全国计算机技术与软件专业技术资格（水平）考试

中级 软件设计师 2016 年 下半年 上午试卷 综合知识

（考试时间 150 分钟）

1. 在答题卡的指定位置上正确写入你的姓名和准考证号，并用正规 2B 铅笔在你写入的准考证号下填涂准考证号。

2. 本试卷的试题中共有 75 个空格，需要全部解答，每个空格 1 分，满分 75 分。

3. 每个空格对应一个序号，有 A 、 B 、 C 、 D 四个选项，请选择一个最恰当的选项作为解答，在答题卡相应序号下填涂该选项。

4. 解答前务必阅读例题和答题卡上的例题填涂样式及填涂注意事项。解答时用正规 2B 铅笔正确填涂选项，如需修改，请用橡皮擦干净，否则会导致不能正确评分。

试题一 在程序运行过程中， CPU 需要将指令从内存中取出并加以分析和执行。 CPU 依据( )来区分在内存中以二进制编码形式存放的指令和数据。

A. 指令周期的不同阶段　　B. 指令和数据的寻址方式　　C. 指令操作码的译码结果
D. 指令和数据所在的存储单元

试题二 计算机在一个指令周期的过程中，为从内存读取指令操作码，首先要将( )的内容送到地址总线上。

A. 指令寄存器(IR)　　B. 通用寄存器(GR)　　C. 程序计数器(PC)　　D. 状态寄存器(PSW)

试题三 设 16 位浮点数，其中阶符 1 位、阶码值 6 位、数符 1 位、尾数 8 位。若阶码用移码表示，尾数用补码表示，则该浮点数所能表示的数值范围是( )。

A．-2^{64}～$(1-2^{-8})2^{64}$
B．-2^{63}～$(1-2^{-8})2^{63}$

C．$-2^{64}\sim(1-2-(1-2^{-8})2^{64}\sim(1-2^{-8})2^{64}$
D．$-(1-2^{-8})2^{63}\sim(1-2^{-8})2^{63}$

试题四　已知数据信息为 16 位，最少应附加(　)位校验位，以实现海明码纠错。
A．3　　B．4　　C．5　　D．6

试题五　将一条指令的执行过程分解为取址、分析和执行三步，按照流水方式执行，若取址时间 $t_{取址}=4\triangle t$ 、分析时间 $t_{分析}=2\triangle t$ 、执行时间 $t_{执行}=3\triangle t$ ，则执行完 100 条指令，需要的时间为(　)△t 。
A．200　　B．300　　C．400　　D．405

试题六　以下关于 Cache 与主存间地址映射的叙述中，正确的是(　)。

A. 操作系统负责管理 Cache 与主存之间的地址映射
B. 程序员需要通过编程来处理 Cache 与主存之间的地址映射
C. 应用软件对 Cache 与主存之间的地址映射进行调度
D. 由硬件自动完成 Cache 与主存之间的地址映射

试题七　可用于数字签名的算法是(　)。
A．RSA　　B．IDEA　　C．RC4　　D．MD5

试题八　(　)不是数字签名的作用。
A. 接收者可验证消息来源的真实性　　B. 发送者无法否认发送过该消息
C. 接收者无法伪造或篡改消息　　D. 可验证接收者合法性

试题九　在网络设计和实施过程中要采取多种安全措施，其中(　)是针对系统安全需求的措施。
A. 设备防雷击　　B. 入侵检测　　C. 漏洞发现与补丁管理　　D. 流量控制

试题一十　(　)的保护期限是可以延长的。
A. 专利权　　B. 商标权　　C. 著作权　　D. 商业秘密权

试题一十一 甲公司软件设计师完成了一项涉及计算机程序的发明。之后，乙公司软件设计师也完成了与甲公司软件设计师相同的涉及计算机程序的发明。甲、乙公司于同一天向专利局申请发明专利。此情形下，( )是专利权申请人。

A. 甲公司　　B. 甲、乙两公司　　C. 乙公司　　D. 由甲、乙公司协商确定的公司

试题一十二 甲、乙两厂生产的产品类似，且产品都使用“B”商标。两厂于同一天向商标局申请商标注册，且申请注册前两厂均未使用“B”商标。此情形下，( )能核准注册。

A. 甲厂　　B. 由甲、乙厂抽签确定的厂　　C. 乙厂　　D. 甲、乙两厂

试题一十三 (第 1 空)在 FM 方式的数字音乐合成器中，改变数字载波频率可以改变乐音的( )，改变它的信号幅度可以改变乐音的( )。

A. 音调　　B. 音色　　C. 音高　　D. 音质

试题一十四 (第 2 空)在 FM 方式的数字音乐合成器中，改变数字载波频率可以改变乐音的( )，改变它的信号幅度可以改变乐音的( )。

A. 音调　　B. 音域　　C. 音高　　D. 带宽

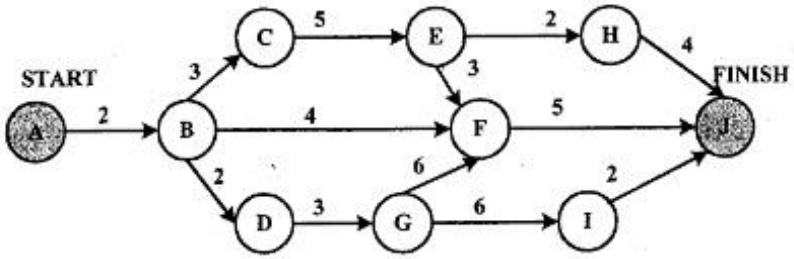
试题一十五 结构化开发方法中，( )主要包含对数据结构和算法的设计。

A. 体系结构设计　　B. 数据设计　　C. 接口设计　　D. 过程设计

试题一十六 在敏捷过程的开发方法中，( )使用了迭代的方法，其中，把每段时间(30 天)一次的迭代称为一个“冲刺”，并按需求的优先级别来实现产品，多个自组织和自治的小组并行地递增实现产品。

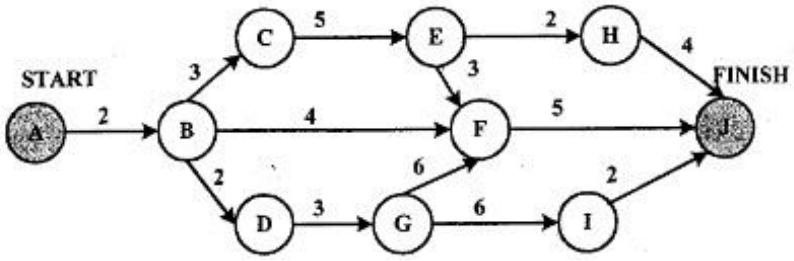
A. 极限编程 XP　　B. 水晶法　　C. 并列争球法　　D. 自适应软件开发

试题一十七 (第 1 空)某软件项目的活动图如下图所示，其中顶点表示项目里程碑，连接顶点的边表示包含的活动，边上的数字表示相应活动的持续时间(天)，则完成该项目的最少时间为( )天。活动 BC 和 BF 最多可以晚开始( )天而不会影响整个项目的进度。



A．11　　B．15　　C．16　　D．18

试题一十八　(第 2 空)某软件项目的活动图如下图所示，其中顶点表示项目里程碑，连接顶点的边表示包含的活动，边上的数字表示相应活动的持续时间(天)，则完成该项目的最少时间为(　)天。活动 BC 和 BF 最多可以晚开始(　)天而不会影响整个项目的进度。



A．0 和 7　　B．0 和 11　　C．2 和 7　　D．2 和 11

试题一十九　成本估算时，(　)方法以规模作为成本的主要因素，考虑多个成本驱动因子。该方法包括三个阶段性模型，即应用组装模型、早期设计阶段模型和体系结构阶段模型。

A．专家估算　　B．Wolverton　　C．COCOMO　　D．COCOMO Ⅱ

试题二十　逻辑表达式求值时常采用短路计算方式。“&&”、“||”、“！”分别表示逻辑与、或、非运算，“&&”、“||”为左结合，“！”为右结合，优先级从高到低为“！”、“&&”、“||”。对逻辑表达式“x&&(y||!z)”进行短路计算方式求值时，(　)。

A．x 为真，则整个表达式的值即为真，不需要计算 y 和 z 的值
B．x 为假，则整个表达式的值即为假，不需要计算 y 和 z 的值
C．x 为真，再根据 z 的值决定是否需要计算 y 的值
D．x 为假，再根据 y 的值决定是否需要计算 z 的值

试题二十一 常用的函数参数传递方式有传值与传引用两种。( )。

A. 在传值方式下，形参与实参之间互相传值　　B. 在传值方式下，实参不能是变量

C. 在传引用方式下，修改形参实质上改变了实参的值。　　D. 在传引用方式下，实参可以是任意的变量和表达式。

试题二十二 二维数组 a[1..N， 1..N]可以按行存储或按列存储。对于数组元素 a[i,j](1<=i,j<=N)，当( )时，在按行和按列两种存储方式下，其偏移量相同。

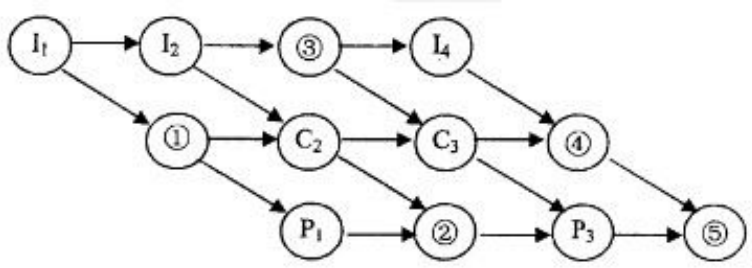
A. i≠j　　B. i=j　　C. i>j　　D. i<j

试题二十三 实时操作系统主要用于有实时要求的过程控制等领域。实时系统对于来自外部的事件必须在( )。

A. 一个时间片内进行处理　　B. 一个周转时间内进行处理

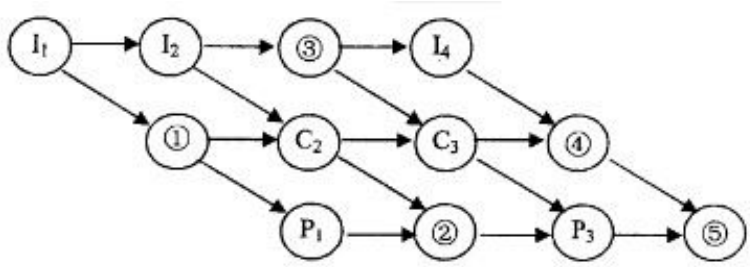
C. 一个机器周期内进行处理　　D. 被控对象规定的时间内做出及时响应并对其进行处理

试题二十四 (第 1 空)假设某计算机系统中只有一个 CPU 、一台输入设备和一台输出设备，若系统中有四个作业 T1 、 T2 、 T3 和 T4 ，系统采用优先级调度，且 T1 的优先级>T2 的优先级>T3 的优先级>T4 的优先级。每个作业 Ti 具有三个程序段：输入 I_i 、计算 C_i 和输出 P_i(i=1， 2， 3， 4)，其执行顺序为 $I_i \rightarrow C_i \rightarrow P_i$ 。这四个作业各程序段并发执行的前驱图如下所示。图中①、②分别为( )，③、④、⑤分别为( )。



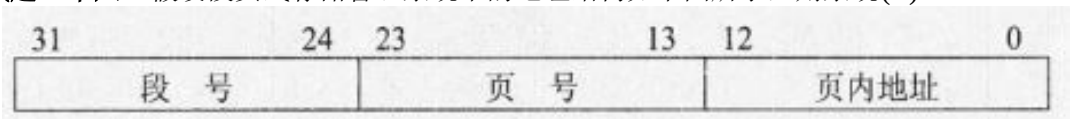
A. I_2、 P_2　　B. I_2、 C_2　　C. C_1、 P_2　　D. C_1、 P_3

试题二十五 (第 2 空)假设某计算机系统中只有一个 CPU 、一台输入设备和一台输出设备，若系统中有四个作业 T1 、 T2 、 T3 和 T4 ，系统采用优先级调度，且 T1 的优先级>T2 的优先级>T3 的优先级>T4 的优先级。每个作业 Ti 具有三个程序段：输入 I_i 、计算 C_i 和输出 P_i(i=1， 2， 3， 4)，其执行顺序为 $I_i \rightarrow C_i \rightarrow P_i$ 。这四个作业各程序段并发执行的前驱图如下所示。图中①、②分别为( )，③、④、⑤分别为( )。



A．C_2、C_4、P_4　　B．I_2、I_3、C_4　　C．I_3、P_3、P_4　　D．I_3、C_4、P_4

试题二十六　假设段页式存储管理系统中的地址结构如下图所示，则系统(　)。



A. 最多可有 256 个段，每个段的大小均为 2048 个页，页的大小为 8K
B. 最多可有 256 个段，每个段最大允许有 2048 个页，页的大小为 8K
C. 最多可有 512 个段，每个段的大小均为 1024 个页，页的大小为 4K
D. 最多可有 512 个段，每个段最大允许有 1024 个页，页的大小为 4K

试题二十七　假设系统中有 n 个进程共享 3 台扫描仪，并采用 PV 操作实现进程同步与互斥。若系统信号量 S 的当前值为-1，进程 P1、P2 又分别执行了 1 次 P(S)操作，那么信号量 S 的值应为(　)。
A．3　　B．-3　　C．1　　D．-1

试题二十八　某字长为 32 位的计算机的文件管理系统采用位示图(bitmap)记录磁盘的使用情况。若磁盘的容量为 300GB，物理块的大小为 1MB，那么位示图的大小为(　)个字。
A．1200　　B．3200　　C．6400　　D．9600

试题二十九　(第 1 空)某开发小组欲为一公司开发一个产品控制软件，监控产品的生产和销售过程，从购买各种材料开始，到产品的加工和销售进行全程跟踪。购买材料的流程、产品的加工过程以及销售过程可能会发生变化。该软件的开发最不适宜采用(　)模型，主要是因为这种模型(　)。
A. 瀑布　　B. 原型　　C. 增量　　D. 喷泉

试题三十　(第 2 空)某开发小组欲为一公司开发一个产品控制软件，监控产品的生产和销售过程，从购买各种材料开始，到产品的加工和销售进行全程跟踪。购买材料的流程、产

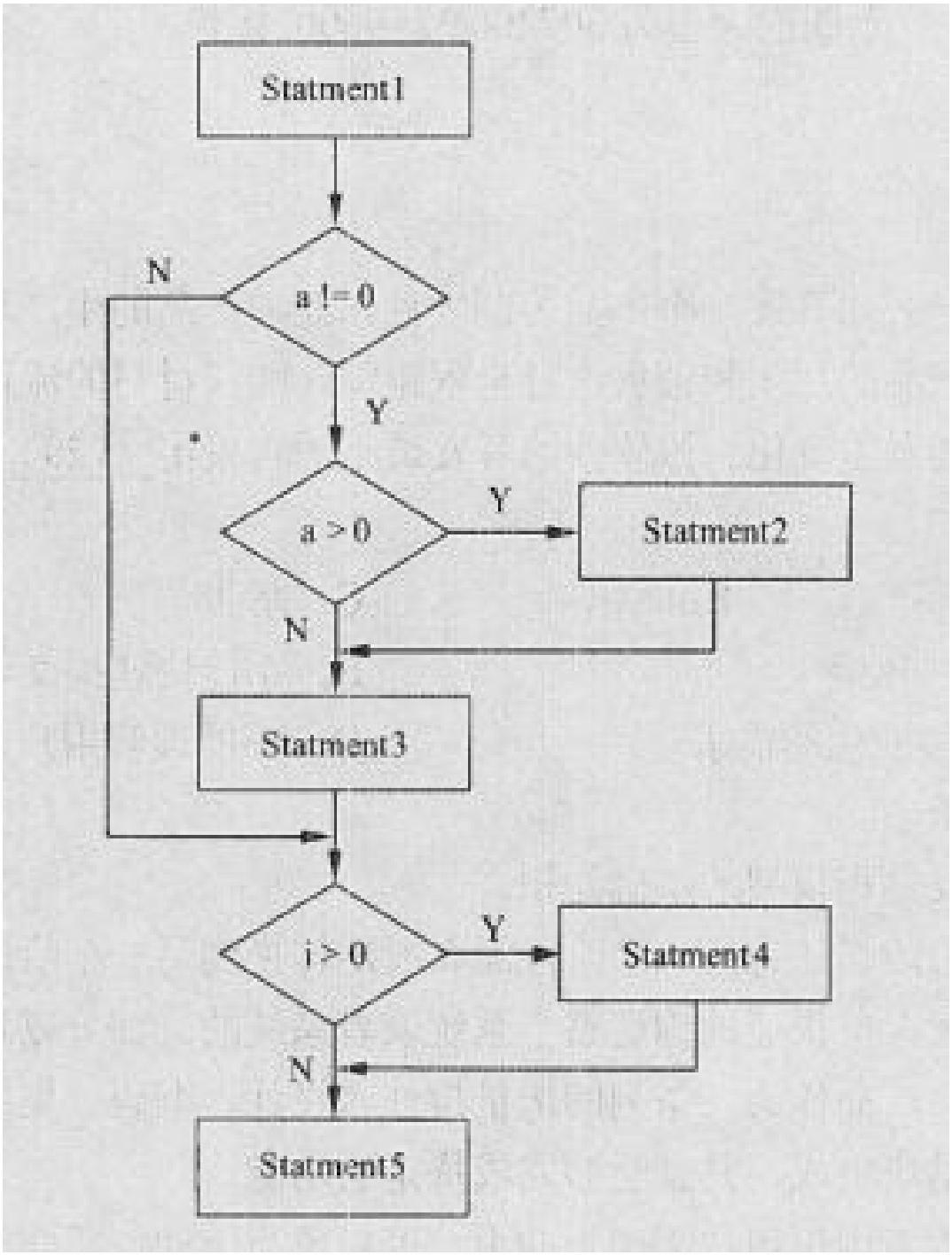
品的加工过程以及销售过程可能会发生变化。该软件的开发最不适宜采用( )模型，主要是因为这种模型( )。

