

程序员

2016年下半年试题

本试卷为：**样式2**

样式1：适用于模拟考试，所有答案在最后面。

样式2：适用于复习，每道题的题目和答案在一起。

本试卷由**跨步软考**提供

我们目前提供的免费服务有：

- 手机APP刷题
- 网页版刷题
- 真题pdf版下载
- 视频课程下载
- 其他资料下载

更多免费服务请访问我们的官网：<http://kuabu.xyz>

你也可以关注我们的微信公众号：**跨步软考**

如果您发现试题有错误，您可以通过以下方式联系我们

- 客服邮箱：service@kuabu.xyz
- 备用邮箱：kuabu@outlook.com
- 您也可以在微信公众号后台留言

本文档所有权归**跨步软考**(kuabu.xyz)，您可以传播甚至修改本文档，但是必须标明出自“**跨步软考 (kuabu.xyz)**”

上午综合试卷

第1题：某质量技术监督部门为检测某企业生产的某个批次的化妆品含铅量是否超标，通常宜采用（ ）的方法。

- A. 普查
- B. 查有无合格证
- C. 抽样检查
- D. 查阅有关单据

参考答案：C

第2题：某企业资料室员工张某和王某负责向系统中录入一批图书信息（如：图书编号、书名、作者、出版社，联系方式等信息）。要求在保证质量的前提下，尽可能高效率地完成任任务。对于如下：A~D四种工作方式，（ ）方式比较恰当。

- A. 张某独立完成图书信息的录入，王某抽查
- B. 张某独立完成图书信息的录入，王某逐条核对
- C. 张某和王某各录一半图书信息，再交叉逐条核对
- D. 张某和王某分工协作，分别录入图书信息的不同字段，再核对并合并在一起

参考答案：C

第3题：在Excel中，假设单元格A1、A2、A3和A4的值分别为23、45、36、18，单元格B1、B2、B3、B4的值分别为29、38、25、21，在单元格C1中输入“=SUM(MAX(A1:A4),MIN(B1:B4))”（输入内容不含引号）并按Enter后，C1单元格显示的内容为（ ）。

- A. 44
- B. 66
- C. 74
- D. 84

参考答案：B

第4题：在Excel中，若在单元格A6中输入“=Sheet1!D5 +Sheet2!B4:D4+Sheet3!A2:G2”，则该公式（ ）。

- A. 共引用了2张工作表的5个单元格的数据
- B. 共引用了2张工作表的11个单元格的数据
- C. 共引用了3张工作表的5个单元格的数据
- D. 共引用了3张工作表的11个单元格的数据

参考答案：D

第5题：“http://www.x123.arts.hk”中的“arts.hk”代表的是（ ）。

- A. 韩国的商业机构
- B. 香港的商业机构
- C. 韩国的艺术机构
- D. 香港的艺术机构

参考答案：D

第6题：在汇编指令中，操作数在某寄存器中的寻址方式称为（ ）寻址。

- A. 直接
- B. 变址
- C. 寄存器
- D. 寄存器间接

参考答案：C

第7题：计算机系统中，虚拟存储体系由（ ）两级存储器构成。

- A. 主存—辅存
- B. 寄存器—Cache
- C. 寄存器—主存
- D. Cache—主存

参考答案：A

第8题：程序计数器（PC）是（ ）中的寄存器。

- A. 运算器
- B. 控制器
- C. Cache
- D. I/O设备

参考答案：B

第9题：中断向量提供（ ）。

- A. 外设的接口地址
- B. 待传送数据的起始和终止地址
- C. 主程序的断点地址
- D. 中断服务程序入口地址

参考答案：D

第10题：在计算机系统中总线宽度分为地址总线宽度和数据总线宽度。若计算机中地址总线的宽度为32位，则最多允许直接访问主存储器（ ）的物理空间。

- A. 40MB
- B. 4GB
- C. 40GB
- D. 400GB

参考答案：B

第11题：为了提高计算机磁盘存取效率，通常可以（ ）。

- A. 用磁盘格式化程序定期对ROM进行碎片整理
- B. 用磁盘碎片整理程序定期对内存进行碎片整理
- C. 用磁盘碎片整理程序定期对磁盘进行碎片整理
- D. 用磁盘格式化程序定期对磁盘进行碎片整理

参考答案：C

第12题：商标权保护的对象是指（ ）。

- A. 商品
- B. 商标
- C. 已使用商标
- D. 注册商标

参考答案：D

第13题：两名以上的申请人分别就同样的软件发明创造申请专利时，（ ）可取得专利权。

- A. 最先发明的人
- B. 最先申请的人
- C. 所有申请的人
- D. 最先使用人

参考答案：B

第14题：自然界的声信号一般都是多种频率声音的复合信号，用来描述组成复合信号的频率范围的参数被称为信号的（ ）。

- A. 带宽
- B. 音域
- C. 响度
- D. 频度

参考答案：A

解析：

信号的带宽是指该信号所包含的各种不同频率成分所占据的频率范围。这是百度对带宽的解释，所以本题应该选带宽。

第15题：以下媒体文件格式中，（ ）是视频文件格式。

- A. WAV
- B. BMP
- C. MOV
- D. MP3

参考答案 : C

第16题 : 使用150DPI的扫描分辨率扫描一幅3x4英寸的彩色照片, 得到原始的24位真彩色图像的数据量是 () Byte。

- A. 1800
- B. 90000
- C. 270000
- D. 810000

参考答案 : D

解析 :

$$150 \times 3 \times 150 \times 4 \times 24 / 8 = 810000$$

第17题 : 下列病毒中, 属于后门类病毒的是 ()。

- A. Trojan.Lmir.PSW.60
- B. Hack.Nether.Client
- C. Macro.word97
- D. Script.Redlof

参考答案 : A

解析 :

一般地, 根据计算机病毒的发作方式和原理, 在病毒名称前面加上相应的代码以表示该病毒的制作原理和发作方式。

例如, 以Trojan.开头的病毒一般为木马病毒, 以VBS.、JS.、Script.开头的病毒一般为脚本病毒, 以Worm.开头的一般为蠕虫病毒等。

第18题 : 安全的电子邮件协议为 ()。

- A. MIME
- B. PGP
- C. POP3
- D. SMTP

参考答案：B

解析：

PGP(Pretty Good Privacy)，是一个基于RSA公钥加密体系的邮件加密软件，提供一种安全的通讯方式。

安全电子邮件协议S/MIME(Secure/Multi—purpose Internet Mail Extensions)

