？问题在于输出操作符的优先级高于条件操作符 所以输出 val1和 val2 比较结果的 true/false 。

# 100条经典C++语言笔试题目

■ 90、`int max( int *ia, int sz );`  
`int max( int *, int = 10 );`

算函数重载？还是重复声明？

**【标准答案】**如果在两个函数的参数表中只有缺省实参不同则第二个声明被视为第一个的重复声明。

# 100条经典C++语言笔试题目

- 91、请编写一个 C 函数，该函数给出一个字节中被置 1 的位的个数。

## 【参考答案】

```
unsigned int TestAsOne0(char log)
{
 int i;
 unsigned int num=0, val;
 for(i=0; i<8; i++)
 {
 val = log >> i; //移位
 val &= 0x01; //与1相与
 if(val)
 num++;
 }
 return num;
}
```



# 100条经典C++语言笔试题目

- 92、编写一个函数，函数接收一个字符串,是由十六进制数组成的一组字符串,函数的功能是把接到的这组字符串转换成十进制数字.并将十进制数字返回。

【参考答案】

```
BOOL HexToDec(LPCTSTR shex,int& idc)
{
 int i,mid;
 int len = strlen(shex);
 if(len>8)
 return FALSE;
 mid = 0; idc = 0;
 for(i=0;i<len;i++)
 {
 if(shex[i]>='0'&&shex[i]<='9')
 mid = shex[i]-'0';
 else if(shex[i]>='a'&&shex[i]<='f')
 mid = shex[i] -'a' +10;
 else if(shex[i]>='A'&&shex[i]<='F')
 mid = shex[i] -'A' +10;
 else
 return FALSE;

 mid <<= ((len-i-1)<<2); // 移位表示变为2的n次方倍
 idc =idc+mid;
 }
 return TRUE;
}
```

# 100条经典C++语言笔试题目

- 93、输入一个字符串，将其逆序后输出。

## 【参考答案】

```
void main()
```

```
{
```

```
 char
```

```
 a[50];memset(a,0,sizeof(a));
```

```
 int i=0,j;
```

```
 char t;
```

```
 cin.getline(a,50,'\n');
```

```
 for(i=0,j=strlen(a)-
```

```
 1;i<strlen(a)/2;i++,j--)
```

```
 {
```

```
 t=a[i];
```

```
 a[i]=a[j];
```

```
 a[j]=t;
```

```
 }
```

```
 cout<<a<<endl;
```

```
}
```

//第二种

```
string str;
```

```
cin>>str;
```

```
str.replace;
```

```
cout<<str;
```

# 100条经典C++语言笔试题目

- 94、编写一个算法`frequency`，统计在一个输入字符串中各个不同字符出现的频度。用适当的测试数据来验证这个算法。



# 100条经典C++语言笔试题目

- 95、假设以数组Q[m]存放循环队列中的元素, 同时以rear和length分别指示环形队列中的队尾位置和队列中所含元素的个数。试给出该循环队列的队空条件和队满条件, 并写出相应的插入(enqueue)和删除(dlqueue)元素的操作。

# 100条经典C++语言笔试题目

## 【参考答案】循环队列类定义

```
#include <assert.h>
```

```
template <class Type> class Queue { //循环队列的类定义
```

```
public:
```

```
 Queue (int=10);
```

```
 ~Queue () { delete [] elements; }
```

```
 void EnQueue (Type & item);
```

```
 Type DeQueue ();
```

```
 Type GetFront ();
```

```
 void MakeEmpty () { length = 0; } //置空队列
```

```
 int IsEmpty () const { return length == 0; } //判队列空否
```

```
 int IsFull () const { return length == maxSize; } //判队列满否
```

```
private:
```

```
 int rear, length; //队尾指针和队列长度
```

```
 Type *elements; //存放队列元素的数组
```

```
 int maxSize; //队列最大可容纳元素个数
```

```
};
```