A. 不能解决风险　　B. 不能快速提交软件　　C. 难以适应变化的需求　　D. 不能理解用户的需求

试题三十一　( )不属于软件质量特性中的可移植性。

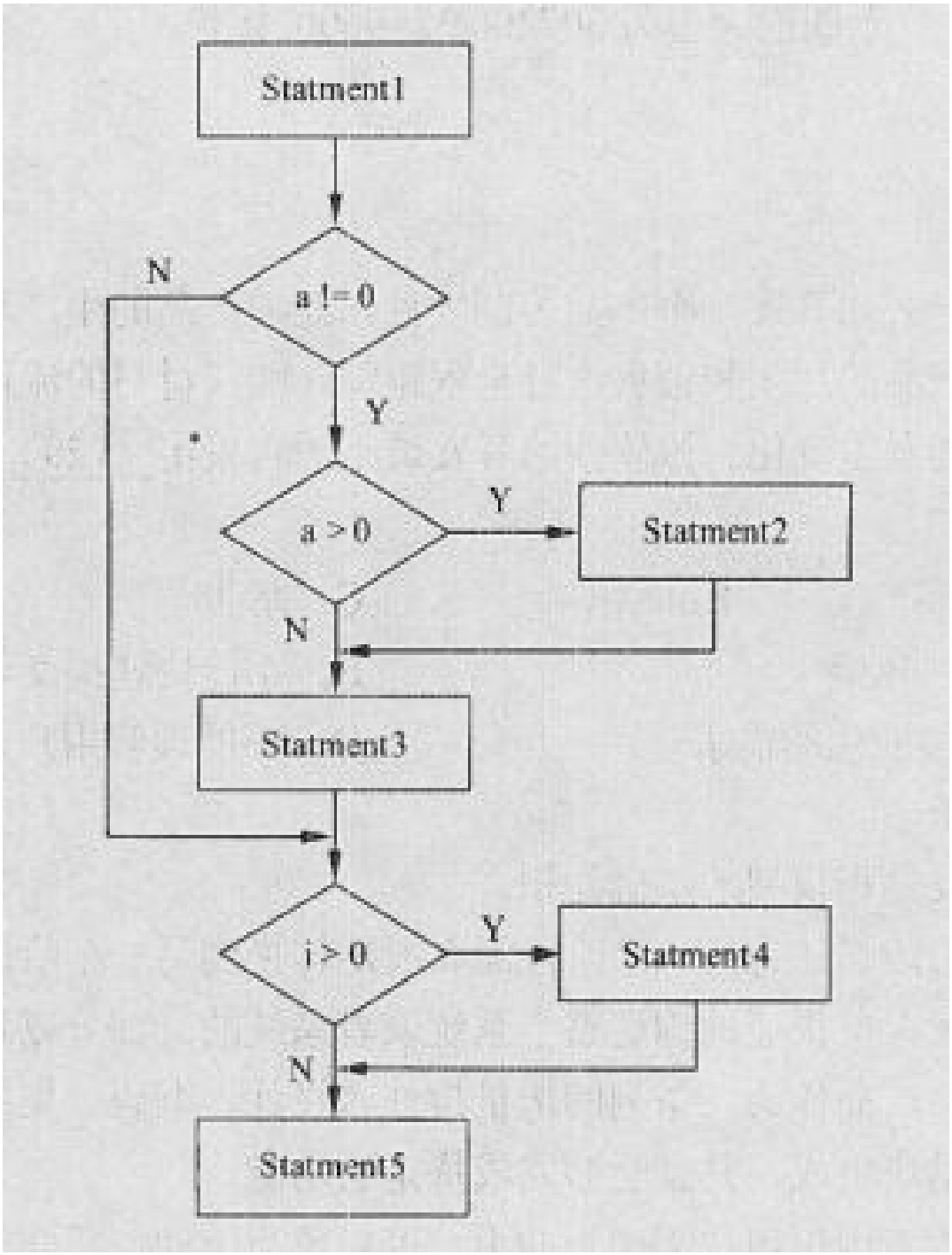
A. 适应性　　B. 易安装性　　C. 易替换性　　D. 易理解性

试题三十二　(第 1 空)对下图所示流程图采用白盒测试方法进行测试，若要满足路径覆盖，则至少需要( )个测试用例。采用 McCabe 度量法计算该程序的环路复杂性为( )。



A．3　　B．4　　C．6　　D．8

试题三十三　(第 2 空)对下图所示流程图采用白盒测试方法进行测试，若要满足路径覆盖，则至少需要(　)个测试用例。采用 McCabe 度量法计算该程序的环路复杂性为(　)。



A．1　　B．2　　C．3　　D．4

试题三十四　计算机系统的(　)可以用 MTBF/(1+MTBF)来度量，其中 MTBF 为平均失效间隔时间。

A. 可靠性　　B. 可用性　　C. 可维护性　　D. 健壮性

试题三十五　以下关于软件测试的叙述中，不正确的是(　)。

A. 在设计测试用例时应考虑输入数据和预期输出结果　　B. 软件测试的目的是证明软件的正确性
C. 在设计测试用例时，应该包括合理的输入条件　　D. 在设计测试用例时，应该包括不合理的输入条件

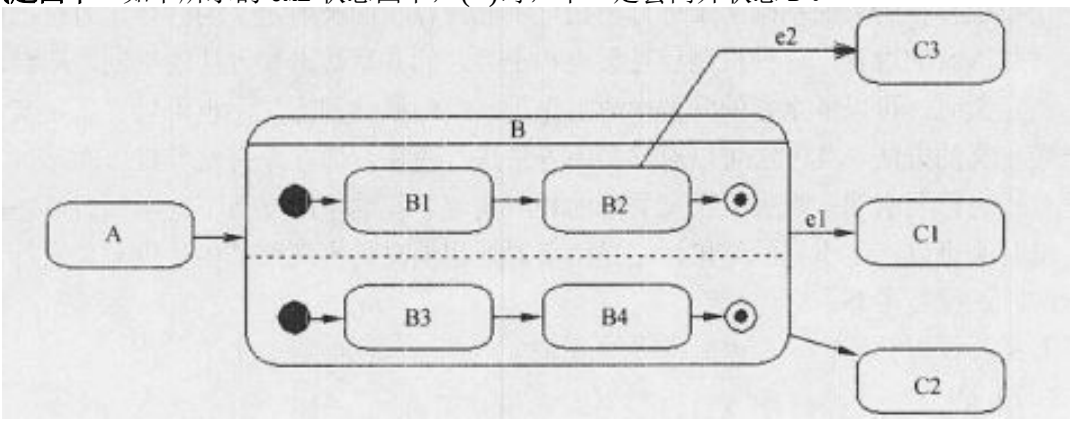
试题三十六　某模块中有两个处理 A 和 B，分别对数据结构 X 写数据和读数据，则该模块的内聚类型为(　)内聚。
A. 逻辑　　B. 过程　　C. 通信　　D. 内容

试题三十七　在面向对象方法中，不同对象收到同一消息可以产生完全不同的结果，这一现象称为(　)。在使用时，用户可以发送一个通用的消息，而实现的细节则由接收对象自行决定。
A. 接口　　B. 继承　　C. 覆盖　　D. 多态

试题三十八　在面向对象方法中，支持多态的是(　)。
A. 静态分配　　B. 动态分配　　C. 静态类型　　D. 动态绑定

试题三十九　面向对象分析的目的是为了获得对应用问题的理解，其主要活动不包括(　)。
A. 认定并组织对象　　B. 描述对象间的相互作用　　C. 面向对象程序设计　　D. 确定基于对象的操作

试题四十　如下所示的 UML 状态图中，(　)时，不一定会离开状态 B 。

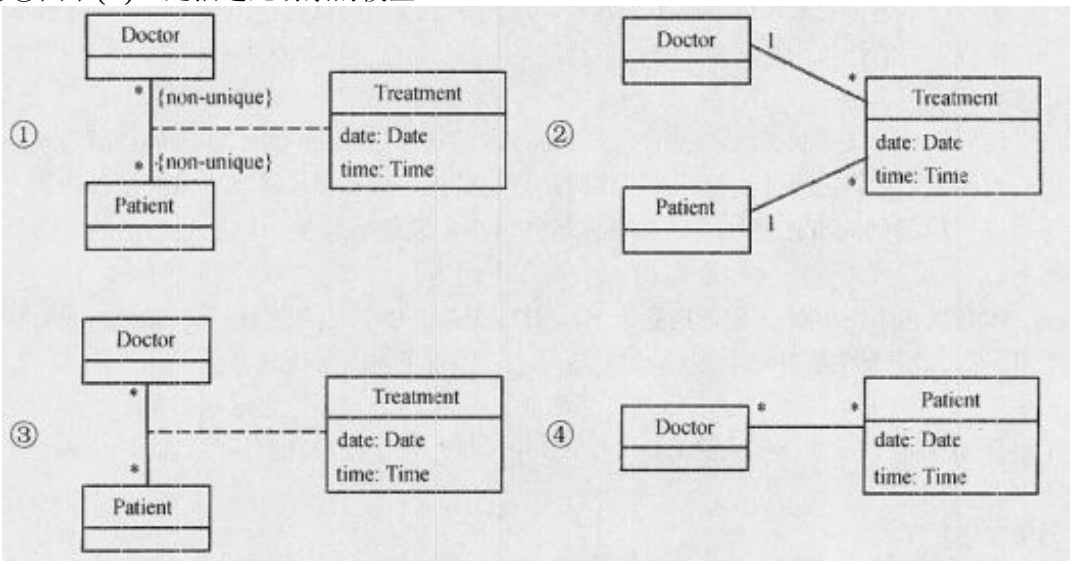


A. 状态 B 中的两个结束状态均达到　　B. 在当前状态为 B2 时，事件 e2 发生
C. 事件 e2 发生　　D. 事件 e1 发生

试题四十一　以下关于 UML 状态图中转换(transition)的叙述中，不正确的是(　)。

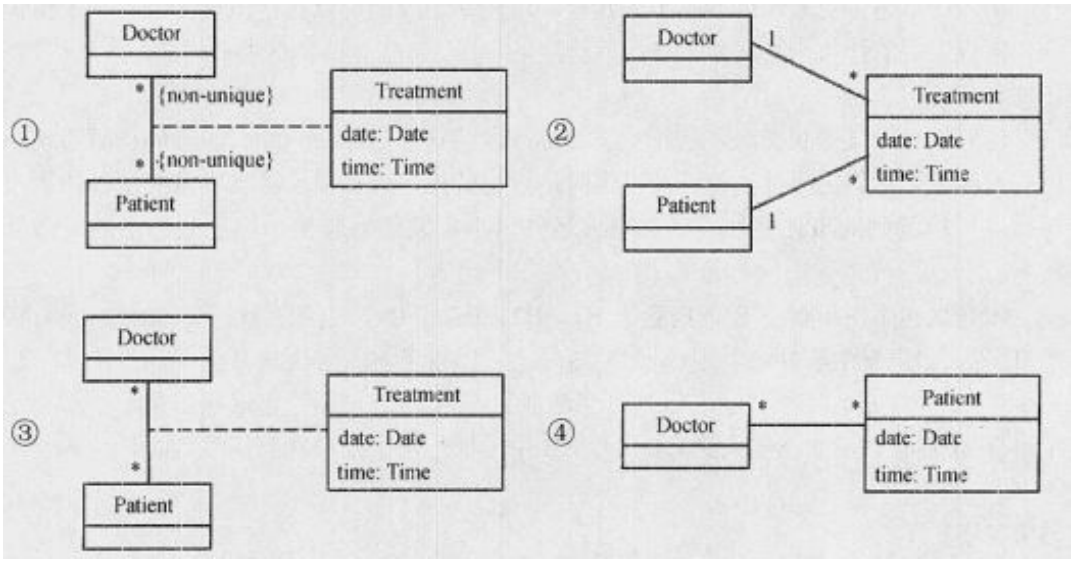
A. 活动可以在转换时执行也可以在状态内执行　　B. 监护条件只有在相应的事件发生时才进行检查
C. 一个转换可以有事件触发器、监护条件和一个状态　　D. 事件触发转换

试题四十二　(第 1 空)下图①②③④所示是 UML(　)。现有场景：一名医生(Doctor)可以治疗多位病人(Patient)，一位病人可以由多名医生治疗，一名医生可能多次治疗同一位病人。要记录哪名医生治疗哪位病人时，需要存储治疗(Treatment)的日期和时间。以下①②③④图中(　)。是描述此场景的模型。



A. 用例图　　B. 对象图　　C. 类图　　D. 协作图

试题四十三　(第 2 空)下图①②③④所示是 UML(　)。现有场景：一名医生(Doctor)可以治疗多位病人(Patient)，一位病人可以由多名医生治疗，一名医生可能多次治疗同一位病人。要记录哪名医生治疗哪位病人时，需要存储治疗(Treatment)的日期和时间。以下①②③④图中(　)。是描述此场景的模型。



A. ①　　B. ②　　C. ③　　D. ④

试题四十四　(第 1 空)(　)模式定义一系列的算法，把它们一个个封装起来，并且使它们可以相互替换，使得算法可以独立于使用它们的客户而变化。以下(　)情况适合选用该模式。

①一个客户需要使用一组相关对象

②一个对象的改变需要改变其它对象

③需要使用一个算法的不同变体

④许多相关的类仅仅是行为有异

A. 命令(Command)
B. 责任链(ChainofResponsibility)
C. 观察者(Observer)
D. 策略(Strategy)

试题四十五　(第 2 空)(　)模式定义一系列的算法，把它们一个个封装起来，并且使它们可以相互替换，使得算法可以独立于使用它们的客户而变化。以下(　)情况适合选用该模式。

①一个客户需要使用一组相关对象

②一个对象的改变需要改变其它对象

③需要使用一个算法的不同变体

④许多相关的类仅仅是行为有异

A. ①②　　B. ②③　　C. ③④　　D. ①④

试题四十六　(第 1 空)(　)模式将一个复杂对象的构建与其表示分离，使得同样的构建过程可以创建不同的表示。以下(　)情况适合选用该模式。

①抽象复杂对象的构建步骤

②基于构建过程的具体实现构建复杂对象的不同表示

③一个类仅有一个实例

④一个类的实例只能有几个不同状态组合中的一种

A. 生成器(Builder)　　B. 工厂方法(FactoryMethod)

C. 原型(Prototype)　　D. 单例(Singleton)

试题四十七　(第 2 空)(　)模式将一个复杂对象的构建与其表示分离，使得同样的构建过程可以创建不同的表示。以下(　)情况适合选用该模式。

①抽象复杂对象的构建步骤

②基于构建过程的具体实现构建复杂对象的不同表示

③一个类仅有一个实例

④一个类的实例只能有几个不同状态组合中的一种

A. ①②　　B. ②③　　C. ③④　　D. ①④

试题四十八　由字符 a 、 b 构成的字符串中，若每个 a 后至少跟一个 b ，则该字符串集合可用正规式表示为(　)。

A. (b|ab)*　　B. (ab*)*　　C. (a*b*)*　　D. (a|b)*

试题四十九　乔姆斯基(Chomsky)将文法分为 4 种类型，程序设计语言的大多数语法现象可用其中的(　)描述。

A. 上下文有关文法　　B. 上下文无关文法　　C. 正规文法　　D. 短语结构文法

试题五十　运行下面的 C 程序代码段，会出现(　)错误。

```
int k=0;
for(;k<100;);
{k++;}
```

A. 变量未定义　　B. 静态语义　　C. 语法　　D. 动态语义

试题五十一　在数据库系统中，一般由 DBA 使用 DBMS 提供的授权功能为不同用户授权，其主要目的是为了保证数据库的(　)。

A. 正确性　　B. 安全性　　C. 一致性　　D. 完整性

试题五十二 (第1空)给定关系模式R(U,F)，其中：U为关系模式R中的属性集，F是U上的一组函数依赖。假设U={A_1， A_2， A_3， A_4}， F={A_1→A_2， A_1A_2→A_3， A_1→A_4， A_2→A_4}，那么关系R的主键应为( )。函数依赖集F中的( )是冗余的。

A．A_1　　B．A_1A_2　　C．A_1A_3　　D．$A_1A_2A_3$

试题五十三 (第2空)给定关系模式R(U,F)，其中：U为关系模式R中的属性集，F是U上的一组函数依赖。假设U={A_1， A_2， A_3， A_4}， F={A_1→A_2， A_1A_2→A_3， A_1→A_4， A_2→A_4}，那么关系R的主键应为( )。函数依赖集F中的( )是冗余的。

A．A_1→A_2　　B．A_1A_2→A_3　　C．A_1→A_4　　D．A_2→A_4

试题五十四 (第1空)给定关系R(A， B， C， D)和关系S(A， C， E， F)，对其进行自然连接运算R S后的属性列为( )个；与 $\sigma_{R.B>S.E}$(R S)等价的关系代数表达式为( )。

A．4　　B．5　　C．6　　D．8

试题五十五 (第2空)给定关系R(A， B， C， D)和关系S(A， C， E， F)，对其进行自然连接运算R S后的属性列为( )个；与 $\sigma_{R.B>S.E}$(R S)等价的关系代数表达式为( )。