第19题：在浮点表示格式中，数的精度是由（ ）的位数决定的。

- A. 尾数
- B. 阶码
- C. 数符
- D. 阶符

参考答案：A

第20题：目前的小型和微型计算机系统中普遍采用的字母与字符编码是（ ）。

- A. BCD码
- B. 海明码
- C. ASCII码
- D. 补码

参考答案：C

第21题：已知 $x = -53/64$ ，若采用8位定点机器码表示，则【x】原=（21），【x】补=（22）。

- A. 01101101
- B. 11101010
- C. 11100010
- D. 01100011

参考答案：B

第22题：已知 $x = -53/64$ ，若采用8位定点机器码表示，则【x】原=（21），【x】补=（22）。

- A. 11000011
- B. 11101010
- C. 10011110
- D. 10010110

参考答案 : D

第23题 : 操作系统通过 () 来组织和管理外存中的信息。

- A. 字处理程序
- B. 设备驱动程序
- C. 文件目录和目录项
- D. 语言翻译程序

参考答案 : C

第24题 : 下列操作系统中, () 保留了网络系统的全部功能, 并具有透明性、可靠性和高性能等特性。

- A. 批处理操作系统
- B. 分时操作系统
- C. 分布式操作系统
- D. 实时操作系统

参考答案 : C

第25题 : 在进程状态转换过程中, 可能会引起进程阻塞的原因是 ()。

- A. 时间片到
- B. 执行V操作
- C. I/O完成
- D. 执行P操作

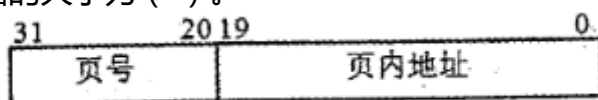
参考答案 : D

第26题：假设系统有 $n(n \geq 3)$ 个进程共享资源 R ，且资源 R 的可用数3。若采用PV操作，则相应的信号量 S 的取值范围应为（ ）。

- A. $-1 \sim n-1$
- B. $-3 \sim 3$
- C. $-(n-3) \sim 3$
- D. $-(n-1) \sim 1$

参考答案：C

第27题：某分页存储管理系统中的地址结构如下图所示。若系统以字节编址，则该系统每个页面的大小为（ ）。



- A. 4096KB
- B. 1MB
- C. 2MB
- D. 4MB

参考答案：B

第28题：以下关于解释方式下运行程序的叙述中，错误的是（ ）。

- A. 先将高级语言程序转换为字节码，再由解释器运行字节码
- B. 由解释器直接分析并执行高级语言程序代码
- C. 先将高级语言程序转换为某种中间代码，再由解释器运行中间代码
- D. 先将高级语言程序转换为机器语言，再由解释器运行机器语言代码

参考答案：C

第29题：编写C程序时通常为了提高可读性而加入注释，注释并不参与程序的运行过程。通常，编译程序在（ ）阶段就会删除源程序中的注释。

- A. 词法分析
- B. 语法分析
- C. 语义分析

D. 代码优化

参考答案：D

第30题：某C语言程序中有表达式 $x\%m$ （即 x 被 m 除取余数），其中， x 为浮点型变量， m 为整型非0常量，则该程序在（30）时会报错，该错误属于（31）错误。

- A. 编译
- B. 预处理
- C. 编辑
- D. 运行

参考答案：A

第31题：某C语言程序中有表达式 $x\%m$ （即 x 被 m 除取余数），其中， x 为浮点型变量， m 为整型非0常量，则该程序在（30）时会报错，该错误属于（31）错误。

- A. 逻辑
- B. 语法
- C. 语义
- D. 运行

参考答案：C

解析：

程序代码中的错误可分为语法错误和语义错误。程序语言的语法表述的是语言的形式，或者说是语言的样子和结构。程序语言还有更重要的一个方面，就是附着于语言结构上的语义。语义揭示了程序本身的含义、施加于语言结构上的限制或者要执行的动作。程序语言的语义分为静态语义和动态语义。编译时进行的是静态语义的分析，主要包括：检查语言结构的语义是否正确，即是否结构正确的句子所表示的意思也合法；执行规定的语义动作，如表达式的求值、符号表的填写、中间代码的生成等。整除取余运算符“%”的有效运算对象是两个整数。

在形式上，只要“%”的两个运算对象存在，其语法就是正确的；

在语义上，“%”的运算对象中有浮点数则不符合整除取余运算的含义。因此，这是运算对象与运算符的类型不匹配错误，属于静态语义错误，在编译阶段可以发现该错误。

第32题：在单CPU计算机系统中，完成相同功能的递归程序比非递归程序（ ）。

- A. 运行时间更短，占用内存空间更少
- B. 运行时间更长，占用内存空间更多

C. 运行时间更短, 占用内存空间更多

D. 运行时间更长, 占用内存空间更少

参考答案: B

第33题: 已知函数f(). G()的定义如下所示, 调用函数f时传递给形参x的值是5。若g(a)采用引用调用(call by reference)方式传递参数, 则函数f的返回值为 (33) ; 若g(a)采用值调用 (.call by value)的方式传递参数, 则函数f的返回值为 (34)。其中, 表达式 "X > > 1" 的含义是将x的值右移1位, 相当于x=2。

f(int x)

```
int a = x >> 1;  
g(a);  
return a + x;
```

g(int x)

```
x = x * (x + 1);  
return;
```

A. 35

B. 32

C. 11

D. 7

参考答案: C

第34题: 已知函数f(). G()的定义如下所示, 调用函数f时传递给形参x的值是5。若g(a)采用引用调用(call by reference)方式传递参数, 则函数f的返回值为 (33) ; 若g(a)采用值调用 (.call by value)的方式传递参数, 则函数f的返回值为 (34)。其中, 表达式 "X > > 1" 的含义是将x的值右移1位, 相当于x=2。

f(int x)

```
int a = x >> 1;  
g(a);  
return a + x;
```

g(int x)

```
x = x * (x + 1);  
return;
```

A. 35

B. 32

C. 11

D. 7

参考答案: B

解析:

在值调用方式下, g函数中调用函数f时传递给形参x的值是5, 也就是说在函数g中的x和

f函数的值已经没有关系了, 因此, 返回函数g中的 $x=5*(5+1)=30$, 再执行f中的"a+x"运算后得到32, 因此空 (34) 应填入的值为32。

第35题: 设数组 $a[0..n-1, 0..m-1]$ ($n>1, m>1$) 中的元素以行为主序存放, 每个元素占用4个存储单元, 则数组元素 $a[i, j]$ ($0 \leq i < n, 0 \leq j < m$) 的存储位置相对于数组空间首地址的偏移量为 ()。