# 100条经典C++语言笔试题目

```
template <class Type>
Queue<Type>:: Queue (int sz) : rear (maxSize-1), length (0), maxSize (sz)
{//建立一个最大具有maxSize个元素的空队列。
 elements = new Type[maxSize]; //创建队列空间
 assert (elements != 0); //断言: 动态存储分配成功与否
}
template<class Type>
void Queue<Type> :: EnQueue (Type &item)
{
 assert (! IsFull ()); //判队列是否不满, 满则出错处理
 length++; //长度加1
 rear = (rear +1) % maxSize; //队尾位置进1
 elements[rear] = item; //进队列
}
template<class Type>
Type Queue<Type> :: DeQueue ()
{
 assert (! IsEmpty ()); //判断队列是否不空, 空则出错处理
 length--; //队列长度减1
 return elements[(rear-length+maxSize) % maxSize]; //返回原队头元素值
}
template<class Type>
Type Queue<Type> :: GetFront ()
{
 assert (! IsEmpty ());
 return elements[(rear-length+1+maxSize) % maxSize]; //返回队头元素值
}
```



# 100条经典C++语言笔试题目

- 96、已知 $A[n]$ 为整数数组，试写出实现下列运算的递归算法：
  - (1) 求数组 $A$ 中的最大整数。
  - (2) 求 $n$ 个整数的和。
  - (3) 求 $n$ 个整数的平均值。



# 100条经典C++语言笔试题目

## 【参考答案】

```
#include <iostream.h>
class RecurveArray
{
private: int *Elements; //数组类声明
 int ArraySize; //数组指针
 int CurrentSize; //数组尺寸
 RecurveArray (int MaxSize =10) : //当前已有数组元素个数 public :
 ArraySize (MaxSize), Elements (new int[MaxSize]){ }
 ~RecurveArray () { delete [] Elements; }
 void InputArray(); //输入数组的内容
 int MaxKey (int n); //求最大值
 int Sum (int n); //求数组元素之和
 float Average (int n); //求数组元素的平均值
};

void RecurveArray :: InputArray ()
{
 //输入数组的内容
 cout << "Input the number of Array: \n";
 for (int i = 0; i < ArraySize; i++) cin >> Elements[i];
}
```

# 100条经典C++语言笔试题目

```
int RecurveArray :: MaxKey (int n)
{
 //递归求最大值
 if (n == 1) return Elements[0];
 int temp = MaxKey (n - 1);
 if (Elements[n-1] > temp) return Elements[n-1];
 else return temp;
}

int RecurveArray :: Sum (int n) {
 //递归求数组之和
 if (n == 1) return Elements[0];
 else return Elements[n-1] + Sum (n-1);
}

float RecurveArray :: Average (int n) {
 //递归求数组的
 //平均值
 if (n == 1) return (float) Elements[0];
 else return ((float) Elements[n-1] + (n - 1) * Average (n - 1)) / n;
}

int main (int argc, char* argv []) {
 int size = -1;
 cout << "No. of the Elements : ";
 while (size < 1) cin >> size;
 RecurveArray ra (size);
 ra.InputArray();
 cout<< "\nThe max is: " << ra.MaxKey (ra.MaxSize) << endl;
 cout<< "\nThe sum is: " << ra.Sum (ra.MaxSize) << endl;
 cout<< "\nthe avr is: " << ra.Average (ra.MaxSize) << endl;
 return 0;
}
```

# 100条经典C++语言笔试题目

■ 97、已知 $p$ 为单链表的表头指针,链表中存储的都是整型数据,试写出实现下列运算的递归算法:

- (1) 求链表中的最大整数。
- (2) 求链表的结点个数。
- (3) 求所有整数的平均值。

# 100条经典C++语言笔试题目

## 【标准答案】

```
#include <iostream.h> //定义在头文件"RecurveList.h"中
class List;
class ListNode { //链表结点类
friend class List;
private:
 int data; //结点数据
 ListNode *link; //结点指针
 ListNode (const int item) : data(item), link(NULL) { } //构造函数
};
class List { //链表类
private:
 ListNode *first, current;
 int Max (ListNode *f);
 int Num (ListNode *f);
 float Avg (ListNode *f, int& n);
public:
 List () : first(NULL), current (NULL) { } //构造函数
 ~List (){ } //析构函数
 ListNode* NewNode (const int item); //创建链表结点, 其值为item
 void NewList (const int retvalue); //建立链表, 以输入retvalue结束
 void PrintList (); //输出链表所有结点数据
 int GetMax () { return Max (first); } //求链表所有数据的最大值
 int GetNum () { return Num (first); } //求链表中数据个数
 float GetAvg () { return Avg (first); } //求链表所有数据的平均值
};
```