A. $\sigma_{2<7}(R\times S)$

B. $\pi_{1,2,3,4,7,8}(\sigma_{1=5\wedge 2>7\wedge 3=6}(R\times S))$

C. $\sigma_{2<'7'}(R\times S)$

D. $\pi_{1,2,3,4,7,8}(\sigma_{1=5\wedge 2>'7'\wedge 3=6}(R\times S))$

试题五十六 下列查询B=“大数据”且F=“开发平台”，结果集属性列为A、B、C、F的关系代数表达式中，查询效率最高的是( )。

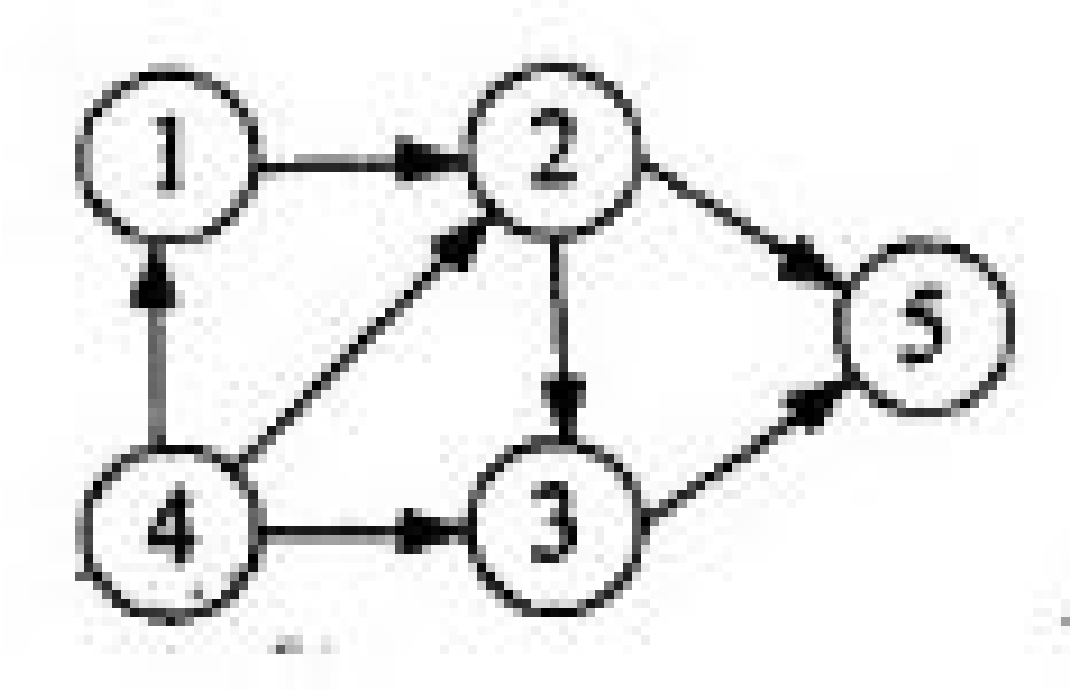
A. $\pi_{1,2,3,8}(\sigma_{2='大数据'\wedge 1=5\wedge 3=6\wedge 8='开发平台'}(R\times S))$

B. $\pi_{1,2,3,8}(\sigma_{1=5\wedge 3=6\wedge 8='开发平台'}(\sigma_{2='大数据'}(R)\times S))$

C. $\pi_{1,2,3,8}(\sigma_{2='大数据'\wedge 1=5\wedge 3=6}(R\times \sigma_{4='开发平台'}(S)))$

D. $\pi_{1,2,3,8}(\sigma_{1=5\wedge 3=6}(\sigma_{2='大数据'}(R)\times \sigma_{4='开发平台'}(S)))$

试题五十七 拓扑序列是有向无环图中所有顶点的一个线性序列，若有向图中存在弧或存在从顶点v到w的路径，则在该有向图的任一拓扑序列中，v一定在w之前。下面有向图的拓扑序列是( )。



A．4 1 2 3 5　　B．4 3 1 2 5　　C．4 2 1 3 5　　D．4 1 3 2 5

试题五十八　(第 1 空)设有一个包含 n 个元素的有序线性表。在等概率情况下删除其中的一个元素，若采用顺序存储结构，则平均需要移动(　)个元素；若采用单链表存储，则平均需要移动(　)个元素。

A．1　　B．(n-1)/2　　C．logn　　D．n

试题五十九　(第 2 空)设有一个包含 n 个元素的有序线性表。在等概率情况下删除其中的一个元素，若采用顺序存储结构，则平均需要移动(　)个元素；若采用单链表存储，则平均需要移动(　)个元素。

A．0　　B．1　　C．(n-1)/2　　D．n/2

试题六十　具有 3 个结点的二叉树有(　)种形态。

A．2　　B．3　　C．5　　D．7

试题六十一　以下关于二叉排序树(或二叉查找树、二叉搜索树)的叙述中，正确的是(　)。

A. 对二叉排序树进行先序、中序和后序遍历，都得到结点关键字的有序序列
B. 含有 n 个结点的二叉排序树高度为$(\log_2 n)+1$
C. 从根到任意一个叶子结点的路径上，结点的关键字呈现有序排列的特点
D. 从左到右排列同层次的结点，其关键字呈现有序排列的特点

试题六十二 （第 1 空）下表为某文件中字符的出现频率，采用霍夫曼编码对下列字符编码，则字符序列“bee”的编码为( )；编码“110001001101”的对应的字符序列为( )。

字符	a	b	c	d	e	f
频率(%)	45	13	12	16	9	5

A．10111011101　　B．10111001100　　C．001100100　　D．110011011

试题六十三 （第 2 空）下表为某文件中字符的出现频率，采用霍夫曼编码对下列字符编码，则字符序列“bee”的编码为( )；编码“110001001101”的对应的字符序列为( )。

字符	a	b	c	d	e	f
频率(%)	45	13	12	16	9	5

A．bad　　B．bee　　C．face　　D．bace

试题六十四 （第 1 空）两个矩阵 A_{m*n} 和 B_{n*p} 相乘，用基本的方法进行，则需要的乘法次数为 m*n*p 。多个矩阵相乘满足结合律，不同的乘法顺序所需要的乘法次数不同。考虑采用动态规划方法确定 M_i， $M_{(i+1)}$，…， Mj 多个矩阵连乘的最优顺序，即所需要的乘法次数最少。最少乘法次数用 m[i,j]表示，其递归式定义为：

其中 i 、 j 和 k 为矩阵下标，矩阵序列中 Mi 的维度为 $(p_{i-1})*p_i$ 采用自底向上的方法实现该算法来确定 n 个矩阵相乘的顺序，其时间复杂度为( )。若四个矩阵 M_1、 M_2、 M_3、 M_4 相乘的维度序列为 2 、 6 、 3 、 10 、 3 ，采用上述算法求解，则乘法次数为( )。

$$m[i,j]=\begin{cases}0 & i \geqslant j \\ \min\limits_{i<k<j}\{m[i,k]+m[k+1,j]+p_{i-1}p_kp_j\} & i<j\end{cases}$$

A．$O(n^2)$　　B．$O(n^2\lg n)$　　C．$O(n^3)$　　D．$O(n^3\lg n)$

试题六十五 （第 2 空）两个矩阵 A_{m*n} 和 B_{n*p} 相乘，用基本的方法进行，则需要的乘法次数为 m*n*p 。多个矩阵相乘满足结合律，不同的乘法顺序所需要的乘法次数不同。考虑采用动态规划方法确定 M_i， $M_{(i+1)}$，…， Mj 多个矩阵连乘的最优顺序，即所需要的乘法次数最少。最少乘法次数用 m[i,j]表示，其递归式定义为：

其中 i 、 j 和 k 为矩阵下标，矩阵序列中 Mi 的维度为 $(p_{i-1})*p_i$ 采用自底向上的方法实现

该算法来确定 n 个矩阵相乘的顺序，其时间复杂度为( )。若四个矩阵 M_1、 M_2、 M_3、 M_4 相乘的维度序列为 2 、 6 、 3 、 10 、 3 ，采用上述算法求解，则乘法次数为( )。

$$m[i,j]=\begin{cases}0 & i \geqslant j \\ \min\limits_{i\leqslant k<j}\{m[i,k]+m[k+1,j]+p_{i-1}p_kp_j\} & i<j\end{cases}$$

A．156　　B．144　　C．180　　D．360

试题六十六　(第 1 空)以下协议中属于应用层协议的是( )，该协议的报文封装在( )。
A．SNMP　　B．ARP　　C．ICMP　　D．X.25

试题六十七　(第 2 空)以下协议中属于应用层协议的是( )，该协议的报文封装在( )。
A．TCP　　B．IP　　C．UDP　　D．ICMP

试题六十八　某公司内部使用 wb.xyz.com.cn 作为访问某服务器的地址，其中 wb 是( )。
A. 主机名　　B. 协议名　　C. 目录名　　D. 文件名

试题六十九　如果路由器收到了多个路由协议转发的关于某个目标的多条路由，那么决定采用哪条路由的策略是( )。
A. 选择与自己路由协议相同的　　B. 选择路由费用最小的　　C. 比较各个路由的管理距离　　D. 比较各个路由协议的版本

试题七十　与地址 220.112.179.92 匹配的路由表的表项是( )。
A．220.112.145.32/22　　B．220.112.145.64/22
C．220.112.147.64/22　　D．220.112.177.64/22

试题七十一　(第 1 空)Software entities are more complex for their size than perhaps any other human construct, because no two parts are alike (at least above the statement level). If they are, we make the two similar parts into one, a( ), open or closed. In this respect software systems differ profoundly from computers,buildings, or automobiles, where repeated elements abound. Digital computers are themselves more complex than most things people build; they have very large numbers of states. This makes conceiving, describing, and

testing them hard. Software systems have orders of magnitude more ( )than computers do.

Likewise, a scaling-up of a software entity is not merely a repetition of the same elements in larger size; it is necessarily an increase in the number of different elements. In most cases, the elements interact with each other in some( )fashion,and the complexity of the whole increases much more than linearly.

The complexity of software is a(an)( )property, not an accidental one. Hence descriptions of a software entity that abstract away its complexity often abstract away its essence.Mathematics and the physical sciences made great strides for three centuries by constructing simplified models of complex phenomena, deriving properties from the models, and verifying those properties experimentally. This worked because the complexities( )in the models were not the essential properties of the phenomena. It does not work when the complexities are the essence.

Many of the classical problems of developing software products derive from this essential complexity and its nonlinear increases with size. Not only technical problems but management problems as well come from the complexity.

A. task　B. job　C. subroutine　D. program

试题七十二 （第2空）Software entities are more complex for their size than perhaps any other human construct, because no two parts are alike (at least above the statement level). If they are, we make the two similar parts into one, a( ), open or closed. In this respect software systems differ profoundly from computers,buildings, or automobiles, where repeated elements abound.

Digital computers are themselves more complex than most things people build; they have very large numbers of states. This makes conceiving, describing, and testing them hard. Software systems have orders of magnitude more ( )than computers do.

Likewise, a scaling-up of a software entity is not merely a repetition of the same elements in larger size; it is necessarily an increase in the number of different elements. In most cases, the elements interact with each other in some( )fashion,and the complexity of the whole increases much more than

linearly.

The complexity of software is a(an)( )property, not an accidental one. Hence descriptions of a software entity that abstract away its complexity often abstract away its essence.Mathematics and the physical sciences made great strides for three centuries by constructing simplified models of complex phenomena, deriving properties from the models, and verifying those properties experimentally. This worked because the complexities( )in the models were not the essential properties of the phenomena. It does not work when the complexities are the essence.

Many of the classical problems of developing software products derive from this essential complexity and its nonlinear increases with size. Not only technical problems but management problems as well come from the complexity.

A. states B. parts C. conditions D. expressions

试题七十三 (第3空)Software entities are more complex for their size than perhaps any other human construct, because no two parts are alike (at least above the statement level). If they are, we make the two similar parts into one, a( ), open or closed. In this respect software systems differ profoundly from computers,buildings, or automobiles, where repeated elements abound.

Digital computers are themselves more complex than most things people build; they have very large numbers of states. This makes conceiving, describing, and testing them hard. Software systems have orders of magnitude more ( )than computers do.

Likewise, a scaling-up of a software entity is not merely a repetition of the same elements in larger size; it is necessarily an increase in the number of different elements. In most cases, the elements interact with each other in some( )fashion,and the complexity of the whole increases much more than linearly.

The complexity of software is a(an)( )property, not an accidental one. Hence descriptions of a software entity that abstract away its complexity often abstract away its essence.Mathematics and the physical sciences made great strides for three centuries by constructing simplified models of complex phenomena, deriving properties from the models, and verifying those properties

experimentally. This worked because the complexities( )in the models were not the essential properties of the phenomena. It does not work when the complexities are the essence.

Many of the classical problems of developing software products derive from this essential complexity and its nonlinear increases with size. Not only technical problems but management problems as well come from the complexity.

A. linear B. nonlinear C. parallel D. additive

试题七十四 (第4空)Software entities are more complex for their size than perhaps any other human construct, because no two parts are alike (at least above the statement level). If they are, we make the two similar parts into one, a( ), open or closed. In this respect software systems differ profoundly from computers,buildings, or automobiles, where repeated elements abound.

Digital computers are themselves more complex than most things people build; they have very large numbers of states. This makes conceiving, describing, and testing them hard. Software systems have orders of magnitude more ( )than computers do.

Likewise, a scaling-up of a software entity is not merely a repetition of the same elements in larger size; it is necessarily an increase in the number of different elements. In most cases, the elements interact with each other in some( )fashion,and the complexity of the whole increases much more than linearly.

The complexity of software is a(an)( )property, not an accidental one. Hence descriptions of a software entity that abstract away its complexity often abstract away its essence.Mathematics and the physical sciences made great strides for three centuries by constructing simplified models of complex phenomena, deriving properties from the models, and verifying those properties experimentally. This worked because the complexities( )in the models were not the essential properties of the phenomena. It does not work when the complexities are the essence.

Many of the classical problems of developing software products derive from this essential complexity and its nonlinear increases with size. Not only technical problems but management problems as well come from the complexity.

A. surface B. outside C. exterior D. essential

试题七十五 (第5空)Software entities are more complex for their size than perhaps any other human construct, because no two parts are alike (at least above the statement level). If they are, we make the two similar parts into one, a( ), open or closed. In this respect software systems differ profoundly from computers,buildings, or automobiles, where repeated elements abound.

Digital computers are themselves more complex than most things people build; they have very large numbers of states. This makes conceiving, describing, and testing them hard. Software systems have orders of magnitude more ( )than computers do.

Likewise, a scaling-up of a software entity is not merely a repetition of the same elements in larger size; it is necessarily an increase in the number of different elements. In most cases, the elements interact with each other in some( )fashion,and the complexity of the whole increases much more than linearly.

The complexity of software is a(an)( )property, not an accidental one. Hence descriptions of a software entity that abstract away its complexity often abstract away its essence.Mathematics and the physical sciences made great strides for three centuries by constructing simplified models of complex phenomena, deriving properties from the models, and verifying those properties experimentally. This worked because the complexities( )in the models were not the essential properties of the phenomena. It does not work when the complexities are the essence.

Many of the classical problems of developing software products derive from this essential complexity and its nonlinear increases with size. Not only technical problems but management problems as well come from the complexity.

A. fixed　B. included　C. ignored　D. stabilized

试题一 答案： A **解析：**

本题查计算机系统基础知识。

指令周期是执行一条指令所需要的时间，一般由若干个机器周期组成，是从取指令、分析指令到执行完所需的全部时间。CPU执行指令的过程中，根据时序部件发出的时钟信号按部就班进行操作。在取指令阶段读取到的是指令，在分析指令和执行指令时，需要操作数时再去读操作数。

试题二 答案： C **解析：**

本题考查计算机系统基础知识。

CPU 首先从程序计数器(PC)获得需要执行的指令地址，从内存(或高速缓存)读取到的指令则暂存在指令寄存器(IR)，然后进行分析和执行。

试题三 答案： B **解析：**

本题考查计算机系统基础知识。

浮点格式表示一个二进制数 N 的形式为 N=2eXF，其中 E 称为阶码，F 叫做尾数。在浮点表示法中，阶码通常为含符号的纯整数，尾数为含符号的纯小数。

指数为纯整数，阶符 1 位、阶码 6 位在补码表示方式下可表示的最大数为 $63(2^6-1)$，最小数为$-64(-2^6)$。尾数用补码表示时最小数为−1 、最大数为 $1-2^{-8}$，因此该浮点表示的最小数为-2^{63}，最大数为$(1-2^{-8})\times 2^{63}$。

试题四 答案： C **解析：**

本题考查计算机系统基础知识。

海明码是利用奇偶性来检错和纠错的校验方法。海明码的构成方法是：在数据位之间插入 k 个校验位，通过扩大码距来实现检错和纠错。

设数据位是 n 位，校验位是 k 位，则 n 和 k 必须满足以下关系：$2^k-1\geqslant n+k$

若数据信息为 n=16 位，则 k=5 是满足 $2^k-1\geqslant n+k$ 的最小值。

试题五 答案： D **解析：**

本题考查计算机系统基础知识。

对于该指令流水线，建立时间为 4△t+2△t+3△t=9△t，此后每 4△t 执行完一条指令，即执行完 100 条指令的时间为 9△t+99*4△t=405△t。

试题六 答案： D **解析：**

本题考查计算机系统基础知识。

存储系统采用 Cache 技术的主要目的是提高存储器的访问速度，因此是由硬件自动完成 Cache 与主存之间的地址映射。

试题七 答案： A **解析：**

本题考查网络安全相关基础知识。

RSA 基于大数定律，通常用于对消息摘要进行签名； IDEA 和 RC4 适宜于进行数据传输加密；MD5 为摘要算法。

试题八 答案： D **解析：**

本题考查数字签名方面的基础知识。

数字签名用于通信的 A.、 B 双方，使得 A 向 B 发送签名的消息 P ，提供以下服务：

①B 可以验证消息 P 确实是来源于 A ；

②A 不能否认发送过消息 P ；

③B 不能编造或改变消息 P 。

数字签名首先需要生成消息摘要，使用非对称加密算法以及私钥对摘要进行加密。接收方使用发送放的公钥对消息摘要进行验证。

试题九 答案： C **解析：**

设备防雷击属于物理线路安全措施，入侵检测和流量控制属于网络安全措施，漏洞发现与补丁管理属于系统安全措施。

试题一十 答案： B **解析：**

发明专利权的期限为二十年，实用新型专利权和外观设计专利权的期限为十年，均自申请日起计算。专利保护的起始日是从授权日开始，有下列情形之一的，专利权在期限届满前终止：①没有按照规定缴纳年费的；②专利权人以书面声明放弃其专利权的。还有一种情况就是专利期限到期，专利终止时，保护自然结束。