- A. $(j*m+i)*4$
- B. $(i*m+j)*4$
- C. $(j*n+i)*4$
- D. $(i*n+j)*4$

参考答案: B

第36题: 线性表采用单循环链表存储的主要特点是 ()。

- A. 从表中任一结点出发都能遍历整个链表
- B. 可直接获取指定结点的直接前驱和直接后继结点
- C. 在进行删除操作后, 能保证链表不断开
- D. 与单链表相比, 更节省存储空间

参考答案: A

第37题: 若某线性表长度为n且采用顺序存储方式, 则运算速度最快的操作是 ()。

- A. 查找与给定值相匹配的元素的位置
- B. 查找并返回第i个元素的值 ($1 \leq i \leq n$)
- C. 删除第i个元素 ($1 \leq i \leq n$)
- D. 在第i个元素 ($1 \leq i \leq n$) 之前插入一个新元素

参考答案: B

第38题: 设元素a、b、c、d依次进入一个初始为空的栈, 则不可能通过合法的栈操作序列得到 ()。

- A. a b c d
- B. b a d c

C. c a d b

D. d c b a

参考答案 : C

第39题: 若要求对大小为 n 的数组进行排序的时间复杂度为 $O(n\log_2 n)$, 且是稳定的 (即如果待排序的序列中两个数据元素具有相同的值, 在排序前后它们的相对位置不变), 则可选的排序方法是 ()。

A. 快速排序

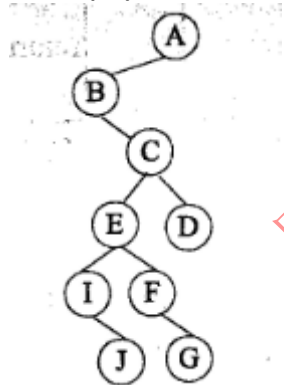
B. 归并排序

C. 堆排序

D. 冒泡排序

参考答案 : B

第40题: 对于一般的树结构, 可以采用孩子-兄弟表示法, 即每个结点设置两个指针域, 一个指针 (左指针) 指示当前结点的第一个孩子结点, 另一个指针 (右指针) 指示当前结点的下一个兄弟结点。某树的孩子-兄弟表示如下图所示。以下关于结点D与E的关系的叙述中, 正确的是 ()。



A. 结点D与结点E是兄弟

B. 结点D与结点E的祖父结点

C. 结点E的父结点与结点D的父结点是兄弟

D. 结点E的父结点与结点D是兄弟

参考答案 : D

第41题: 搜索引擎会通过日志文件把用户每次检索使用的所有查询串都记录下来, 每个查询串的长度不超过255字节。假设目前有一千万个查询记录 (重复度比较高, 其实互异的查询

串不超过三百万个;显然,一个查询串的重复度越高,说明查询它的用户越多,也就是越热门)。现要统计最热门的10个查询串,且要求使用的内存不能超过1GB。以下各方法中,可行且效率最高的方法是()。

- A. 将一千万个查询串存入数组并进行快速排序,再统计其中每个查询串重复的次数
- B. 将一千万个查询串存入数组并进行堆排序,再统计其中每个查询串重复的次数
- C. 利用哈希表保存所有的查询串并记下每个查询串的重复次数,再利用小根堆选出重复次数最多的10个查询串
- D. 利用哈希表保存所有的查询串并记下每个查询串的重复次数,再利用大根堆选出重复次数最多的10个查询串

参考答案:D

第42题:设某无向图的顶点个数为 n ,则该图最多(42)条边;若将该图用邻接矩阵存储,则矩阵的行数和列数分别为(43)。

- A. n
- B. $n*(n-1)/2$
- C. $n*(n+1)/2$
- D. $n*n$

参考答案:B

第43题:设某无向图的顶点个数为 n ,则该图最多(42)条边;若将该图用邻接矩阵存储,则矩阵的行数和列数分别为(43)。

- A. n 、 n
- B. n 、 $n-1$
- C. $n-1$ 、 n
- D. $n+1$ 、 n

参考答案:A

第44题:在面向对象方法中,(44)定义了父类和子类的概念。子类在原有父类接口的基础上,用适合于自己要求的实现去置换父类中的相应实现称为(45)。

- A. 封装
- B. 继承

C. 覆盖 (重置)

D. 多态

参考答案 : B

第45题 : 在面向对象方法中, (44) 定义了父类和子类的概念。子类在原有父类接口的基础上, 用适合于自己要求的实现去置换父类中的相应实现称为 (45)。

A. 封装

B. 继承

C. 覆盖 (重置)

D. 多态

参考答案 : C

第46题 : 在UML用例图中, 参与者表示 ()。

A. 人、硬件或其他系统可以扮演的角色

B. 可以完成多种动作的相同用户

C. 不管角色的实际物理用户

D. 带接口的物理系统或者硬件设计

参考答案 : A

第47题 : UML中关联是一个结构关系, 描述了一组链。两个类之间 ()。

A. 不能有多个关联

B. 可以有多个由不同角色标识的关联

C. 必须有一个关联

D. 多个关联必须聚合成一个关联

参考答案 : B

第48题 : 创建型设计模式抽象了实例化过程, 帮助一个系统独立于如何创建、组合和表示的那些对象。以下 () 模式是创建型模式。

A. 组合(Composite)

- B. 构建器(Builder)
- C. 桥接(Bridge)
- D. 策略 (Strategy)