# 100条经典C++语言笔试题目

```
ListNode* List :: NewNode (const int item) { //创建新链表结点
 ListNode *newnode = new ListNode (item);
 return newnode;
}

void List :: NewList (const int retvalue) { //建立链表, 以输入retvalue结束
 first = NULL; int value; ListNode *q;
 cout << "Input your data:\n"; //提示
 cin >> value; //输入
 while (value != retvalue)
 {
 //输入有效
 q = NewNode (value); //建立包含value的新结点
 if (first == NULL) first = current = q; //空表时, 新结点成为链表第一个
 //结点

 else { current->link = q; current = q; } //非空表时, 新结点链入链尾
 cin >> value;
 //再输入
 }
 current->link = NULL; //链尾封闭
}

void List :: PrintList ()
{
 //输出链表
 cout << "\nThe List is : \n";
 ListNode *p = first;
 while (p != NULL) {
 cout << p->data << ' '; p = p->link; }
 cout << '\n';
 }
}
```

# 100条经典C++语言笔试题目

```
int List :: Max (ListNode *f)
{
 //递归算法：求链表中的最大值
 if (f ->link == NULL) return f ->data; //递归结束条件
 int temp = Max (f ->link); //在当前结点的后继链表中求最大值
 if (f ->data > temp)
 return f ->data; //如果当前结点的值还要大, 返回当前结点值
 else return temp; //否则返回后继链表中的最大值
}

int List :: Num (ListNode *f)
{
 //递归算法：求链表中结点个数
 if (f == NULL) return 0; //空表, 返回0
 return 1+ Num (f ->link); //否则, 返回后继链表结点个数加1
}

float List :: Avg (ListNode *f , int& n)
{
 //递归算法：求链表中所有元素的平均值
 if (f ->link == NULL) //链表中只有一个结点, 递归结束条件
 {
 n = 1; return (float) (f ->data);
 }
 else
 { float Sum = Avg (f ->link, n) * n; n++; return (f ->data + Sum) / n; }
}
```



# 100条经典C++语言笔试题目

```
#include "RecurveList.h"
```

//定义在主文件中

```
int main (int argc, char* argv[])
```

```
{
```

```
 List test; int finished;
```

```
 cout << "输入建表结束标志数据：" ;
```

```
 cin >> finished;
```

//输入建表结束标志数据

```
 test.NewList (finished);
```

//建立链表

```
 test.PrintList ();
```

//打印链表

```
 cout << "\nThe Max is : " << test.GetMax ();
```

```
 cout << "\nThe Num is : " << test.GetNum ();
```

```
 cout << "\nThe Ave is : " << test.GetAve () << "\n";
```

```
 printf ("Hello World!\n");
```

```
 return 0;
```

```
}
```

# 100条经典C++语言笔试题目

- 98、字符串的替换操作replace (String &s, String &t, String &v)  
是指：

$$A[i][j] = \begin{cases} C[i][j], & \text{当 } i \geq j \text{ 时} \\ C[j][i], & \text{当 } i < j \text{ 时} \end{cases}$$

若t是s的子串，则用串v替换串t在串s中的所有出现；若t不是s的子串，则串s不变。例如，若串s为“aabbabcbaabaaacbab”，串t为“bab”，串v为“abdc”，则执行replace操作后，串s中的结果为“aababdbccbaabaaacabdc”。试利用字符串的基本运算实现这个替换操作。