商标权保护的期限是指商标专用权受法律保护的有效期限。我国注册商标的有效期为十年，自核准注册之日起计算。注册商标有效期满可以续展；商标权的续展是指通过指定程序，延续原注册商标的有效期限，使商标注册人继续保持其注册商标的专用权。

在著作权的期限内，作品受著作权法保护；著作权期限届满，著作权丧失，作品进入公有领域。

法律上对商业秘密的保密期限没有限制，只要商业秘密的四个基本特征没有消失，权利人可以将商业秘密一直保持下去。权利人也可以根据实际状况，为商业秘密规定适当的期限。

试题一十一 答案： D 解析：

当两个以上的申请人分别就同样的发明创造申请专利的，专利权授给最先申请的人。如果两个以上申请人在同一日分别就同样的发明创造申请专利的，应当在收到专利行政管理部门的通知后自行协商确定申请人。如果协商不成，专利局将驳回所有申请人的申请，即均不授予专利权。我国专利法规定：“两个以上的申请人分别就同样的发明创造申请专利的，专利权授予最先申请的人。”我国专利法实施细则规定：“同样的发明创造只能被授予一项专利。依照专利法第九条的规定，两个以上的申请人在同一日分别就同样的发明创造申请专利的，座当在收到国务院专利行政部门的通知后自行协商确定申请人。”

试题一十二 答案： B 解析：

我国商标注册以申请在先为原则，使用在先为补充。当两个或两个以上申请人在同一种或者类似商品上申请注册相同或者近似商标时，申请在先的人可以获得注册。对于同日申请的情况，商标法及其实施条例规定保护先用人的利益，使用在先的人可以获得注册“使用”包括将商标用于商品-、商品包装、容器以及商品交易书或者将商标用于广告宣传、展览及其他商业活动中。如果同日使用或均未使用，则采取申请人之间协商解决，不愿协商或者协商不成的，由各申请人抽签决定。商标局通知各申请人以抽签的方式确定一个申请人，驳回其他人的注册申请。商标局已经通知但申请人未参加抽签的，视为放弃申请。

试题一十三 答案： A 解析：

音调(Pitch)用来表示人的听觉分辨.个声音的调子高低的程度，主要由声音的频率决定，同时也与声音强度有关。对一定强度的纯音，音调随频率的升降而升降；对定频率的纯音、低频纯音的音调随声强增加而下降，高频纯者的音调却随强度增加而上升。

音色(Timbre)是指声音的感觉特性，不同的人声和不同的声响都能区分为不同的音色，即音频泛音或谐波成分。

音高是指各种不同高低的声音.(即首的高度)，是首的基本特征的一种。

在 FM 方式音乐合成器中，数字载波波形和调制波形有很多种，不同型号的 FM 合成器所选用的波形也不同。各种不同乐音的产生是通过组合各种波形和各种波形参数并采用各种不同的方法实现的。改变数字载波频率可以改变乐音的音调，改变它的幅度可以改变乐音的音高。

试题一十四 答案： C 解析：

音调(Pitch)用来表示人的听觉分辨.个声音的调子高低的程度，主要由声音的频率决定，

同时也与声音强度有关。对一定强度的纯音，音调随频率的升降而升降；对定频率的纯音、低频纯音的音调随声强增加而下降，高频纯者的音调却随强度增加而上升。

音色(Timbre)是指声音的感觉特性，不同的人声和不同的声响都能区分为不同的音色，即音频泛音或谐波成分。

音高是指各种不同高低的声音.(即首的高度)，是首的基本特征的一种。

在 FM 方式音乐合成器中，数字载波波形和调制波形有很多种，不同型号的 FM 合成器所选用的波形也不同。各种不同乐音的产生是通过组合各种波形和各种波形参数并采用各种不同的方法实现的。改变数字载波频率可以改变乐音的音调，改变它的幅度可以改变乐音的音高。

试题一十五 答案： D 解析：

本题考查软件设计的基础知识。

结构化设计主要包括：

①体系结构设计：定义软件的主要结构元素及其关系。

②数据设计：基于实体联系图确定软件涉及的文件系统的结构及数据库的表结构。

③接口设计：描述用户界面，软件和其他硬件设备、其他软件系统及使用人员的外部接口，以及各种构件之间的内部接口。

④过程设计：确定软件各个组成部分内的算法及内部数据结构，并选定某种过程的表达形式来描述各种算法。

试题一十六 答案： C 解析：

本题考查敏捷方法的基础知识。

在 20 世纪 W 年代后期，一些开发人员抵制严格化软件开发过程，试图强调灵痹性在快速有效的软件傘产中的作用：，提出了敏捷宣言，即个人和交互胜过过程和工具；可以运行的软件胜过面面俱到的文档；与客户合作胜过合同谈判；对变化的反应胜过遵循计划。

基于这些基本思想，有很多敏捷过程的典型方法。其中，极限编程 XP 是激发开发人员创造性、使得管理负担最小的一组技术；水晶法 Crystal 认为每一个不同的项目都需要一套不同的策略、约定和方法论；并列争球法(Scrum)使用迭代的方法，其中把每 30 天一次的迭代成为一个冲刺，并按需求的优先级来实现产品。多个自组织和自治小组并行地递增实现产品，并通过简短的日常情况会议进行协调。

自适应软件开发(ASD)有六个基本的原则：

①在自适应软件开发中，有一个使命作为指导，它设立了项目的目标，但并不描述如何达到这个目标；

②特征被视为客户键值的关键，因此，项 0 是围绕着构造的构件来组织并实现特征；
③过程中的迭代是很重要的因此重做与做同样重要，变化也包含其中；
④变化不视为是一种更正，而是对软件开发实际情况的调整；
⑤确定的交付时间迫使开发人员认证考虑每一个生产版本的关键需求；
⑥风险也包含其中，它使开发人员首先跟踪最艰难的问题。

试题一十七 答案： D 解析：

本题考查软件项目管理的基础知识。

活动图是描述一个项目中各个工作任务相互依赖关系的一种模型，项目的很多重要特性可以通过分析活动图得到，如估算项目完成时间，计算关键路径和关键活动等。根据上图计算出关键路径为 A-B-C-E-F-J 和 A-B-D-G-F-J，其长度为 18。关键路径上的活动均为关键活动。活动 BC 在关键路径上，因此松弛时间为 0。活动 BF 不在关键路径±，包含该活动的最长路径为 A-B-F-J，其长度为 11，因此该活动的松弛时间为 18-11=7。

试题一十八 答案： A 解析：

本题考查软件项目管理的基础知识。

活动图是描述一个项目中各个工作任务相互依赖关系的一种模型，项目的很多重要特性可以通过分析活动图得到，如估算项目完成时间，计算关键路径和关键活动等。根据上图计算出关键路径为 A-B-C-E-F-J 和 A-B-D-G-F-J，其长度为 18。关键路径上的活动均为关键活动。活动 BC 在关键路径上，因此松弛时间为 0。活动 BF 不在关键路径±，包含该活动的最长路径为 A-B-F-J，其长度为 11，因此该活动的松弛时间为 18-11=7。

试题一十九 答案： D 解析：

本题考查软件项目管理的基础知识。

存在多种软件项目管理的成本估算方法。其中专家估算方法主要依赖于专家的背景和经验，具有较大的主观性。 Wolverton 模型基于一个成本矩阵，定义不同的软件类型(如控制、输入/输出等)和难易(容易和困难)的成本，基于此计算软件开发的成本。 COCOMO 模型将规模视为成本的主要因素，考虑多个成本驱动因子。在后来的版本 COCOMOII 中，还考虑了软件开发的不同阶段，包含三个阶段性模型，即应用组装模型、早期设计阶段模型和体系结构阶段模型。

试题二十 答案： B 解析：

本题考查逻辑运算知识。

由“逻辑与”“逻辑或”运算构造的逻辑表达可采用短路计算的方式求值。

“逻辑与”运算“&&”的短路运算逻辑为： a&&b 为真当且仅当 a 和 b 都为真，当 a 为假，无论 b 的值为真还是假，该表达式的值即为假，也就是说此时不需要再计算 b 的值。

“逻辑或”运算“||”的短路运算逻辑为： a||b 为假当且仅当 a 和 b 都为假，当 a 为真，无论 b 的值为真还是假，该表达式的值即为真，也就是说此时不需要再计算 b 的值。

对逻辑表达式“x&&(y||！ z)”进行短路计算方式求值时， x 为假则整个表达式的值即为假，不需要计算 y 和 z 的值。若 x 的值为真，则再根据 y 的值决定是否需要计算 z 的值， y 为真就不需要计算 Z 的值， y 为假则需要计算 z 的值。

试题二十一 答案： C 解析：

本题考查程序语言基础知识。

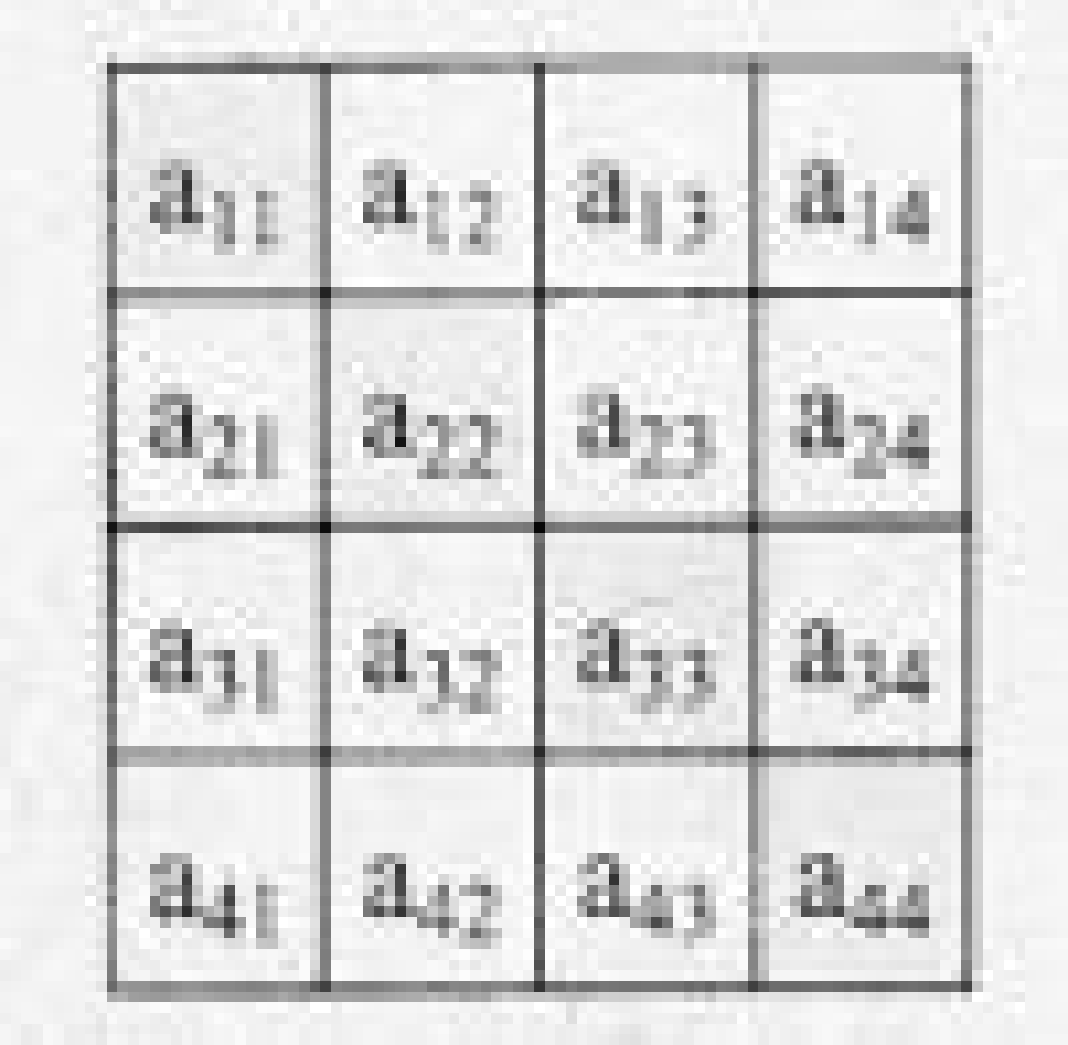
传值调用和引用调用是常用的两种参数传递方式。在传值调用方式下，是将实参的值传递给形参，该传递是单方向的，调用结束后不会再将形参的值传给实参。在引用调用方式下，实质上是将实参的地址传递给形参，借助指针在间接访问数据方式下(或者将形参看作是实参的别名)，在被调用函数中对形参的修改实质上是对实参的修改。

试题二十二 答案： B 解析：

本题考查数据存储知识。

二维数组 a[1..N， 1..N]用来表示一个 N*N 的方阵，主对角线上元素的行下标和列下标相同，以 4*4 的矩阵为例，如下图所示。

对于主对角线中的元素，无论按行方式排列还是按列方式排列，其在序列中的位置都是相同的。



试题二十三 答案： D 解析：

本题考查操作系统基础知识。

实时是指计算机对于外来信息能够以足够快的速度进行处理，并在被控对象允许的时间范围内做出快速响应。因此，实时操作系统与分时操作系统的第一点区别是交互性强弱不同，分时系统交互型强，实时系统交互性弱但可靠性要求高；第二点区别是对响应时间的敏感牲强，对随机发幸的外部事件必须在被控制对象规定的时间做出及时响应并对其进行处理；第三点区别是系统的设计目标不同，分时系统是设计成一个多用方的通用系统，交互能力强；而实时系统大都是专用桌统。

试题二十四 答案： C 解析：

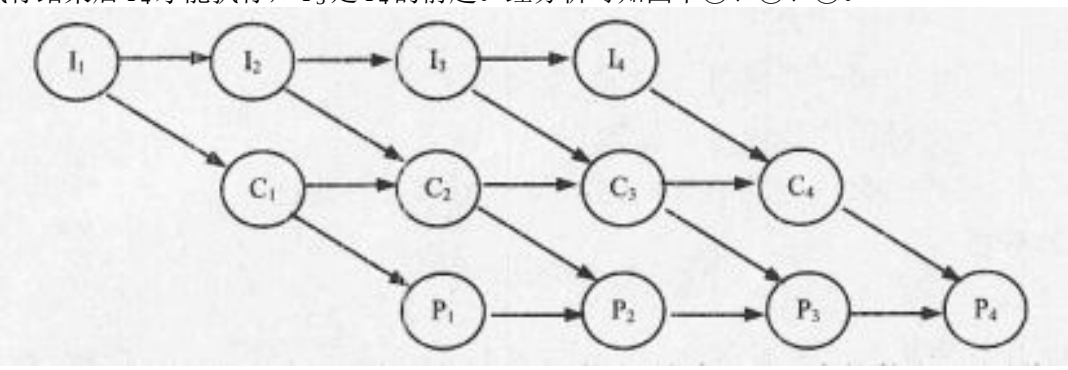
本题考蜜操作系统基础知识。

前趋图是一个有向无环图，由节点和有向边组成，节点代表各程序段的操作，而节点间的

有向边表示两个程序段操作之间存在的前趋关系(“→”)。程序段 P_i 和 J_j 的前趋关系可表示成 $P_i \rightarrow P_j$，其中 P_i 是 P_j 的前趋， P_j 是 P_i 的后继，其含义是 P_i 执行结束后 P_j 才能执行。本题完整的前趋图如下图所示，具体分析如下。

根据题意， I_1 执行结束后 C_1 才能执行， P_1 执行结束后 P_2 才能执行，因此 I_1 是 C_1 的前趋，P_1 是 P_2 的前趋。可见，①、②分别为 C_1 、 P_2 。

根据题意， I_2 执行结束后 I_3 才能执行，即 I_2 是 I_3 前趋，所以③应为 I_3 。又因为计算机系统中只有一个 CPU 和一台输出设备，所以 C_3 执行结束后 C_4 才能执行， C_3 是 C_4 的前趋； P_3 执行结束后 P_4 才能执行， P_3 是 P_4 的前趋。经分析可知图中③、④、⑤。



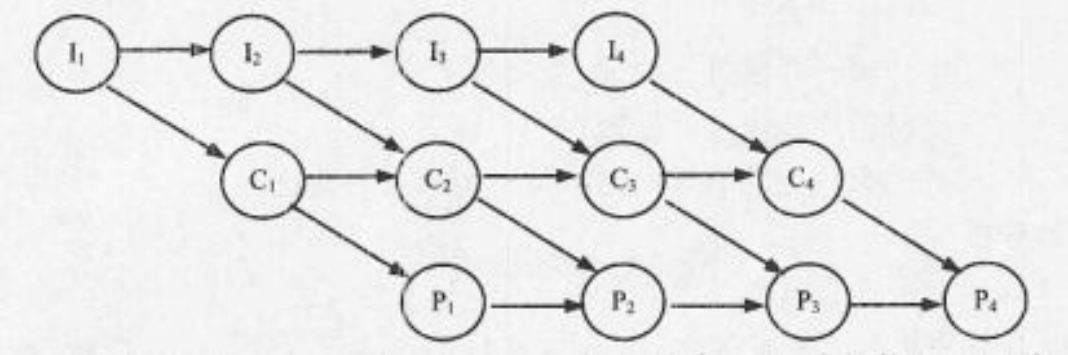
试题二十五 答案： D 解析：

本题考蜜操作系统基础知识。

前趋图是一个有向无环图，由节点和有向边组成，节点代表各程序段的操作，而节点间的有向边表示两个程序段操作之间存在的前趋关系(“→”)。程序段 Pi 和 Jj 的前趋关系可表示成 $P_i \rightarrow P_j$，其中 P_i 是 P_j 的前趋， P_j 是 P_i 的后继，其含义是 P_i 执行结束后 P_j 才能执行。本题完整的前趋图如下图所示，具体分析如下。

根据题意， I_1 执行结束后 C_1 才能执行， P_1 执行结束后 P_2 才能执行，因此 I_1 是 C_1 的前趋，P_1 是 P_2 的前趋。可见，①、②分别为 C_1 、 P_2 。