参考答案：B

第49题：如果模块A的三个处理都对同一数据结构操作，则模块A的内聚类型是（ ）。

- A. 逻辑内聚
- B. 时间内聚
- C. 功能内聚
- D. 通信内聚

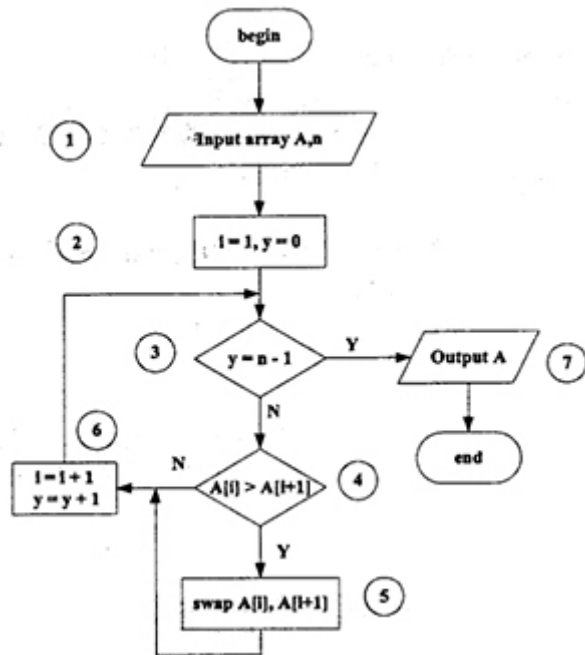
参考答案：D

第50题：修改现有软件系统的设计文档和代码以增强可读性，这种行为属于（ ）维护。

- A. 正确性
- B. 适应性
- C. 完善性
- D. 预防性

参考答案：C

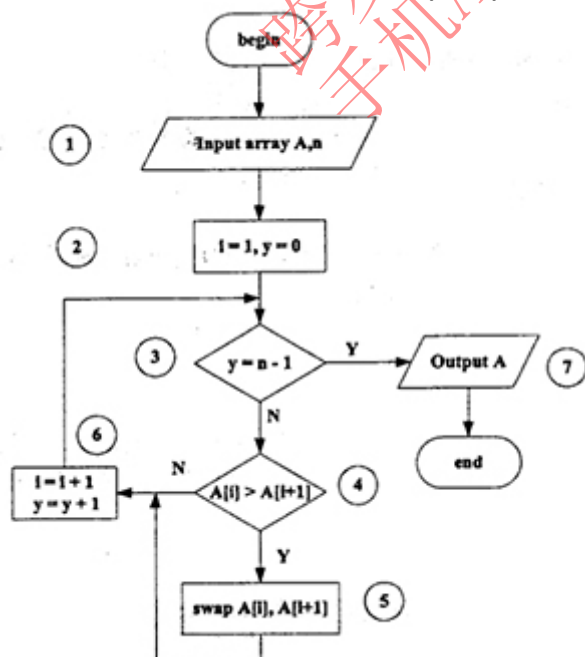
第51题：对下面流程图用白盒测试方法进行测试，要满足路径覆盖，至少需要（ 51 ）个测试用例。白盒测试方法主要用于（ 52 ）。



- A. 3
- B. 4
- C. 5
- D. 6

参考答案：B

第52题：对下面流程图用白盒测试方法进行测试，要满足路径覆盖，至少需要（51）个测试用例。白盒测试方法主要用于（52）。



A. 单元测试

- B. 集成测试
- C. 系统测试
- D. 接收测试

参考答案：A

第53题：以下关于软件测试的叙述中，不正确的是（ ）。

- A. 社会对软件的依赖和对软件质量要求的提高是软件测试行业发展的基础
- B. 系统建设的出资方应提供测试方案
- C. 对软件质量的定性判断需要测试提供的定量数据支撑
- D. 测试团队与开发团队的共同目标是改善软件质量

参考答案：B

第54题：为在网上搜索内容而输入关键词时，常可能打错别字。当系统显示没有匹配项后，有些系统还会向你推荐某些关键词。为实现这种推荐，采用（ ）方法更有效。

- A. 聘请一批专家，对每个常用关键词给出一些易错成的词
- B. 聘请一批专家，对每个常输错的关键词给出纠正后的词
- C. 查阅有关的词典，对每个常用的关键词找出易错成的词
- D. 利用系统内记录的用户操作找出用户纠错词的对应关系

参考答案：D

第55题：以下关于编程的叙述中，不正确的是（ ）。

- A. 当程序员正沉浸于算法设计和编程实现时，不希望被干扰或被打断
- B. 程序员需要用清晰易懂的语言为用户编写操作使用手册
- C. 为提高程序的可读性，程序中的注释应重点解释程序中各语句的语义
- D. 编程需要个性化艺术，也要讲究团队协作，闭门造车往往事倍功半

参考答案：C

第56题：用户界面设计的原则不包括（ ）。

- A. 适合用户的业务领域和操作习惯

- B. 保持界面元素、布局与术语的一致性
- C. 提供反馈机制, 注重用户体验
- D. 按照五年后的发展潮流进行时尚设计

参考答案: D

第57题: 以下关于程序员工作的叙述中, 不正确的是()。

- A. 软件开发比软件测试有更高的技术含量
- B. 程序员需要通过实践了解自己的编程弱点
- C. 程序员应平衡测试时间、测试成本和质量之间的关系
- D. 最佳的编程方案必须同时兼顾程序质量和资源节约

参考答案: A

第58题: 某企业研发信息系统的过程中, () 不属于数据库管理员(DBA)的职责。

- A. 决定数据库中的信息内容和结构
- B. 决定数据库的存储结构和存取策略
- C. 进行信息系统程序的设计与编写
- D. 定义数据的安全性要求和完整性约束条件

参考答案: C

第59题: 某高校人事管理系统中, 规定讲师每课时的教学酬金不能超过100元, 副教授每课时的教学酬金不能超过130元, 教授每课时的教学酬金不能超过160元。这种情况下所设置的数据完整性约束条件称之为()。

- A. 实体完整性
- B. 用户定义完整性
- C. 主键约束完整性
- D. 参照完整性

参考答案: B

解析:

本题考查对数据库完整性约束方面的基础知识。

数据库完整性(Database Integrity)是指数据库中数据的正确性和相容性。数据库完整性由各种各样的完整性约束来保证, 因此可以说数据库完整性设计就是数据库完整性约束的设计。数据库完整性约束包括实体完整性、参照完整性和用户定义完整性。

实体完整性(Entity Integrity)指表中行的完整性。主要用于保证操作的数据(记录)非空、唯一且不重复。即实体完整性要求每个关系(表)有且仅有一个主键, 每一个主键值必须唯一, 而且不允许为“空”(NULL)或重复。

参照完整性(Referential Integrity)属于表间规则。在关系数据库中, 关系之间的联系是通过公共属性实现的。这个公共属性经常是一个表的主键, 同时是另一个表的外键。参照完整性体现在两个方面: 实现了表与表之间的联系, 外键的取值必须是另一个表的主键的有效值, 或是“空”值。参照完整性规则要求: 若属性组F是关系模式R1的主键, 同时F也是关系模式R2的外键, 则在R2的关系中, F的取值只允许两种可能: 空值或等于R1关系中某个主键值。