# 100条经典C++语言笔试题目

## 【参考答案】

```
String & String :: Replace (String & t, String &v)
{
 if ((int id = Find (t)) == -1) //没有找到，当前字符串不改，返回
 { cout << "The (replace) operation failed." << endl; return *this; }
 String temp(ch);//用当前串建立一个空的临时字符串
 ch[0] = '\0'; curLen = 0; //当前串作为结果串，初始为空
 int j, k = 0, l; //存放结果串的指针
 while (id != -1) {
 for (j = 0; j < id; j++) ch[k++] = temp.ch[j];
 curLen += id + v.curLen; //修改结果串连接后的长度
 if (curLen <= maxLen) l = v.curLen; //确定替换串v传送字符数l
 else { l = curLen - maxLen; curLen = maxLen; }
 for (j = 0; j < l; j++) ch[k++] = v.ch[j];
 //连接替换串v到结果串ch后面
 if (curLen == maxLen) break; //字符串超出范围
 for (j = id + t.curLen; j < temp.curLen; j++)
 temp.ch[j - id - t.curLen] = temp.ch[j]; //删改原来的字符串
 temp.curLen -= (id + t.curLen);
 id = temp.Find (t); }
 return *this;
}
```

# 100条经典C++语言笔试题目

- 99、试编写一个求解Josephus问题的函数。用整数序列1, 2, 3, .....,  $n$ 表示顺序围坐在圆桌周围的人，并采用数组表示作为求解过程中使用的数据结构。然后使用 $n = 9, s = 1, m = 5$ ，以及 $n = 9, s = 1, m = 0$ ，或者 $n = 9, s = 1, m = 10$ 作为输入数据，检查你的程序的正确性和健壮性。

# 100条经典C++语言笔试题目

## 【参考答案】

```
void Josephus(int A[], int n, s, m){
 int i, j, k, tmp;
 if (m == 0) {
 cout << "m = 0是无效的参数 !" << endl;
 return;
 }
 for (i = 0; i < n; i++) A[i] = i + 1; /*初始化，执行n次*/
 i = s - 1; /*报名起始位置*/
 for (k = n; k > 1; k--) { /*逐个出局，执行n-1次*/
 if (i == k) i = 0;
 i = (i + m - 1) % k; /*寻找出局位置*/
 if (i != k-1) {
 tmp = A[i]; /*出局者交换到第k-1位置*/
 for (j = i; j < k-1; j++) A[j] = A[j+1];
 A[k-1] = tmp;
 }
 }
 for (k = 0; k < n / 2; k++) { /*全部逆置, 得到出局序列*/
 tmp = A[k]; A[k] = A[n-k+1]; A[n-k+1] = tmp;
 }
}
```

# 100条经典C++语言笔试题目

- 100、编写类 String 的构造函数、析构函数和赋值函数已知类 String 的原型为：

```
class String
{
 public:
 String(const char *str = NULL); // 普通构造函数
 String(const String &other); // 拷贝构造函数
 ~String(void); // 析构函数
 String & operate =(const String &other); // 赋值函数
 private:
 char *m_data; // 用于保存字符串
};
```

请编写 String 的上述 4 个函数。

# 100条经典C++语言笔试题目

## ■ 【标准答案】

**// String 的析构函数**

**String::~~String(void)**

**{**

**delete [] m\_data;**

**// 由于 m\_data 是内部数据类型，也可以写成 delete m\_data;**

**}**

# 100条经典C++语言笔试题目

**// String 的普通构造函数**

```
String::String(const char *str)
```

```
{
```

```
if(str==NULL)
```

```
{
```

```
 m_data = new char[1]; // 若能加 NULL 判断则更好
```

```
 *m_data = '\0';
```

```
}
```

```
else
```

```
{
```

```
 int length = strlen(str);
```

```
 m_data = new char[length+1];
```

```
 strcpy(m_data, str);
```

```
}
```

# 100条经典C++语言笔试题目

## ■ // 拷贝构造函数

```
String::String(const String &other)
{
 int length = strlen(other.m_data);
 m_data = new char[length+1]; // 若能加 NULL 判断则更好
 strcpy(m_data, other.m_data);
}
```

# 100条经典C++语言笔试题目

## ■ // 赋值函数

```
String & String::operate =(const String &other)
{
 if(this == &other)
 return *this;
 delete [] m_data;
 int length = strlen(other.m_data);
 m_data = new char[length+1];
 strcpy(m_data, other.m_data);
 return *this;
}
```