根据题意， I_2 执行结束后 I_3 才能执行，即 I2 是 I3 前趋，所以③应为 I_3 。又因为计算机系统中只有一个 CPU 和一台输出设备，所以 C_3 执行结束后 C_4 才能执行， C_3 是 C_4 的前趋； P_3 执行结束后 P_4 才能执行， P_3 是 P_4 的前趋。经分析可知图中③、④、⑤。



试题二十六 答案： B 解析：

本题考查操作系统页式存储管理方面的基础知识。从图中可见，页内地址的长度是 13 位，2^{11}=8192，即 8K；页号部分的地址长度是 11 位，每个段最大允许有 2^{11}=2048 个页；段号部分的地址长度是 8 位，2^8=256，最多可有 256 个段。

试题二十七 答案： B 解析：

本题考查操作系统 PV 操作方面的基础知识。

系统采用 PV 操作实现进程同步与互斥，若有 n 个进程共享 3 台扫描仪，那么信号量 S 初值应为 3。若系统当前信号量 S 的值为-1，此时，P_1，P_2 又分别执行了 1 次 P(S)操作，那么程序执行 P(S)操作时，信号量 S 的值减 1 后等于-2；当 P_2 进程执行 P(S)操作时，信号量 S 的值减 1 后等于-3。

试题二十八 答案： D 解析：

本题考查操作系统文件管理方面的基础知识。

根据题意，若磁盘的容量为 300GB，物理块的大小为 1MB，则该磁盘的物理块数为 300*1024=307200 个，位示图的大小为 307200/32=9600 个字。

试题二十九 答案： A 解析：

本题考查软件开发过程模型的基础知识。

瀑布模型将开发阶段描述为从一个阶段瀑布般地转换到另一个阶段的过程。

原型模型中，开发人员快速地构造整个系统或者系统的一部分以理解或澄清问题。增量模型是把软件产品作为一系列的增量构件来设计、编码、集成和测试，每个构件由多个相互作用的模块组成，并且能够完成特定的功能。

喷泉模型开发过程章中以用户需求为动力，以对象为驱动，适合于面向对象的开发方法。
在上述几种开发过程模型中，瀑布模型不能适应变化的需求。

试题三十 答案： C 解析：

本题考查软件开发过程模型的基础知识。

瀑布模型将开发阶段描述为从一个阶段瀑布般地转换到另一个阶段的过程。

原型模型中，开发人员快速地构造整个系统或者系统的一部分以理解或澄清问题。增量模型是把软件产品作为一系列的增量构件来设计、编码、集成和测试，每个构件由多个相互作用的模块组成，并且能够完成特定的功能。

喷泉模型开发过程章中以用户需求为动力，以对象为驱动，适合于面向对象的开发方法。
在上述几种开发过程模型中，瀑布模型不能适应变化的需求。

试题三十一 答案： D 解析：

本题考查软件质量的塞础知识。

ISO/IEC 软件质量模型定义了六个软件质量特性，即功能性、可靠性、易使用性、效率、可维护性和可移植性。对每个质量特性定义其子特性。其中可移植性包括子特性：适应性、易安装性、一致性和易替换性。

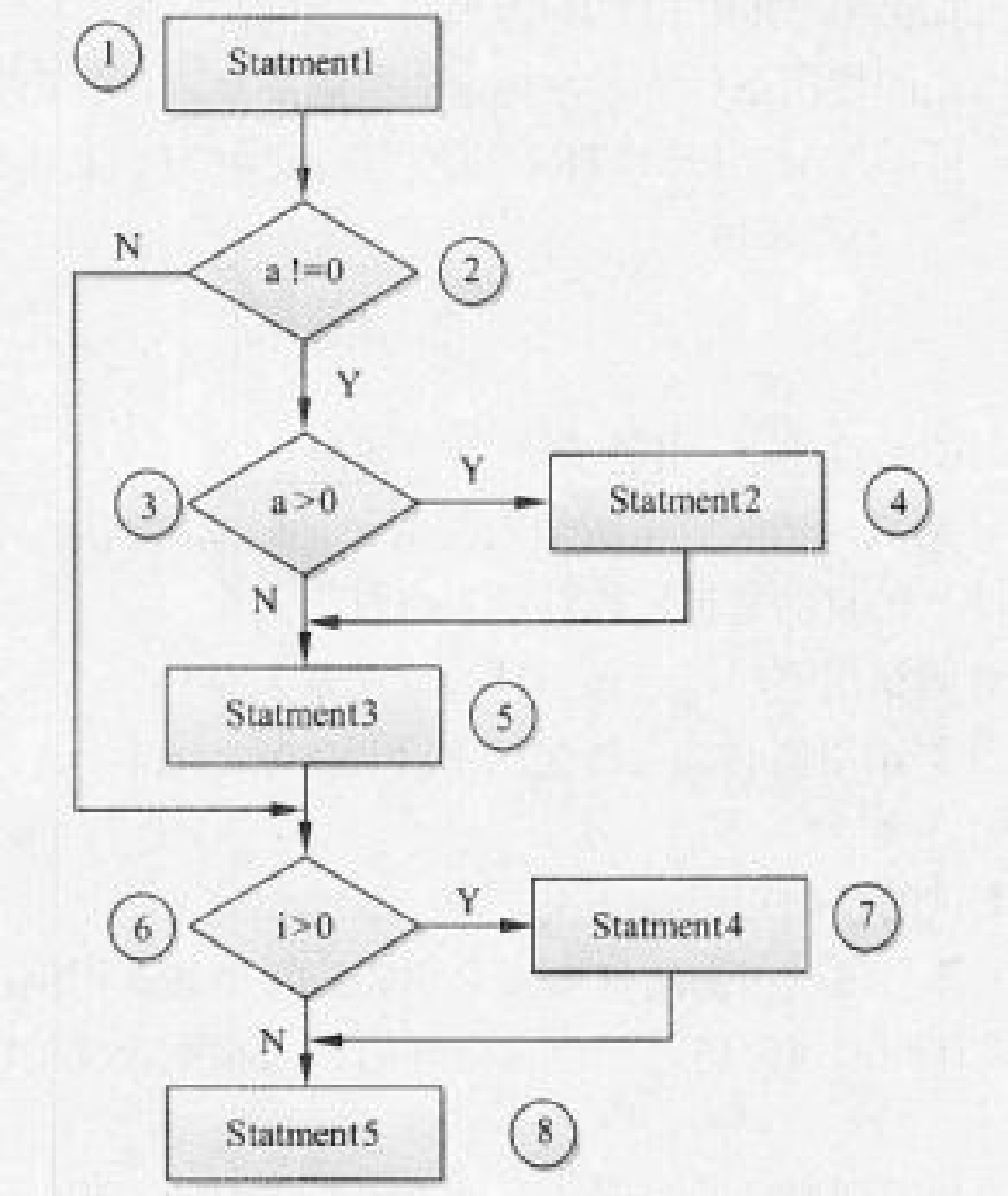
试题三十二 答案： C 解析：

本题考查软件测试的基础知识。

白盒测试和黑盒测试是两种最常用的软件测试方法。路径覆盖是白盒测试的一种具体方法。路径覆盖是指设计若干个测试用例，覆盖程序中的所有路径。

路径覆盖应使程序中每一条可能的路径至少执行一次。该流程图中一共有六条路径：①②③④⑤⑥⑦⑧，①②③④⑤⑥⑧，①②③⑤⑥⑦⑧，①②③⑤⑥⑧，①②⑥⑦⑧，①②⑥⑧，因此，实现路径覆盖至步需要 6 个测试用例。

McCabe 度量法是一种基于程序控制流的复杂性度量方法，环路复杂性为 V(G)=m-n+2，图中 m=10， n=8， V(G)=10-8+2=4 。



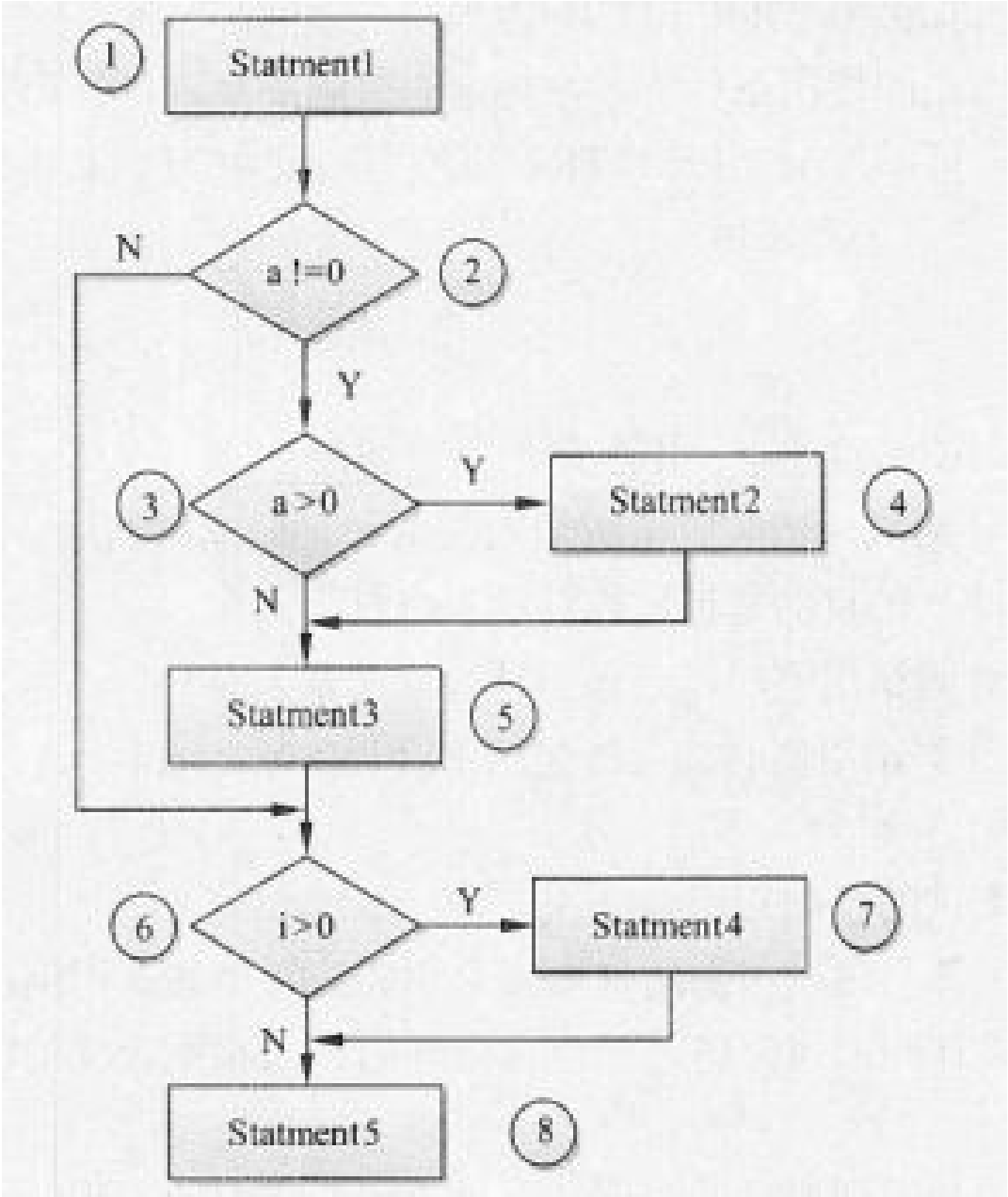
试题三十三 答案： D **解析：**

本题考查软件测试的基础知识。

白盒测试和黑盒测试是两种最常用的软件测试方法。路径覆盖是白盒测试的一种具体方法。路径覆盖是指设计若干个测试用例，覆盖程序中的所有路径。

路径覆盖应使程序中每一条可能的路径至少执行一次。该流程图中一共有六条路径：①②③④⑤⑥⑦⑧，①②③④⑤⑥⑧，①②③⑤⑥⑦⑧，①②③⑤⑥⑧，①②⑥⑦⑧，①②⑥⑧，因此，实现路径覆盖至步需要 6 个测试用例。

McCabe 度量法是一种基于程序控制流的复杂性度量方法，环路复杂性为 V(G)=m-n+2，图中 m=10， n=8， V(G)=10-8+2=4。



试题三十四 答案： B **解析：**

本题考查软件质量基础知识。

可靠性、可用性和可维护性是软件的质量属性，软件工程中，用 0-1 之间的数来度量。

可靠性是指一个系统对于给定的时间间隔内、在给定条件下无失效运作的概率。可以用 MTTF/(1+MTTF)来度量，其中 MTTF 为平均无故障时间。

可用性是在给定的时间点上，一个系统能够按照规格说明正确运作的概率。可以用 MTBF/(

1+MTBF)来度量，其中 MTBF 为平均失效间隔时间。

可维护性是在给定的使用条件下，在规定的时间间隔内，使用规定的过程和资源完成维护活动的概率。可以用 1/(1+MTTR)来度量，其中 MTTR 为平均修复时间。

试题三十五 答案： B 解析：

本题考查软件测试的基础知识。

在设计测试用例时应考虑输入数据和预期输出结果、在设计测试用例时，应该包括合理的输入条件、在设计测试用例时，应该包括不合理的输入条件选项都与测试用例的基本概念相关，每个测试用例应该包含输入数据和预期输出结果。在设计测试用例时，要包含合理的输入和不合理的输入。因此，这三个选项均正确。

软件测试的目的是发现更多的错误，而不是证明软件的正确性。

试题三十六 答案： C 解析：

本题考查软件设计的基础知识。

模块间的耦合和模块的内聚是度量模块独立性的两个准则。内聚是模块功能强度的度量，即模块内部各个元素彼此结合的紧密程度。一个模块内部各个元素之间的紧密程度越高，则其内聚性越高，模块独立性越好。模块内聚类型主要有以下几类：

偶然内聚，巧合内聚：指一个模块内的各处理元素之间没有任何联系。

逻辑内聚：指模块内执行 g 干个逻辑上相似的功能，通过参数确定该模块完成哪一个功能。

时间内聚：把需要同时执行的动作组合在一起形成的模块。

过程内聚：指一个模块完成多个任务，这些任务必须按指定的过程执行。

通信内聚：指模块内的所有处理元素都在同一个数据结构上操作，或者各处理使用相同的输入数据或产生相同的输出数据。

顺序内聚：指一个模块中的各个处理元素都密切相关于同一个功能且必须顺序执行，前一个功能元素的输出就是下一功能元素的输入。

功能内聚：指模块内的所有元素共同作用完成一个功能，缺一不可。

本题中，逻辑和过程对相同的数据结构操作，属于通信内聚。

试题三十七 答案： D 解析：

本题考查面向对象的基础知识。

在面向对象系统中，对象是基本的运行时实体，它既包括数据(属性)，也包括作用于数据的操作(行为)，访问对象的这些操作也称为接口。 1 组大体上相似的对象定义为一个类。

一个类所包含的方法和数据描述一组对象的共同行为和属性，这些对象共享这些行为和属性。有些类之间存在一般和特殊关系，在定义和实现一个类的时候，可以在一个已经存在的类的基础上来进行，把这个已经存在的类所定义的内容作为自己的内容，并加入新的内容，这种机制就是父类和子类之间共享数据和方法的机制，即继承。在子类定义时，可以继承它的父类(或祖先类)中的属性和方法，也可以重新定义父类中已经定义的方法，其方法可以对父类中方法进行覆盖，即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。在继承的支持下，不同对象在收到同一消息是可以产生不同的结果，这是由于对通用消息的实现细节由接收对象自行决定的缘故，这就是多态。

试题三十八 答案： D 解析：

本题考查面向对象的基本知识。

多态的实现受到继承的支持，利用类的继承的层次关系，把具有通用功能的消息存放在高层次，而不同的实现这一功能的行为放在较低层次。一个对象发送通用消息请求服务时，要根据接收对象的具体情况将求的操作与实现的方法进行连接，即动态绑定，以实现在这些低层次上生成的对传给用消息以不同的响应。

试题三十九 答案： C 解析：

本题考查面向对象的基本知识。

面向对象分析的目的是为了获得对应用问题的理解，以确定系统的功能、性能要求。面向对象分析方法是将数据和功能结合在一起作为一个综合对象来考虑。面向对象分析技术可以将系统的行为和信息间的关系表示为迭代构造特征。面向对象分析包含 5 个活动：认定对象、组织对象、描述对象间的相互作用、定义对象的操作、定义对象的内部信息。

试题四十 答案： C 解析：

本题考查面向对象和统一建模语(UML)的基础知识。

状态图(statediagram)展现了一个状态机，用于描述一个对象在其生存期间的动态行为，表现为 4 个对象所经历的状态序列，它由状态、转换、事件和活动组成。状态图关注系统的动态视图，它对于接口、类和协作的行为建模尤为重要强调对象行为的事件顺序。状态图通常包括简单状态和组合状态、转换(事件和动作)。

可以用状态图对系统的动态方面建模。这些动态方面可以包括出现在系统体系结构的任何视图中的任何一种对象的按事件排序的行为，这些对象包括类(各主动类)、接口、构件和节点。

当对象处于某个状态时，这个状态被称为徼活状态(activestate)。任何从激活状态出发的转换所标识的事件被检测到发生时，进行转换，而从当前状态出发的事件如果没有标注所检测到的事件名称，就忽略该事件，不激发任何转换，当前状态仍然是激活状态。
本题叙述中图示状态 B 内嵌套了 B1 、 B2 、 B3 和 B4 。咨激活状态是 B 且内嵌为状态 B2 时，如果发生事件 e2 ，则转移到 C3 状态；如果当前激活状态 B 的子状态不是 B2 ，则事件 e2 发生后，不激发状态转换。当激活状态为 B 时，不论内嵌状态是哪个，则发生事件 e1 后，激活状态转换到 C1 ；或者 B 中内嵌的两个结束状态均达到时，会离开状态 B 。