用户定义完整性(User-defined Integrity)也称域完整性规则, 是对数据表中字段属性的约束, 包括字段的值域、字段的类型和字段的有效规则(如小数位数)等约束, 是由确定关系结构时所定义的字段的属性决定的。例如, 百分制成绩的取值范围在0~100之间; 性别取值为“男”或“女”等。

第60题: 某教学管理数据库中, 学生、课程关系模式分别为: S (学号, 姓名, 性别, 家庭住址, 电话), 关系S的主键为学号; C (课程号, 课程名, 学分), 关系C的主键为课程号。假设一个学生可以选择多门课程, 一门课程可以由多个学生选择; 一旦学生选择某门课程必定有该课程的成绩。由于学生与课程之间的“选课”联系类型为(60), 所以对该联系(61)。

查询“软件工程”课程的平均成绩、最高成绩与最低成绩之间差值的SQL语句如下:

```
SELECT AVG (成绩) AS平均成绩, (62)
FROM C, SC
WHERE C.课程名='软件工程' AND C.课程号=SC.课程号;
```

A. n:m

B. 1:n

C. n:1

D. 1:1

参考答案: A

第61题: 某教学管理数据库中, 学生、课程关系模式分别为: S (学号, 姓名, 性别, 家庭住址, 电话), 关系S的主键为学号; C (课程号, 课程名, 学分), 关系C的主键为课程号。假设一个学生可以选择多门课程, 一门课程可以由多个学生选择; 一旦学生选择某门课程必定有该课程的成绩。由于学生与课程之间的“选课”联系类型为(60), 所以对该联系(61)。

查询“软件工程”课程的平均成绩、最高成绩与最低成绩之间差值的SQL语句如下:

```
SELECT AVG (成绩) AS平均成绩, (62)
FROM C, SC
WHERE C.课程名='软件工程' AND C.课程号=SC.课程号;
```

- A. 不需要构建一个独立的关系模式
- B. 需要构建一个独立的关系模式, 且关系模式为: SC (课程号, 成绩)
- C. 需要构建一个独立的关系模式, 且关系模式为: SC (学生号; 成绩)
- D. 需要构建一个独立的关系模式, 且关系模式为: SC (学生号, 课程号, 成绩)

参考答案: D

第62题: 某教学管理数据库中, 学生、课程关系模式分别为: S (学号, 姓名, 性别, 家庭住址, 电话), 关系S的主键为学号; C (课程号, 课程名, 学分), 关系C的主键为课程号。假设一个学生可以选择多门课程, 一门课程可以由多个学生选择; 一旦学生选择某门课程必定有该课程的成绩。由于学生与课程之间的“选课”联系类型为 (60), 所以对该联系 (61)。

查询“软件工程”课程的平均成绩、最高成绩与最低成绩之间差值的SQL语句如下:

SELECT AVG (成绩) AS平均成绩, (62)

FROM C, SC

WHERE C.课程名='软件工程' AND C.课程号=SC.课程号;

- A. 差值AS MAX (成绩) -MIN (成绩)
- B. MAX (成绩) -MIN (成绩) AS差值
- C. 差值IN MAX (成绩) -MIN (成绩)
- D. MAX (成绩) -MIN (成绩) IN差值

参考答案: B

第63题: 某宾馆有200间标准客房, 其入住率与客房单价有关。根据历史统计, 客房最高单价为160元时入住率为50%, 单价每降低1元, 入住率就会增加0.5%。据此选定价格为 () 时, 宾馆每天的收入最大。

- A. 120元
- B. 130元
- C. 140元
- D. 150元

参考答案: B

第64题: 菲波那契(Fibonacci)数列定义为

$f(1)=1, f(2)=1, n>2$ 时 $f(n)=f(n-1)+f(n-2)$

据此可以导出, $n>1$ 时, 有向量的递推关系式:

$(f(n+1), f(n)) = f(f(n), f(n-1))A$

其中A是 2×2 矩阵 (64)。从而, $f(n+1), f(n) = (f(2), f(1)) * (65)$ 。

A. $\begin{pmatrix} 0 & 1 \\ 1 & 1 \end{pmatrix}$

B. $\begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix}$

C. $\begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}$

D. $\begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix}$

参考答案：D

第65题：菲波那契(Fibonacci)数列定义为

$f(1)=1, f(2)=1, n>2$ 时 $f(n)=f(n-1)+f(n-2)$

据此可以导出, $n>1$ 时, 有向量的递推关系式：

$(f(n+1), f(n)) = f(f(n), f(n-1))A$

其中A是 2×2 矩阵 (64)。从而, $f(n+1), f(n) = (f(2), f(1)) * (65)$ 。

A. A^{n-1}

B. A^n

C. A^{n+1}

D. A^{n+2}

参考答案：A

第66题：Windows系统中定义了一些用户组，拥有完全访问权的用户组是 ()。

A. Power Users

B. Users

C. Administrators

D. Guests

参考答案 : C

第67题 : 浏览器本质上是一个 ()。

A. 连入Internet的TCP/IP程序

B. 连入Internet的SNMP程序

C. 浏览Web页面的服务器程序

D. 浏览Web页面的客户程序

参考答案 : D

第68题 : 在HTML文件中, 标签的作用是 ()。

A. 换行

B. 增大字体

C. 加粗

D. 锚

参考答案 : C

第69题 : 在HTML中, border属性用来指定表格 ()。

A. 边框宽度

B. 行高

C. 列宽

D. 样式

参考答案 : A

第70题 : 某PC出现网络故障, 一般应首先检查()。

A. DNS服务器

B. 路由配置

- C. 系统病毒
- D. 物理连通性

参考答案 : D

第71题 : Since tablet computers and smart phones have () interface, many people believe that all home and business computers will eventually have this kind of interface too.

- A. CRT
- B. LED
- C. Touch-screen
- D. Large screen

参考答案 : C

第72题 : () are specialized programs that assist you locating information on the web.

- A. OS
- B. Browse
- C. DBMS
- D. Search engines

参考答案 : D

第73题 : Program () describes program" s objectives, desired output, input data required, processing requirement, and documentation.

- A. specification
- B. flowchart
- C. structure
- D. address

参考答案 : C

第74题: A good program should be () by programmers other than the person who wrote it.

- A. reliable
- B. understandable
- C. structured
- D. blocked

参考答案: B

第75题: () refers to the process of testing and then eliminating errors.