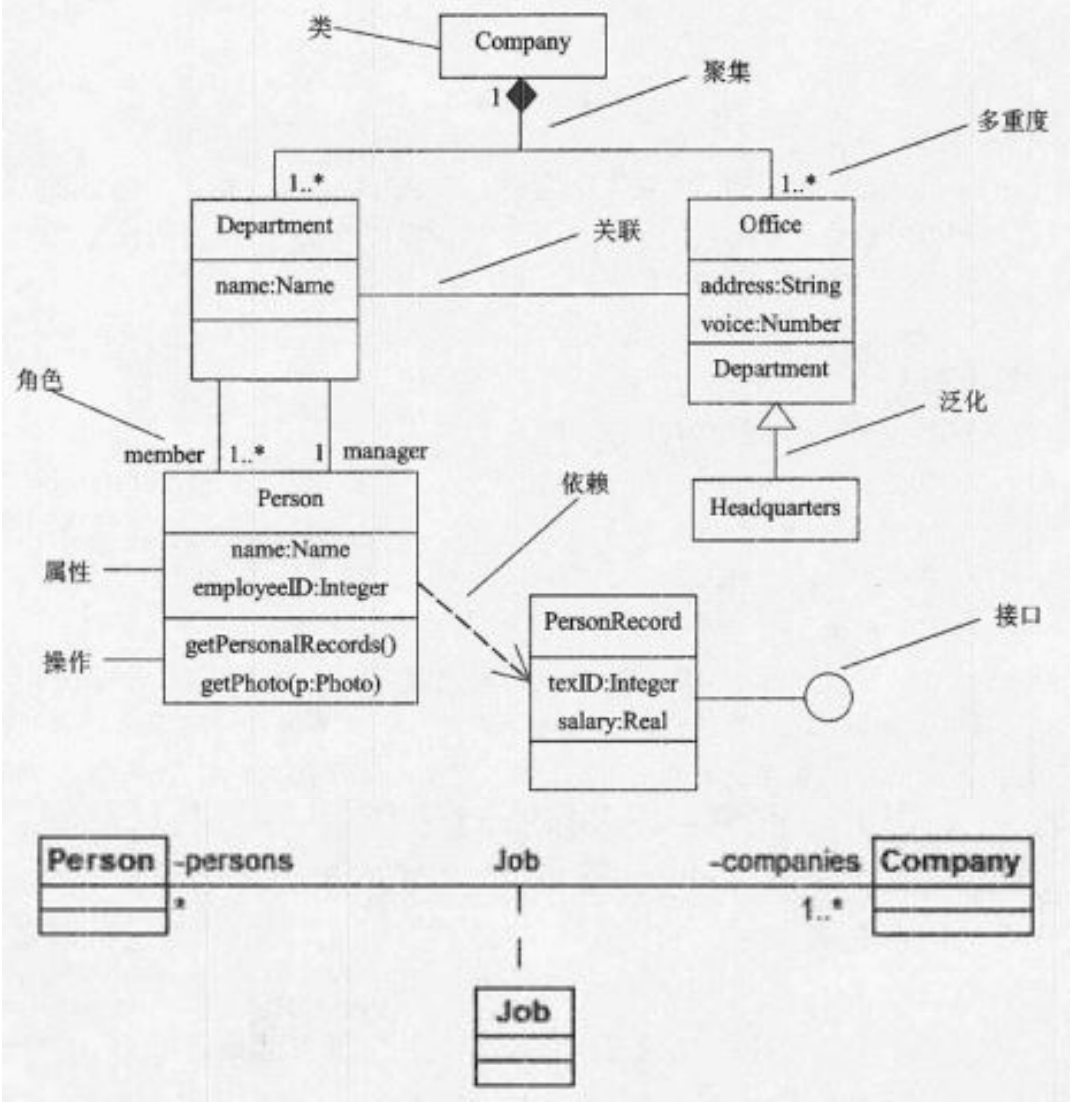
试题四十一 答案： C 解析：
本题考查面向对象和统一建模语言(UML)的基础知识。
状态图(statediagram)展现了一个状态机，关注系统的动态视图，强调对象行为的事件顺序引起的对象状态变化。
一般情况下，活动可以在状态转换时执行，也可以走状态内执行。检测到事件可能导致对象从一个状态移动到另一个状态，这样的移动即为转换，即事件触发转换，
这样能引起转换的事件称为触发器。事件发生时，检查监护条件，如果满足相应的事件，则进行相应的转换，如果都没满足，则此事件没有引起状态的改变。

试题四十二 答案： C 解析：
本题考查统一建模语言(UML)的基础知识。
一个类定义了一组大体上相似的对象。一个类所包含的方法和数据描述一组对象的共同行为和属性。把一组对象的共同特征加以抽象并存储在一个类中的能力，是面向对象技术最重要的一点。类图(classdiagram)展现了一组对象、接口、协作和它们之间的关系。在面向对象系统的建模中所建立的最常见的图就是类图。类图给出系统的静态设计视图。包含主动类的类图给出了系统的静态进程视图。
类图中通常包括类、接、口、协作、依赖、泛化和关联关系等内容(如下图所示)。类图中也可以包含注解和约束。类图还可以含有包或子系统，二者都用于把模型元素聚集成更大的组块。

在关联关系中，还可能拥有一些特性，构成类特性，即可看作是一个拥有关联特性的类，该关系兼具关联和类的特色。它定义了用于连接一些分类器，还定义属于关联关系本身的特性，这些特性只属亍关联关系本身。例如要建模员工(Person)和公司(Company)之间的工作关系，有一个重要的属性是工作岗位及其岗位工资。如果将岗位工资属性放在 Person 类和 Company 类都不合适，这一属性应该放在关联关系上，这样就需要建模一个关联类 Jo

b，用来设置岗位和岗位工资。

本题叙述中，一名医生(Doctor)可以治疗多位病人(Patient)，一位病人可以由多名医生治疗，这样，医生类和病人类之间的关联关系的两端多重度均为多(*)。另外，一名医生可能多次治疗同一位病人，那么，要记录哪名医生治疗哪位病人时，需要存储治疗(Treatment)的日期和时间。这一治疗日期和时间属性放在医生类和病人类都不合适，所以这一属性应该放在关联关系上，构成关联类治疗(Treatment)，并且一名医生可以多次治疗同，位病人，所以医生和病人组合并不唯一(non-unique)。



试题四十三 答案： A **解析：**

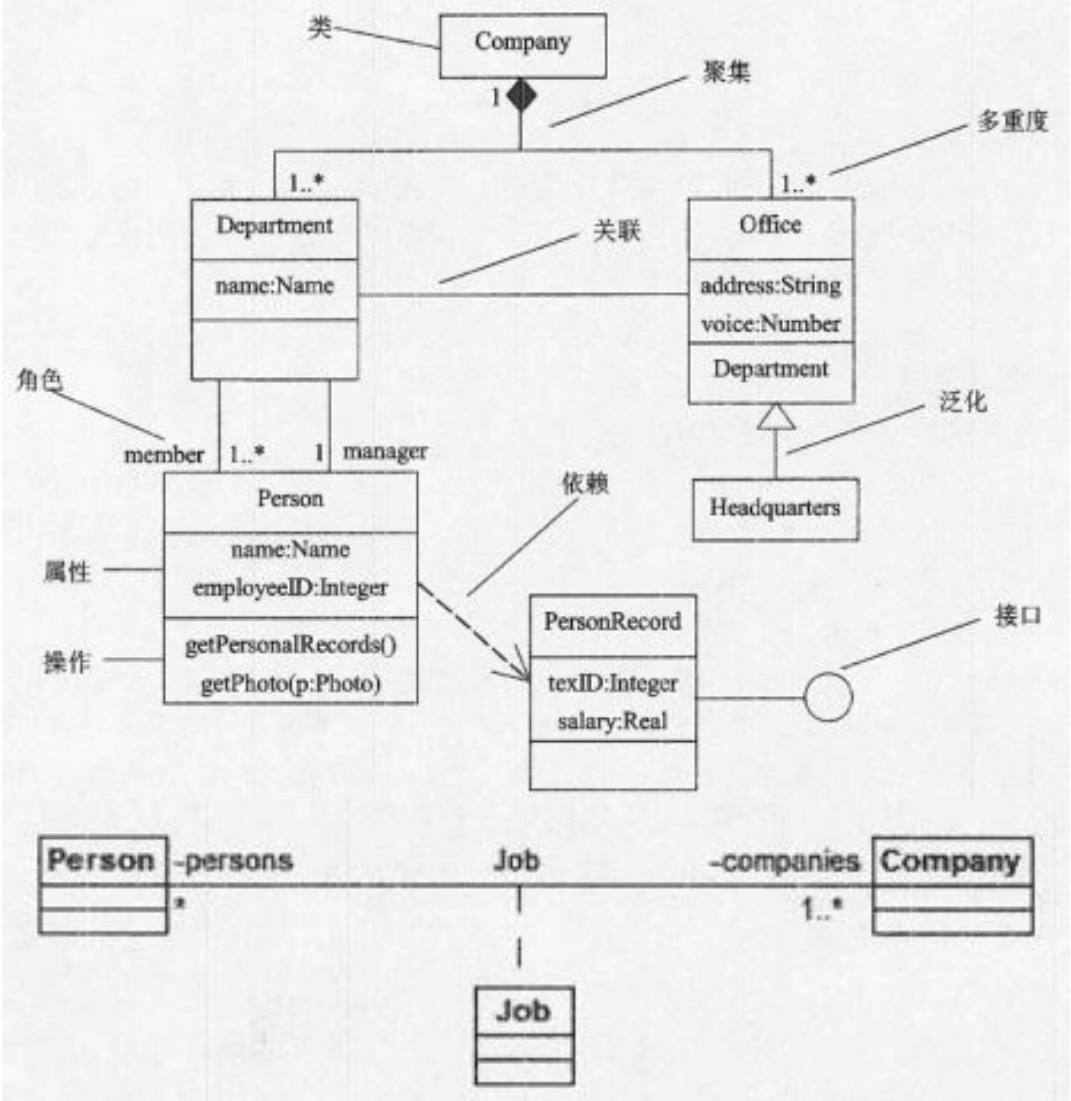
本题考查统一建模语言(UML)的基础知识。

一个类定义了一组大体上相似的对象。一个类所包含的方法和数据描述一组对象的共同行

为和属性。把一组对象的共同特征加以抽象并存储在一个类中的能力，是面向对象技术最重要的一点。类图(classdiagram)展现了一组对象、接口、协作和它们之间的关系。在面向对象系统的建模中所建立的最常见的图就是类图。类图给出系统的静态设计视图。包含主动类的类图给出了系统的静态进程视图。
类图中通常包括类、接、口、协作、依赖、泛化和关联关系等内容(如下图所示)。类图中也可以包含注解和约束。类图还可以含有包或子系统，二者都用于把模型元素聚集成更大的组块。

在关联关系中，还可能拥有一些特性，构成类特性，即可看作是一个拥有关联特性的类，该关系兼具关联和类的特色。它定义了用于连接一些分类器，还定义属于关联关系本身的特性，这些特性只属亍关联关系本身。例如要建模员工(Person)和公司(Company)之间的工作关系，有一个重要的属性是工作岗位及其岗位工资。如果将岗位工资属性放在 Person 类和 Company 类都不合适，这一属性应该放在关联关系上，这样就需要建模一个关联类 Job，用来设置岗位和岗位工资。

本题叙述中，一名医生(Doctor)可以治疗多位病人(Patient)，一位病人可以由多名医生治疗，这样，医生类和病人类之间的关联关系的两端多童度均为多(*)。另外，一名医生可能多次治疗同一位病人，那么，要记录哪名医生治疗哪位病人时，霈要存储治疗(Treatment)的日期和时间。这一治疗日期和时间属性放在医生类和病人类都不合适，所以这一属性应该放在关联关系上，构成关联类治疗(Treatment)，并且一名医生可以多次治疗同，位病人，所以医生和病人组合并不唯一(non-unique)。



试题四十四 答案： D **解析：**

本题考查设计模式的基本概念。每种设计模式都有特定的意图和适用情况。

命令(Command)将一个请求封装为一个对象，从而使得可以用不同的请求对客户进行参数化；对请求排队或记录请求日志，以及支持可撤销的操作。命令模式适用于以下几种情况：

①抽象出待执行的动作以参数化某对象，此模式是过程语言中的回调(callback)机制的一个面向对象的替代方式；

②在不同的时刻指定、排列和执行请求；

③支持取消操作；

④支持修改日志，这样当系统崩溃时，这些修改可以被重做一遍；

⑤用构建在原语操作上的高层操作构造一个系统。

责任链(ChainofResponsibility)使多个对象都有机会处理请求，从而避免请求的发送者和接收者之间的耦合关系。将这些对象连成一条链，并沿着这条链传递该请求，直到有一个对象处理它为止。责任链模式适用于以下几种情况：

①有多个的对象可以处理一个请求，哪个对象处理该请求在运行时刻自动确定；

②在不明确指定接收者的情况下，向多个对象中的一个提交一个请求；

③可处理一个请求的对象集合应被动态指定。

观察者(Observer)模式定义对象间的一种一对多的依赖关系，当一个对象的状态发生改变时，所有依赖于它的对象都得到通知并被自动更新。观察者适用于以下几种情况：

①当一个抽象模型有两个方面，其中一个方面依赖于另一个方面，将这两者封装在独立地对象中以使它们可以各自独立地改变和复用；

②当对一个对象的改变需要同时改变其他对象，而不知道具体有多少对象有待改变时；

③当一个对象必须通知其他对象，它又不能假定其他对象是谁，即：不希望这些对象是紧耦合的。

策略(Strategy)定义一系列的算法，把它们一个个封装起来，并且使它们可以相互替换。此模式使得算法可以独立于使用它们的客户而变化 a 策略模式适用于以下几种情况：

①许多相关的类仅仅是行为有异。“策略”提供了一种用多个行为中的一个行为来配置一个类的方法；

②需要使用一个算法的不同变体。例如，定义一些反应不同空间的空间/时间权衡的算法。当这些变体实现为一个算法的类层次时，可以使用策略模式；

③算法使用客户不应该知道的数据。可使用策略模式以避免暴露复杂的、与算法相关的数据结构；

④一个类定义了多种行为，并且这些行为在这个类的操作中以多个条件语句的形式出现，将相关的条件分支移入它们各自的 Strategy 类中，以代替这些条件语句。

试题四十五 答案： C **解析：**

本题考查设计模式的基本概念。每种设计模式都有特定的意图和适用情况。

命令(Command)将一个请求封装为一个对象，从而使得可以用不同的请求对客户进行参数化；对请求排队或记录请求日志，以及支持可撤销的操作。命令模式适用于以下几种情况：

①抽象出待执行的动作以参数化某对象，此模式是过程语言中的回调(callback)机制的一个面向对象的替代方式；

②在不同的时刻指定、排列和执行请求；

③支持取消操作；

④支持修改日志，这样当系统崩溃时，这些修改可以被重做一遍；
⑤用构建在原语操作上的高层操作构造一个系统。
责任链(ChainofResponsibility)使多个对象都有机会处理请求，从而避免请求的发送者和接收者之间的耦合关系。将这些对象连成一条链，并沿着这条链传递该请求，直到有一个对象处理它为止。责任链模式适用于以下几种情况：
①有多个的对象可以处理一个请求，哪个对象处理该请求在运行时刻自动确定；
②在不明确指定接收者的情况下，向多个对象中的一个提交一个请求；
③可处理一个请求的对象集合应被动态指定。
观察者(Observer)模式定义对象间的一种一对多的依赖关系，当一个对象的状态发生改变时，所有依赖于它的对象都得到通知并被自动更新。观察者适用于以下几种情况：
①当一个抽象模型有两个方面，其中一个方面依赖于另一个方面，将这两者封装在独立地对象中以使它们可以各自独立地改变和复用；
②当对一个对象的改变需要同时改变其他对象，而不知道具体有多少对象有待改变时；
③当一个对象必须通知其他对象，它又不能假定其他对象是谁，即：不希望这些对象是紧耦合的。
策略(Strategy)定义一系列的算法，把它们一个个封装起来，并且使它们可以相互替换。此模式使得算法可以独立于使用它们的客户而变化 a 策略模式适用于以下几种情况：
①许多相关的类仅仅是行为有异。“策略”提供了一种用多个行为中的一个行为来配置一个类的方法；
②需要使用一个算法的不同变体。例如，定义一些反应不同空间的空间/时间权衡的算法。当这些变体实现为一个算法的类层次时，可以使用策略模式；
③算法使用客户不座该知道的数据。可使用策略模式以避免暴露复杂的、与算法相关的数据结构；
④一个类定义了多种行为，并且这些行为在这个类的操作中以多个条件语句的形式出现，将相关的条件分支移入它们各自的 Strategy 类中，以代替这些条件语旬。

试题四十六 答案： A **解析：**
本题考查设计模式的基本概念。每种设计模式都集中于一个特定的面向对象设计问题或设计要点，有特定的意图和适用情况。
生成器(Builder)模式将一个复杂对象的构建与它的表示分离，使得同样的构建过程可以创建不同的表示。生成器模式适用于以下几种情况：
①当创建复杂对象的算法应该独立于该对象的组成部分以及它们的装配方式时；
②当构造过程必须允许被构造的对象有不同的表示时。