- A. Debugging
- B. Programming
- C. Analysis
- D. Maintenance

参考答案: A

跨步软考 (kuabu.xyz)
手机APP, 微信公众号

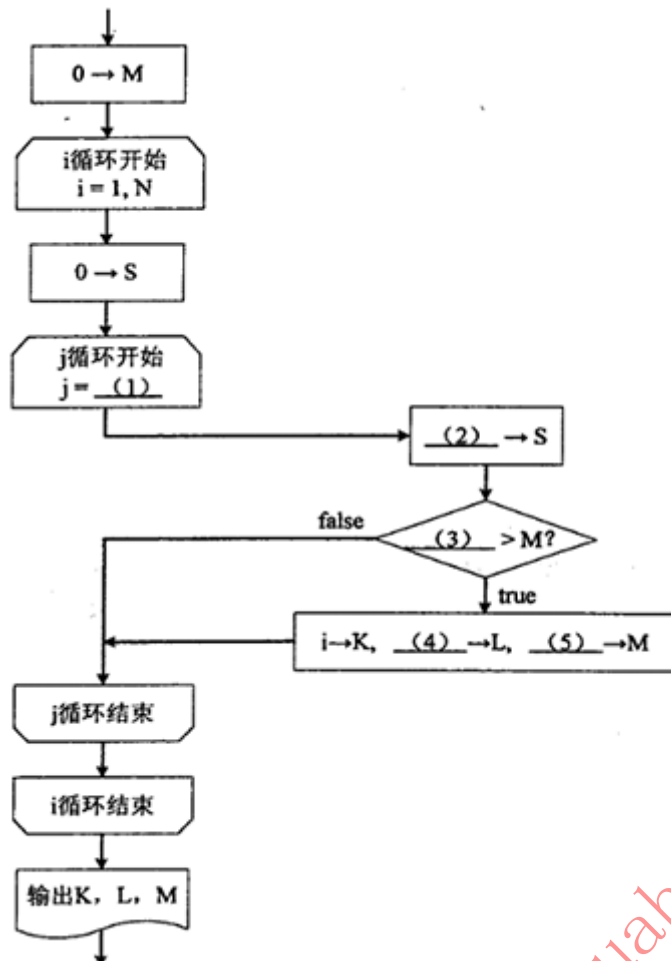
下午案例分析

第1题: 阅读以下说明和流程图, 填补流程图中的空缺, 将解答填入答题纸的对应栏内。

【说明】

设有整数数组A[1:N] ($N > 1$), 其元素有正有负。下面的流程图在该数组中寻找连续排列的若干个元素, 使其和达到最大值, 并输出其起始下标K、元素个数L以及最大的和值M。

例如, 若数组元素依次为3, -6, 2, 4, -2, 3, -1, 则输出K=3, L=4, M=7。该流程图中考察了A[1:N]中所有从下标i到下标j ($j \geq i$) 的各元素之和S, 并动态地记录其最大值M。问题: 1.1 【流程图】



注：循环开始框内应给出循环控制变量的初值和终值，默认递增值为1，格式为：循环控制变量=初值，终值

跨步软考[www.kuabu.xyz]答案解析：

- 1、j=i+1
- 2、A[j]
- 3、S
- 4、j
- 5、A[j]

第2题：阅读以下代码，回答问题1至问题3，将解答填入答题纸的对应栏内。

【代码1】

```

#include <stdio.h>
void swap(int x, int y)
{
    int tmp = x; x = y; y = tmp;
}
int main()
{

```

```
int a= 3, b= 7;
printf("a1= %d b1=%d ",a,b);
Swap( a, b);
Printf("a2 = %d b2=%d " ,a,b) ;
return 0;
}
```

【代码2】

```
#include<stdio.h>
#define SPACE " //空格字符
Int main()
{
char str[128] =" Nothing is impossible! ";
int i,num =0,wordMark=0;

for(i=0;str[i];i++)
If(str[i]=SPACE)
WordMark=0;
else
If(wordMark=0){
wordMark=1;
num++;
}

Printf( "%d/n" ,num)
return 0;
}
```

【代码3】

```
#include<stdio.h>
#define SPACE " //空格字符

int countStrs(char *);
int main()
{
char str[128] = " Nothing is impossible! ";
Printf( '%d/n, ( 1 ) (str))
return 0;
}

int countStrs(char *p)
{
int num=0, wordMark= 0;
for( ( 2 ) ; p++) {
If( ( 3 ) =SPACE)
wordMark= 0;
else
if( !wordMark ) {
```

```
wordMark = 1;  
++ num  
}  
}
```

return (4)

}问题：2.1 (4分)

写出代码1运行后的输出结果。问题：2.2 (3分)

写出代码2运行后的输出结果。问题：2.3 (8分)

代码3的功能与代码2完全相同, 请补充3中的空缺, 将解答写入答题纸的对应栏内。

跨步软考[www.kuabu.xyz]答案解析：

a1=3

b1=7

a2=7

b2=3

跨步软考[www.kuabu.xyz]答案解析：

3

跨步软考[www.kuabu.xyz]答案解析：

1) CountStr

2) p[i]

3) p[i]

4) num

第3题：阅读以下说明和代码, 填补代码中的空缺, 将解答填入答题纸的对应栏内。

【说明】

下面的程序利用快速排序中划分的思想在整数序列中找出第k小的元素 (即将元素从小到大排序后, 取第k个元素)。

对一个整数序列进行快速排序的方法是：在待排序的整数序列中取第一个数作为基准值, 然后根据基准值进行划分, 从而将待排序的序列划分为不大于基准值者 (称为左子序列) 和大于基准值者 (称为右子序列), 然后再对左子序列和右子序列分别进行快速排序, 最终得到非递减的有序序列。

例如, 整数序列 "19, 12, 30, 11, 7, 53, 78, 25" 的第3小元素为12。整数序列 "19, 12, 7, 30, 11, 11, 7, 53, 78, 25, 7" 的第3小元素为7。

函数partition (int a[], int low, int high) 以a[low]的值为基准, 对a[low]、a[low+1]、...、a[high]进行划分, 最后将该基准值放入a[i] ($low \leq i \leq high$), 并使得a[low]、a[low+1]、...、

$A[i-1]$ 都小于或等于 $a[i]$, 而 $a[i+1]$ 、 $a[i+2]$ 、...、 $a[high]$ 都大于 $a[i]$ 。

函数findkthElem(int a[],int startIdx,int endIdx,int k)在 $a[startIdx]$ 、 $a[startIdx+1]$ 、...、 $a[endIdx]$ 中找出第k小的元素。问题：3.1 【代码】

```
#include <stdio.h>
```

```
#include <stdlib.h>
```

```
Int partition ( int a [], int low, int high )
```

```
{//对 a[low..high]进行划分, 使得a[low..i]中的元素都不大于a[i+1..high]中的元素。
```

```
int pivot=a[low]; //pivot表示基准元素
```

```
Int i=low,j=high;
```

```
while( ( 1 ) ){
```

```
While(i<j&& a[j]>pivot)--j;
```

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

```
While(i<j&& a[i]>pivot)++i;
```

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

```
}
```

```
( 2 ) ; //基准元素定位
```

```
return i ;
```

```
}
```

```
Int findkthElem ( int a[] , int startIdx , int endIdx, int k )
```

```
{//整数序列存储在a[startIdx..endIdx]中, 查找并返回第k小的元素。
```

```
if (startIdx<0 || endIdx<0 || startIdx>endIdx || k<1 || k-1>endIdx || k-1<startIdx)
```

```
Return -1; //参数错误
```

```
if(startIdx<endIdx){
```

```
int loc=partition(a, startIdx, endIdx); //进行划分, 确定基准元素的位置
```

```
if (loc==k-1) //找到第k小的元素
```

```
return ( 3 ) ;
```

```
if(k-1 < loc) //继续在基准元素之前查找
```

```
return findkthElem(a , ( 4 ) , k ) ;
```

```
else // 继续在基准元素之后查找
```

```
return findkthElem(a , ( 5 ) , k);
```

```
}
```

```
return a[startIdx] ;
```

```
}
```

```
int main()
```

```
{
```

```
int i, k;
```

```
int n;
```

```
int a[] = {19, 12, 7, 30, 11, 11, 7, 53, 78, 25, 7};
```

```
n= sizeof(a) / sizeof(int) //计算序列中的元素个数
```

```
for (k=1;k < n+1 ; k++){
```

```
for(i=0;i < n;i++){
```

```
printf( "%d\t" ,a[i]);
```

```
}  
printf( " " );  
printf( "elem %d=%d ,k,findkthElem(a,0,n-1,k));//输出序列中第k小的元素  
}  
return 0 ;  
}
```

跨步软考[www.kuabu.xyz]答案解析：

- 1、!i=j
- 2、a[i]=pivot
- 3、a[loc]
- 4、stratIdx,Loc-1
- 5、Loc+1,endIdx

第4题：阅读以下说明和代码，填补代码中的空缺，将解答填入答题纸的对应栏内。

【说明】

图是很多领域中的数据模型，遍历是图的一种基本运算。从图中某顶点v出发进行广度优先遍历的过程是：

- ①访问顶点v；
- ②访问V的所有未被访问的邻接顶点W1,W2,...,Wk；
- ③依次从这些邻接顶点W1,W2,...,Wk出发，访问其所有未被访问的邻接顶点；依此类推，直到图中所有访问过的顶点的邻接顶点都得到访问。

显然，上述过程可以访问到从顶点V出发且有路径可达的所有顶点。对于从v出发不可达的顶点u，可从顶点u出发再次重复以上过程,直到图中所有顶点都被访问到。

例如，对于图4-1所示的有向图G，从a出发进行广度优先遍历，访问顶点的一种顺序为a、b、c、e、f、d。

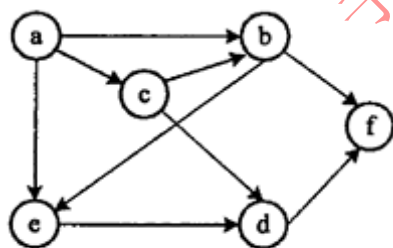


图 4-1

	a	b	c	d	e	f
a	0	1	1	0	1	0
b	0	0	0	0	1	1
c	0	1	0	1	0	0
d	0	0	0	0	0	1
e	0	0	0	1	0	0
f	0	0	0	0	0	0

图 4-2

设图G采用数组表示法 (即用邻接矩阵arcs存储), 元素arcs[i][j]定义如下:

$$\text{arcs}[i][j] = \begin{cases} 1 & \text{若G中存在边}(v_i, v_j)\text{或弧} \langle v_i, v_j \rangle \\ 0 & \text{若G中不存在边}(v_i, v_j)\text{或弧} \langle v_i, v_j \rangle \end{cases}$$

图4-1的邻接矩阵如图4-2所示, 顶点a~f对应的编号依次为0~5.因此, 访问顶点a的邻接顶点的顺序为b, c, e.

函数BFSTraverse(Graph G)利用队列实现图G的广度优先遍历.

相关的符号和类型定义如下:

```
#define MaxN : 50 /*图中最多顶点数*/  
typedef int AdjMatrix[MaxN][MaxN];  
typedef struct{  
int vexnum, edgenum; /*图中实际顶点数和边(弧)数*/  
AdjMatrix arcs; /*邻接矩阵*/  
}Graph;  
typedef int QElemType;  
enum {ERROR=0;OK=1};
```

代码中用到的队列运算的函数原型如表4-1所述, 队列类型名为QUEUE.

表4-1 实现队列运算的函数原型及说明

函数原型	说明
InitQueue(QUEUE *Q)	初始化一个空队列
isEmpty(QUEUE Q)	判断队列是否为空, 是则为1, 否则为0
EnQueue(QUEUE *Q, QElemType qe)	将元素 qe 加入队列
DeQueue(QUEUE *Q, QElemType *te)	从队列头部删除元素, 并通过参数 te 带回其值

问题: 4.1

【代码】

```
int BFSTraverse(Graph G)  
{//图G进行广度优先遍历, 图采用邻接矩阵存储  
unsigned char*visited; //visited[]用于存储图G中各顶点的访问标志, 0表示未访问  
int v, w; u;  
QUEUE Q;  
//申请存储顶点访问标志的空间, 成功时将所申请空间初始化为0  
visited=(char*)calloc(G.vexnum, sizeof(char));  
If ( 1 ) )  
return ERROR;  
  
( 2 ) ; //初始化Q为空队列  
for( v=0; v<G.vexnum; v++){  
if(!visited[v]){ //从顶点v出发进行广度优先遍历  
printf("%d", v); //访问顶点v并将其加入队列  
visited[v]=1;  
( 3 ) ;  
while( ! isEmpty(Q)){  
( 4 ) ; //出队列并用u表示出队的元素  
for(v=0;v<G.vexnum; w++){  
if ( G.arcs[u][w]!=0&& ( 5 ) ){ //w是u的邻接顶点且未访问过  
printf("%d", w); //访问顶点w  
visited[w]=1;  