工厂方法(FactoryMethod)定义一个用于创建对象的接口，让子类决定将哪一个类实例化，使一个类的实例化延迟到其子类。工厂方法适用于以下几种情况：

①当一个类不知道它所必须创建的对象的类的时候；

②当一个类希望由它的子类来指定它所创建的对象的时候；

③当类将创建对象的职责委托给多个帮助子类中的某一个，并且你希望将哪一个帮助子类是代理者这一信息局部化的时候。

原型(Prototype)模式用原型实例指定创建对象的种类，并且通过拷贝这个原型来创建新的对象。原型模式适用于以下几种情况：

①当一个系统应该独立于它的产品创建、构成和表示时；

②当要实例化的类是在运行时刻指定时，例如，通过动态装载；

③为了避免创建一个与产品类层次平行的工厂类层次时；

④当一个类的实例只能有几个不同状态组合中的一种时，建立相应数目的原型并克隆它们可能比每次用合适的状态手工实例化该类更方便一些。

单例(Singleton)设计模式是一种创建型模式，其意图是保证一个类仅有一个实例，并提供一个访问这个唯一实例的全局访问点。单例模式适用于以下情况：

①当类只能有一个实例而且客户可以从一个众所周知的访问点访问它时；

②当这个唯一实例应该是通过子类化可扩展的，并且客户应该无须更改代码就能使用一个扩展的实例时。

试题四十七 答案： A **解析：**

本题考查设计模式的基本概念。每种设计模式都集中于一个特定的面向对象设计问题或设计要点，有特定的意图和适用情况。

生成器(Builder)模式将_个复杂对象的构建与它的表示分离，使得同样的构建过程可以创建不同的表示。生成器模式适用于以下几种情况：

①当创建复杂对象的算法应该独立于该对象的组成部分以及它们的装配方式时；

②当构造过程必须允许被构造的对象有不同的表示时。

工厂方法(FactoryMethod)定义一个用于创建对象的接口，让子类决定将哪一个类实例化，使一个类的实例化延迟到其子类。工厂方法适用于以下几种情况：

①当一个类不知道它所必须创建的对象的类的时候；

②当一个类希望由它的子类来指定它所创建的对象的时候；

③当类将创建对象的职责委托给多个帮助子类中的某一个，并且你希望将哪一个帮助子类是代理者这一信息局部化的时候。

原型(Prototype)模式用原型实例指定创建对象的种类，并且通过拷贝这个原型来创建新的

对象。原型模式适用于以下几种情况：

①当一个系统应该独立于它的产品创建、构成和表示时；

②当要实例化的类是在运行时刻指定时，例如，通过动态装载；

③为了避免创建一个与产品类层次平行的工厂类层次时；

④当一个类的实例只能有几个不同状态组合中的一种时，建立相应数目的原型并克隆它们可能比每次用合适的状态手工实例化该类更方便一些。

单例(Singleton)设计模式是一种创建型模式，其意图是保证一个类仅有一个实例，并提供一个访问这个唯一实例的全局访问点。单例模式适用于以下情况：

①当类只能有一个实例而且客户可以从一个众所周知的访问点访问它时；

②当这个唯一实例应该是通过子类化可扩展的，并且客户应该无须更改代码就能使用一个扩展的实例时。

试题四十八 答案： A 解析：

本题考查程序语言知识。

正规式(b|ab)表示的正规集为{b， ab}，(b|ab)*表示的正规集为{ε ， b， ab， bb， bab， abb， abab， bbb， bbab， babb， babab， abbb， abbab， ababb， ababab，...}，用自然语言描述就是每个 a 后面都至少有 1 个 b 。

正规式(ab*)表示的正规集为{ε ， a， ab， abb， abbb， abbbb，…}，(ab*)*表示的正规集为{aa， aab， aabb， aabbb， aabbbb， aba， abba， abbba， abab， abbab，...}，用自然语言描述就是除了空串，每个串中都至少有 1 个 a 。

正规式(a*b*)*和(a|b)*是等价的，它们都表示{ε ， a， b， aa， ab， ba， bb， am，aab， aba， abb， baa， bab， bab， bbb，，用自然语言描述就是用 a 、 b 构成的任何字符串。

试题四十九 答案： B 解析：

本题考查程序语言知识。

程序语言的大多数语法现象可用乔姆斯基的上下文无关文法描述。

试题五十 答案： D 解析：

本题考查程序语言知识。

代码段中“for(； k<100 ；)；”的循环体为空语句，循环条件中的 k 值在循环中没有改

变，因此“k<100”是一直成立的，此代码段是无限循环的，只有运行时才能表现出来，属于动态语义错误。

试题五十一 答案： B 解析：

本题考查数据库安全控制方面的基础知识。

数据库管理系统的安全措施有 3 个方面：

①权限机制：通过权限机制，限定用户对数据的操作权限，把数据的操作限定在具有指定权限的用户范围内，以保证数据的安全。在标准 SQL 中定义了授权语句 GRANT 来实现权限管理。

②视图机制：通过建立用户视图，用户或应用程序只能通过视图来操作数据，保证了视图之外的数据的安全性。

③数据加密：对数据库中的数据进行加密，可以防止数据在存储和传输过程中失密。

试题五十二 答案： B 解析：

本题考查关系数据库规范化理论方面的基础知识。

根据题意， $F=\{A_1\rightarrow A_2,\ A_1A_2\rightarrow A_3,\ A_1\rightarrow A_4,\ A_2\rightarrow A_4\},\}$，不难得出属性 A_1A_2 决定全属性 U，所以 A_1A_2 为候选关键字。由于 $A_1\rightarrow A_2$， $A_2\rightarrow A_4$ 可以推出 A_1A_4(传递率)，所以函数依赖集 $A_1\rightarrow A_4$ 是冗余的。

试题五十三 答案： C 解析：

本题考查关系数据库规范化理论方面的基础知识。

根据题意， $F=\{A_1\rightarrow A_2,\ A_1A_2\rightarrow A_3,\ A_1\rightarrow A_4,\ A_2\rightarrow A_4\},\}$，不难得出属性 A1A2 决定全属性 U，所以 A_1A_2 为候选关键字。由于 $A_1\rightarrow A_2$， $A_2\rightarrow A_4$ 可以推出 A_1A_4(传递率)，所以函数依赖集 $A_1\rightarrow A_4$ 是冗余的。

试题五十四 答案： C 解析：

本题考查关系代数运算方面的基础知识。

自然连接是一种特殊的等值连接，它要性列>第 7 个属性列(即 R.B>S.E)，同时满足第 3 个属性列=第 6 个属性列(即 R.C=S.C)。

选取运算 $\sigma_{1=5\wedge 2>'7'\wedge 3=6}(R\times S)$ 中的条件“2>‘7’”与题意不符，其含义是 R.B 的值大于 7(属性列数字 7 加了单引号表示数值 7)，而不是 R.B>S.E 。求两个关系中进行比较的分量

必须是相同的属性组，并且在结果集中去掉右边重复的属性列。对关系 R 和 S 进行自然连接运算后的属性列数为 6 个，即为 R.A， R.B， R.C， R.D， S.E， S.F 。

试题五十五 答案： B 解析：

本题考查关系代数运算方面的基础知识。

自然连接是一种特殊的等值连接，它要性列>第 7 个属性列(即 R.B>S.E)，同时满足第 3 个属性列=第 6 个属性列(即 R.C=S.C)。

选取运算 $\sigma_{1=5\wedge 2>'7'\wedge 3=6}(RxS)$ 中的条件“2>‘7’”与题意不符，其含义是 R.B 的值大于 7(属性列数字 7 加了单引号表示数值 7)，而不是 R.B>S.E 。求两个关系中进行比较的分量必须是相同的属性组，并且在结果集中去掉右边重复的属性列。对关系 R 和 S 进行自然连接运算后的属性列数为 6 个，即为 R.A， R.B， R.C， R.D， S.E， S.F 。

对于试题(55)， R×S 的结果集的属性列为 R.A， R.B， R.C， R.D， S.A， S.C， S.E， S.F，选取运算 σ 是对关系进行横向运算，没有去掉重复属性列。选项 B“$\pi_{1，2，3，4，7，8}(\sigma_{1=5\wedge 2>7\wedge 3=6}(RxS))$”的含义为 R 与 S 的笛卡儿积中选择第 1 个属性列=第 5 个属性列(即 R.A=S.A)，同时满足第 2 个属性

试题五十六 答案： D 解析：

本题考查关系代数运算方面的基础知识。

关系代数表达式查询优化的原则如下：

①提早执行选取运算。对于有选择运算的表达式，应优化成尽可能先执行选择运算的等价表达式，以得到较小的中间结果，减少运算量以及从外存读块的次数。

②合并乘积与其后的选择堤算为连接运算。在表达式中，当乘积运算后面是选择运算时，应该合并为连接运算，使选择与乘积一道完成，以避免做完乘积后，需再扫描个大的乘积关系进行选择运算。

③将投影运算与其后的其他运算同时进行，以避免重复扫描关系。

④将投影运算和其前后的二目运算结合起来，使得没有必要为去掉某些字段再扫描一遍关系。

⑤在执行连接前对关系适当地预处理，就能快速地找到要连接的元组。方法有两种：索引连接法、排序合并连接法。

⑥存储公共子表达式。对于有公共子表达式的结果应存于外存(中间结果)，这样，当从外存读出它的时闻比计算的时间少时，就可节约操作时间。

显然，根据原则①尽量提早执行选取运算。

试题五十七 答案： A **解析：**

本题考查数据结构基础知识。

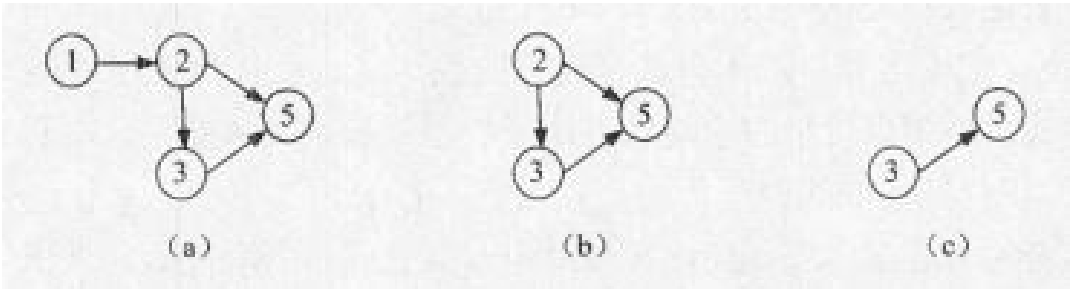
对有向无环图网进行拓扑排序的方法如下：

①在 AOV 网中选择一个入度为零(没有前驱)的顶点 v 且输出它；

②从网中删除该顶点 v 以及与该顶点有关的所有边；

③重复上述两步，直至网中不存在入度为零的顶点为止。

按照上述方法，拓扑序列的第一个顶点为 4，执行①和②步之后的有向图如下图(a)所示。接下来再输出的顶点只能为 1，因此执行①和②步之后的有向图如下图(b)所示。接下来再输出的顶点只能为 2，因此①和②步之后的有向图如下图(c)所示。因此，拓扑序列为 41235。



试题五十八 答案： B **解析：**

本题考查数据结构基础知识。

线性表是一个线性序列，在顺序存储方式下，若删除其中一个元素，需要将其后的元素逐个前移，使得元素之间没有空闲单元。表长为 n 时，共有 n 个可删除的元素，删除元素 a_1 时需要移动 n-1 个元素，删除元素 a_n 时不需要移动元素，因此，等概率下删除一个元素时平均的移动元素次数 E_{delete} 为

线性表若采用单链表存储，插入和删除元素的实质都是对相关指针的修改，而不需要移动元素。

$$E_{delete}=\sum_{i=1}^{n}q_i\times(n-i)=\frac{1}{n}\sum_{i=1}^{n}(n-i)=\frac{n-1}{2}$$

试题五十九 答案： A **解析：**

本题考查数据结构基础知识。

线性表是一个线性序列，在顺序存储方式下，若删除其中一个元素，需要将其后的元素逐

个前移，使得元素之间没有空闲单元。表长为 n 时，共有 n 个可删除的元素，删除元素 a_1 时需要移动 n-1 个元素，删除元素 a_n 时不需要移动元素，因此，等概率下删除一个元素时平均的移动元素次数 E_{delete} 为

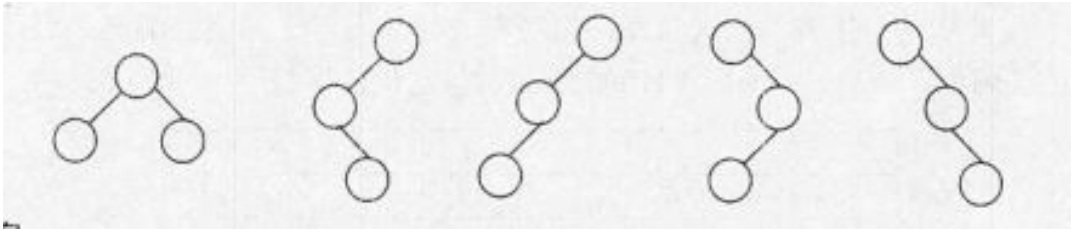
线性表若采用单链表存储，插入和删除元素的实质都是对相关指针的修改，而不需要移动元素。

$$E_{delete}=\sum_{i=1}^{n}q_i\times(n-i)=\frac{1}{n}\sum_{i=1}^{n}(n-i)=\frac{n-1}{2}$$

试题六十 答案： C 解析：

本题考查数据结构基础知识。

具有 3 个结点的二叉树有以下 5 种形态，如下图所示。



试题六十一 答案： D 解析：

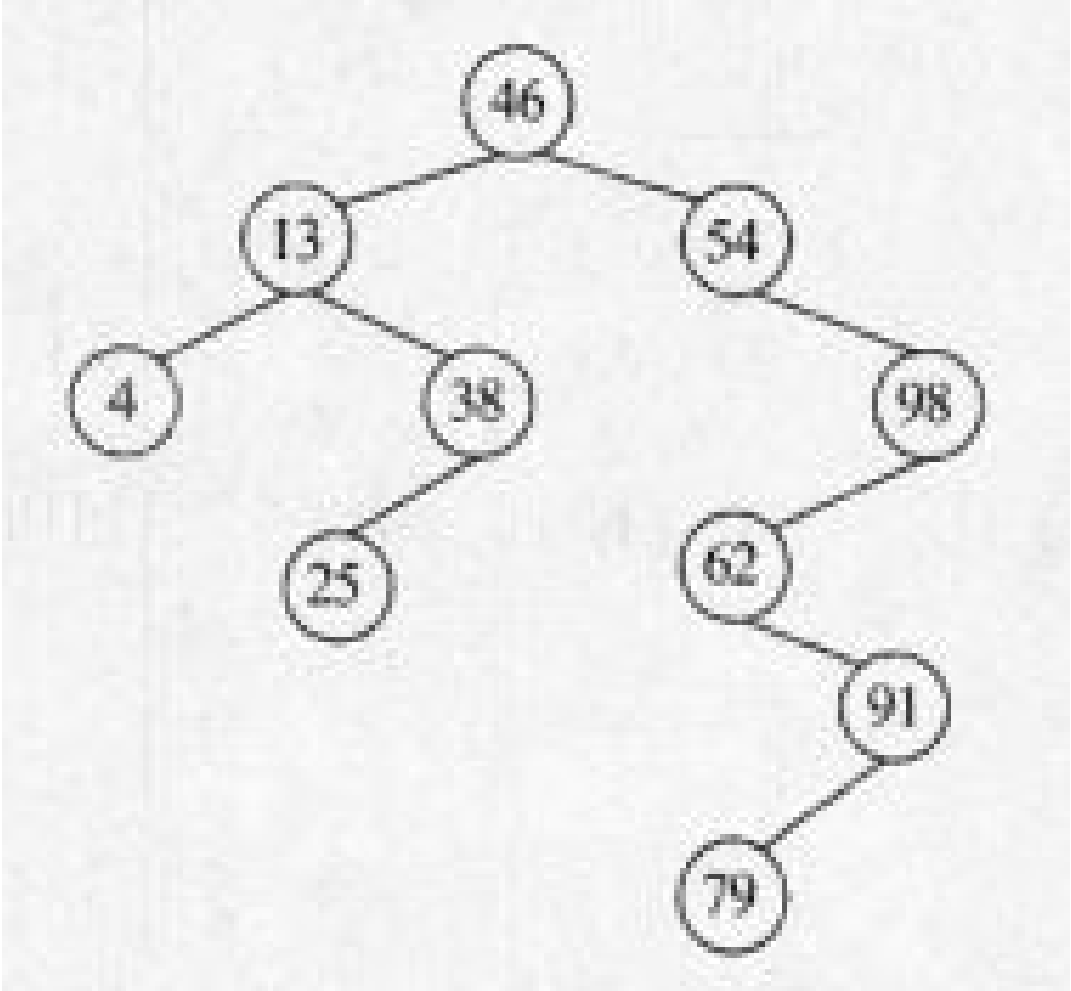
本题考查数据结构基础知识。

二叉查找树又称为 p 叉排序树或二叉检索树，它或者是一棵空树，或者是具有如下性质的二叉树：①若它的左子树非空，则左子树中所有结点的值均小于根结点的值；②若它的右子树非空，则右子树中所有结点的值均大于根结点的值；③左、右子树本身就是叉查找树。某二叉排序树如下图所示。

以上图为例，对非空二叉排序树进行中序遍历，得到递增有序的序列，先序和后序序列则不是。

二叉排序树中结点在左、右子树上的分布并不均匀，极端情况下， n 个结点的二叉排序树的高度为 n 。

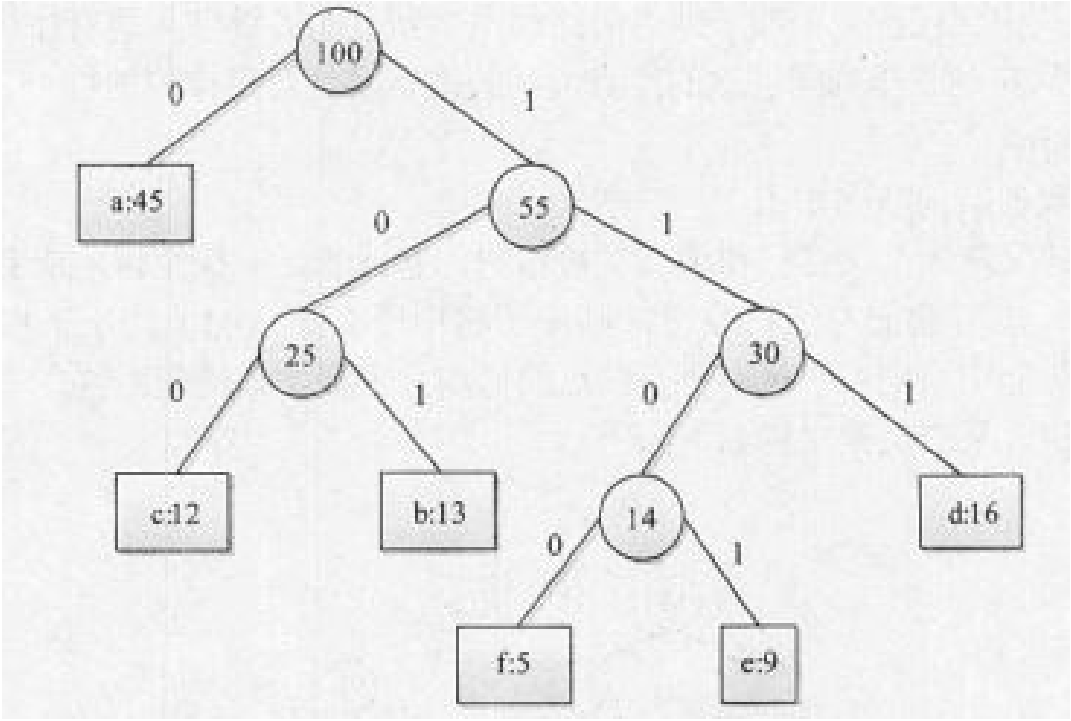
以上图为例，从 46 到 25 的路径上的结点关键码序列为 46， 13， 38， 25，并不是一个有序序列。



试题六十二 答案： A **解析：**

本题考查算法设计与分析的基础知识。题干中给出的实例的霍夫曼编码树如下图所示。

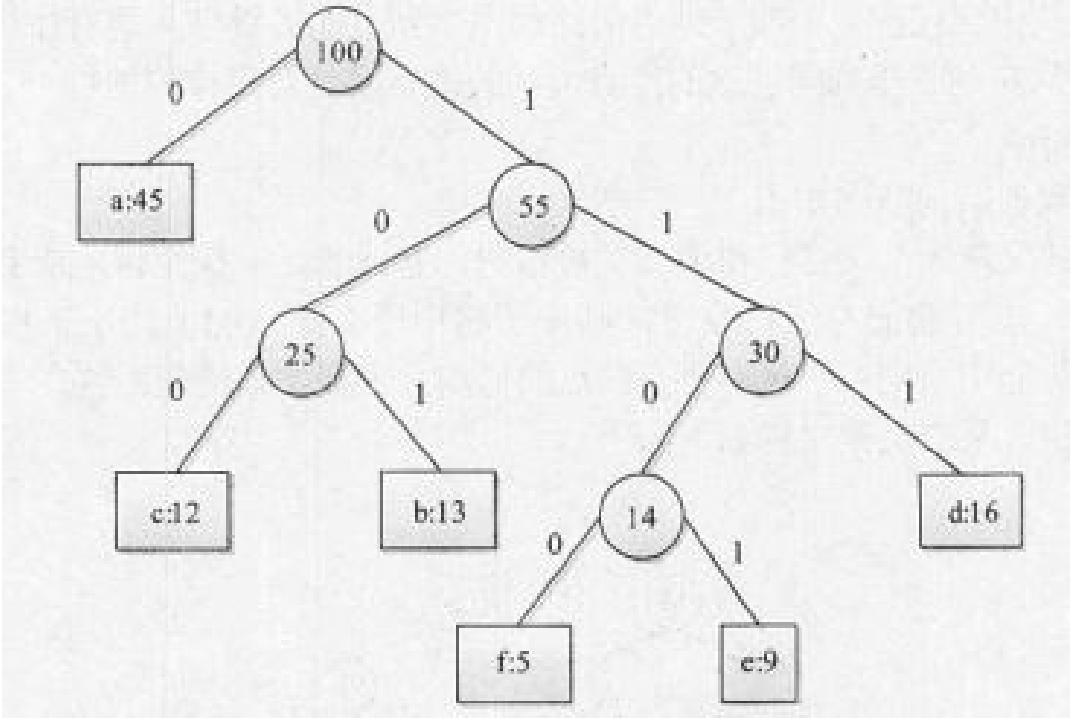
根据该图， bee 的编码为 10111011101 。而编码“110001001101”对应的字符序列则为 face 。



试题六十三 答案： C **解析：**

本题考查算法设计与分析的基础知识。题干中给出的实例的霍夫曼编码树如下图所示。

根据该图， bee 的编码为 10111011101 。而编码“110001001101”对应的字符序列则为 face 。

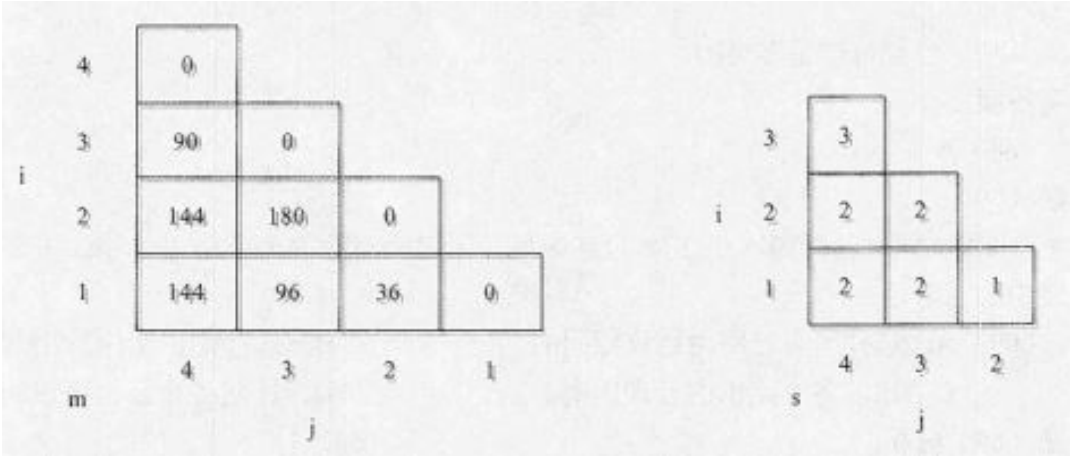


试题六十四 答案：　C 解析：

本题考查算法设计与分析的基础知识。

矩阵链乘是一个最优化问题，求解 n 个矩阵相乘的最优加括号方式，可以用动态规划方法来求解。题目已经给出动态规划求解的递归式。根据上式计算 m 的值，同时记录 k 的值到 s 中。

可以得到最优的加括号方式((M1M2)(M3M4))，乘法次数为 144 。
而根据该递归式自底向上求解时，应该用三重循环进行，即矩阵链长度 i 从 1 到 n-1+1 ，矩阵链起始位置，即 i 从 1 到 n-1+1 ，矩阵链分开的位置 k ，从 i 到 j-1 。因此时间复杂度为 0(nsup3)。

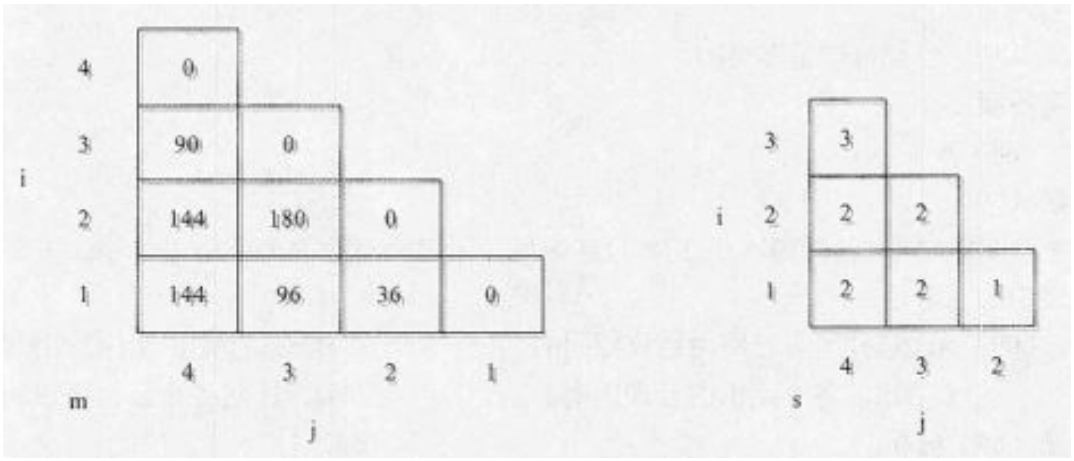


试题六十五 答案：　B 解析：

本题考查算法设计与分析的基础知识。

矩阵链乘是一个最优化问题，求解 n 个矩阵相乘的最优加括号方式，可以用动态规划方法来求解。题目已经给出动态规划求解的递归式。根据上式计算 m 的值，同时记录 k 的值到 s 中。

可以得到最优的加括号方式((M1M2)(M3M4))，乘法次数为 144 。
而根据该递归式自底向上求解时，应该用三重循环进行，即矩阵链长度 i 从 1 到 n-1+1 ，矩阵链起始位置，即 i 从 1 到 n-1+1 ，矩阵链分开的位置 k ，从 i 到 j-1 。因此时间复杂度为 0(nsup3)。



试题六十六 答案： A **解析：**

属于应用层协议的是简单网络管理协议 SNMP，它的传输层协议是 UDP。ARP 和 ICMP 都属于网络层协议。X.25 是分组交换网上的协议，也归于网络层。

试题六十七 答案： C **解析：**

属于应用层协议的是简单网络管理协议 SNMP，它的传输层协议是 UDP。ARP 和 ICMP 都属于网络层协议。X.25 是分组交换网上的协议，也归于网络层。

试题六十八 答案： A **解析：**

本题考查 URL 的基础知识。

URL(UniformResourceLocator，统一资源定位符)是对互联网的资源位置和访问方法的一种简洁的表示，是互联网上资源的地址。互联网上的每个文件都有一个唯一的 URL，它包含的信息指出文件的位置以及浏览器应该怎么处理它。

一个标准 URL 的格式如下：

协议：//主机名.域名.域名后缀或 IP 地址(：端口号)/目录/文件名

其中，目录可能是多级的。

试题六十九 答案： C **解析：**

各种路由来源的管理距离如下表所示。

如果路由器收到了由多个路由协议转发的、关于某个目标的多条路由，则比较各个路由的管理距离，并采用管理距离小的路由来源提供的路由信息。

路由来源	管理距离	路由来源	管理距离
直连路由	0	IS-IS	115
静态路由	1	RIP	120
EIGRP 汇总路由	5	EGP	140
外部 BGP	20	ODR（按需路由）	160
内部 EIGRP	90	外部 EIGRP	170
IGRP	100	内部 BGP	200
OSPF	110	未知	255

试题七十 答案： D 解析：

地址 220.112.145.32/22 的二进制形式是 11011100.01110000.10010001. 00100000

地址 220.112.145.64/22 的二进制形式是 11011100.01110000.10010001. 01000000

地址 220.112.147.64/22 的二进制形式是 11011100.01110000.10010011. 01000000

地址 220.112.177.64/22 的二进制形式是 11011100.01110000.10110001. 01000000

而地址 220.112.179.92 的二进制形式是 11011100.01110000.10110011.01011100

所以与地址 220.112.179.92 匹配的是 220.112.177.64/22 。

试题七十一 答案： C 解析：

规模上，软件实体可能比任何由人类创造的其他实体要复杂，因为没有任何两个软件部分是相同的(至少是在语句的级别)。如果有相同的情况，我们会把它们合并成供调用的子函数。在这个方面，软件系统与计算机、建筑或者汽车大不相同，后者往往存在着大量重复的部分。

数字计算机本身就比人类建造的大多数东西复杂。计算机拥有大量的状态，这使得构思、描述和测试都非常困难。软件系统的状态又比计算机系统状态多若干个数量级。

同样，软件实体的扩展也不仅仅是相同元素重复添加，而必须是不同元素实体的添加。大多数情况下，这些元素以非线性递增的方式交互，因此整个软件的复杂度以更大的非线性级数增长。

软件的复杂度是必要属性，不是次要因素。因此，抽掉复杂度的软件实体描述常常也去掉了一些本质属性。数学和物理学在过去三个世纪取得了巨大的进步，数学家和物理学家们建立模型以简化复杂的现象，从模型中抽取出各种特性，并通过试验来验证这些特性。这些方法之所以可行一一是因为模型中忽略的复杂度不是被研究现象的必要属性。当复杂度是本质特性时，这些方法就行不通了。

上述软件特有的复杂度问题造成了很多经典的软件产品开发问题。复杂度不仅仅导致技术上的困难，还引发了很多管理上的问题。

试题七十二 答案： A 解析：

规模上，软件实体可能比任何由人类创造的其他实体要复杂，因为没有任何两个软件部分是相同的(至少是在语句的级别)。如果有相同的情况，我们会把它们合并成供调用的子函数。在这个方面，软件系统与计算机、建筑或者汽车大不相同，后者往往存在着大量重复的部分。

数字计算机本身就比人类建造的大多数东西复杂。计算机拥有大量的状态，这使得构思、描述和测试都非常困难。软件系统的状态又比计算机系统状态多若干个数量级。

同样，软件实体的扩展也不仅仅是相同元素重复添加，而必须是不同元素实体的添加。大多数情况下，这些元素以非线性递增的方式交互，因此整个软件的复杂度以更大的非线性级数增长。

软件的复杂度是必要属性，不是次要因素。因此，抽掉复杂度的软件实体描述常常也去掉了一些本质属性。数学和物理学在过去三个世纪取得了巨大的进步，数学家和物理学家们建立模型以简化复杂的现象，从模型中抽取出各种特性，并通过试验来验证这些特性。这些方法之所以可行一一是因为模型中忽略的复杂度不是被研究现象的必要属性。当复杂度是本质特性时，这些方法就行不通了。

上述软件特有的复杂度问题造成了很多经典的软件产品开发问题。复杂度不仅仅导致技术上的困难，还引发了很多管理上的问题。

试题七十三 答案： B 解析：

规模上，软件实体可能比任何由人类创造的其他实体要复杂，因为没有任何两个软件部分是相同的(至少是在语句的级别)。如果有相同的情况，我们会把它们合并成供调用的子函数。在这个方面，软件系统与计算机、建筑或者汽车大不相同，后者往往存在着大量重复的部分。

数字计算机本身就比人类建造的大多数东西复杂。计算机拥有大量的状态，这使得构思、描述和测试都非常困难。软件系统的状态又比计算机系统状态多若干个数量级。

同样，软件实体的扩展也不仅仅是相同元素重复添加，而必须是不同元素实体的添加。大多数情况下，这些元素以非线性递增的方式交互，因此整个软件的复杂度以更大的非线性级数增长。

软件的复杂度是必要属性，不是次要因素。因此，抽掉复杂度的软件实体描述常常也去掉了一些本质属性。数学和物理学在过去三个世纪取得了巨大的进步，数学家和物理学家们建立模型以简化复杂的现象，从模型中抽取出各种特性，并通过试验来验证这些特性。这些方法之所以可行一一是因为模型中忽略的复杂度不是被研究现象的必要属性。当复杂度是本质特性时，这些方法就行不通了。

上述软件特有的复杂度问题造成了很多经典的软件产品开发问题。复杂度不仅仅导致技术上的困难，还引发了很多管理上的问题。

试题七十四 答案： D 解析：

规模上，软件实体可能比任何由人类创造的其他实体要复杂，因为没有任何两个软件部分是相同的(至少是在语句的级别)。如果有相同的情况，我们会把它们合并成供调用的子函数。在这个方面，软件系统与计算机、建筑或者汽车大不相同，后者往往存在着大量重复的部分。

数字计算机本身就比人类建造的大多数东西复杂。计算机拥有大量的状态，这使得构思、描述和测试都非常困难。软件系统的状态又比计算机系统状态多若干个数量级。

同样，软件实体的扩展也不仅仅是相同元素重复添加，而必须是不同元素实体的添加。大多数情况下，这些元素以非线性递增的方式交互，因此整个软件的复杂度以更大的非线性级数增长。

软件的复杂度是必要属性，不是次要因素。因此，抽掉复杂度的软件实体描述常常也去掉了一些本质属性。数学和物理学在过去三个世纪取得了巨大的进步，数学家和物理学家们建立模型以简化复杂的现象，从模型中抽取出各种特性，并通过试验来验证这些特性。这些方法之所以可行一一是因为模型中忽略的复杂度不是被研究现象的必要属性。当复杂度是本质特性时，这些方法就行不通了。

上述软件特有的复杂度问题造成了很多经典的软件产品开发问题。复杂度不仅仅导致技术上的困难，还引发了很多管理上的问题。

试题七十五 答案： C 解析：

规模上，软件实体可能比任何由人类创造的其他实体要复杂，因为没有任何两个软件部分是相同的(至少是在语句的级别)。如果有相同的情况，我们会把它们合并成供调用的子函数。在这个方面，软件系统与计算机、建筑或者汽车大不相同，后者往往存在着大量重复的部分。

数字计算机本身就比人类建造的大多数东西复杂。计算机拥有大量的状态，这使得构思、描述和测试都非常困难。软件系统的状态又比计算机系统状态多若干个数量级。

同样，软件实体的扩展也不仅仅是相同元素重复添加，而必须是不同元素实体的添加。大多数情况下，这些元素以非线性递增的方式交互，因此整个软件的复杂度以更大的非线性级数增长。

软件的复杂度是必要属性，不是次要因素。因此，抽掉复杂度的软件实体描述常常也去掉了一些本质属性。数学和物理学在过去三个世纪取得了巨大的进步，数学家和物理学家们

建立模型以简化复杂的现象，从模型中抽取出各种特性，并通过试验来验证这些特性。这些方法之所以可行一一是因为模型中忽略的复杂度不是被研究现象的必要属性。当复杂度是本质特性时，这些方法就行不通了。

上述软件特有的复杂度问题造成了很多经典的软件产品开发问题。复杂度不仅仅导致技术上的困难，还引发了很多管理上的问题。