```



```
EnQueue(&Q, w);  
}  
}  
}  
}  
free(visited);  
return OK;  
)//BFSTraverse
```

跨步软考[www.kuabu.xyz]答案解析：

- 1、visited==NULL
- 2、InitQueue(&Q)
- 3、EnQueue(&Q,v)
- 4、DeQueue(&Q,&u)
- 5、visited==0

第5题：阅读以下说明和Java程序，填补代码中的空缺，将解答填入答题纸的对应栏内。

【说明】：

以下Java代码实现一个简单的聊天室系统（ChatRoomSystem），多个用户(User)可以向聊天室(ChatRoom)发送消息，聊天室将消息展示给所有用户。类图如图5-1所示。

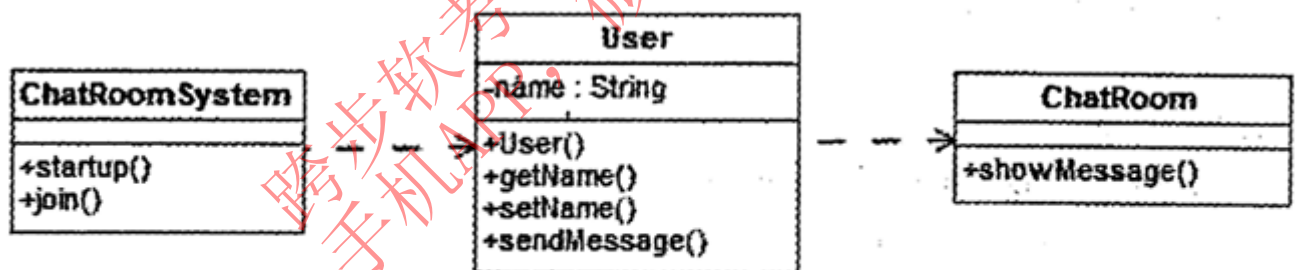


图 5-1 类图

问题：5.1 【Java代码】

```
class ChatRoom {  
public static void showMessage(User user, string message) {  
System.out.println "[" + user.getName() + "] : " + message);  
}  
}
```

```
class User {  
private String name;  
  
public String getName() {  
return name;  
}  
}
```



```
public void setName(String name) {
    this.name = name;
}

public User(String name) {
    ( 1 ) =name ;
}

public void sendMessage(String message) {
    ( 2 ) (this, message);
}

public class ChatRoomSystem {
    public void startup() {
        User zhang= new User("John");
        User li =new User("Leo");

        zhang.sendMessage("Hi! Leo! ");
        li.sendMessage("Hi! John!");
    }
    public void join(User user) {
        ( 3 ) ("Hello Everyone! I am" + user.getName());
    }
    public static void main(String[] args) {
        ChatRoomSystem crs= ( 4 ) ;
        Crs.startup();
        Crs.join( ( 5 ) )( "Wayne" ));
    }
}

/*
```

程序运行结果：

[John] : Hi! Leo!

[Leo] : Hi! John!

[Wayne] : Hello Everyone! I am Wayne

*/

跨步软考[www.kuabu.xyz]答案解析：

- 1、this.name
- 2、ChatRoom.sendMessage
- 3、user.sendMessage
- 4、new ChatRoomSystem()
- 5、new User

第6题：阅读下列说明和C++代码，填补代码中的空缺，将解答填入答题纸的对应栏内。

【说明】

以下C++代码实现一个简单的聊天室系统（ChatRoomSystem），多个用户(User)可以向聊天室（ChatRoom）发送消息，聊天室将消息展示给所有用户。类图如图6-1所表示。

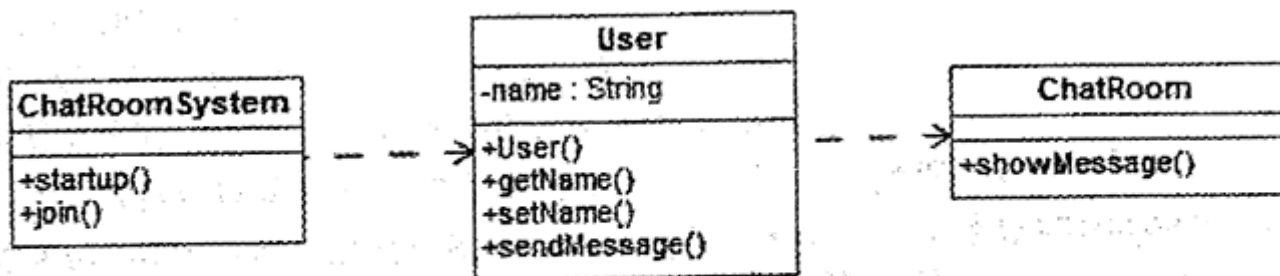


图 6-1 类图

问题：6.1 【C++代码】

```
#include<iostream>
```

```
#include <string>
```

```
using namespace std ;
```

```
class User {
```

```
private:
```

```
string name;
```

```
public:
```

```
User(string name){
```

```
    ( 1 ) =name ;
```

```
}
```

```
~User(){
```

```
void setName(string name) {
```

```
    this->name=name;
```

```
}
```

```
string getName(){
```

```
    return name;
```

```
}
```

```
void sendMessage(string message);
```

```
};
```

```
class ChatRoom { .
```

```
public:
```

```
static void showMessage(User* user, string message) {
```

```
    cout<<"["<<user->getName()"] : "<<message<<endl;
```

```
}
```

};

```
void User::sendMessage(string message) {  
    ( 2 ) (this,message);  
}
```

```
class ChatRoomSystem{  
public: . .  
void startup0(){  
    User* zhang = new User( "John");  
    User* li = new User("Leo");  
  
    zhang->sendMessage("Hi! Leo!");  
    li->sendMessage("Hi! John!");  
}
```

```
void join(User* user) {  
    ( 3 ) ("HelloEveryone!I am"+user->getName()); . ;  
}.  
};
```

```
int main(){  
    ChatRoomSystem*crs= ( 4 ) ;  
    crs->startup();  
    crs->join( ( 5 ) ("Wayne"));  
    delete crs;  
}  
/*
```

程序运行结果：

[John] : Hi! Leo!

[Leo] : Hi! John!

[Wayne] : Hello Everyone!iam Wayne

/*

跨步软考[www.kuabu.xyz]答案解析：

- 1、this->name
- 2、ChatRoom::showMessage
- 3、user->sendMessage
- 4、new ChatRoomSystem()
- 5、new User