位于 CPU 与主存之间的高速缓冲存储器（Cache)用于存放部分主存数据的拷贝， 主存地址与 Cache 地址之间的转换工作由（1）完成。

(1)A.硬件　　B.软件　　C.用户　　D.程序员

【答案】A

【解析】本题考查高速缓冲存储器（Cache）的工作特点。

提供“高速缓存”的目的是为了让数据存取的速度适应 CPU 的处理速度，其基于的原理是内存中“程序执行与数据访问的局域性行为”，即一定程序执行时间和空间内，被访问的代码集中于一部分。为了充分发挥高速缓存的作用，不仅依靠“暂存刚刚访问过的数据”，还要使用硬件实现的指令预测与数据预取技术,即尽可能把将要使用的数据预先从内存中取到高速缓存中。

一般而言，主存使用 DRAM 技术，而 Cache 使用昂贵但较快速的 SRAM 技术。目前微计算机上使用的 AMD 或 Intel 微处理器都在芯片内部集成了大小不等的数据 高速缓存和指令高速缓存，通称为 L1 高速缓存（LI Cache,即第一级片上高速缓冲存储器)；而比 L1 容量更大的 L2 高速缓存曾经被放在 CPU 外部（主板或者 CPU 接口卡上)， 但是现在已经成为 CPU 内部的标准组件;更昂贵的顶级家用和工作站 CPU 甚至会配备比 L2 高速缓存还要大的 L3 高速缓存。

内存单元按字节编址，地址 0000A000H～0000BFFFH 共有（2)个存储单元。

(2)A.8192K　　B.1024K　　C.13K　　D.8K

【答案】D

【解析】本题考查存储器的地址计算知识。

每个地址编号为一个存储单元（容量为 1 个字节)，地址区间 0000A000H～0000BFFFH 共有 1FFF+1 个地址编号（即 213)，1K=1024,因此该地址区间的存储单元数也就是 8K。

相联存储器按（3）访问。

(3)A.地址　　B.先入后出的方式　　C.内容　　D.先入先出的方式

【答案】C

【解析】本题考查相联存储器的概念。

相联存储器是一种按内容访问的存储器。其工作原理就是把数据或数据的某一部分作为关键字,将该关键字与存储器中的每一单元进行比较,找出存储器中所有与关键字相同的数据字。

相联存储器可用在高速缓冲存储器中，在虚拟存储器中用来作段表、页表或快表存储器，还用在数据库和知识库中。

若 CPU 要执行的指令为：MOV R1, #45（即将数值 45 传送到寄存器 R1 中），则该指令中采用的寻址方式为（4）。

(4)A.直接寻址和立即寻址　　B.寄存器寻址和立即寻址

C.相对寻址和直接寻址　　D.寄存器间接寻址和直接寻址

【答案】B

【解析】本题考查指令系统基础知识。

指令中的寻址方式就是如何对指令中的地址字段进行解释，以获得操作数的方法或 获得程序转移地址的方法。常用的寻址方式有：

•立即寻址。操作数就包含在指令中。

•直接寻址。操作数存放在内存单元中，指令中直接给出操作数所在存储单元的地址。

•寄存器寻址。操作数存放在某一寄存器中，指令中给出存放操作数的寄存器名。

•寄存器间接寻址。操作数存放在内存单元中，操作数所在存储单元的地址在某个寄存器中。

•间接寻址。指令中给出操作数地址的地址。

•相对寻址。指令地址码给出的是一个偏移量（可正可负），操作数地址等于本条 指令的地址加上该偏移量。

•变址寻址。操作数地址等于变址寄存器的内容加偏移量。

题目给出的指令中，R1 是寄存器，属于寄存器寻址方式，45 是立即数，属于立即寻址方式。

一条指令的执行过程可以分解为取指、分析和执行三步，在取指时间 t 取 ft=3Δt、分析时间 t 分析=2Δt、执行时间 t 执行=4Δt 的情况下，若按串行方式执行，则 10 条指令全部执行完需要（5） Δt。若按照流水方式执行，则执行完 10 条指令需要（6） Δt。

(5)A.40　　B.70　　C.90　　D.100

(6)A.20　　B.30　　C.40　　D.45

【答案】C　D

【解析】本题考查指令执行的流水化概念。

根据题目中给出的数据，每一条指令的执行过程需要 9Δt。在串行执行方式下，执行完一条指令后才开始执行下一条指令，10 条指令共耗时 90Δt。

(6) 若按照流水方式执行，则在 第 i+2 条指令处于执行阶段时就可以分析第 i+1 条指令，同时取第 i 条指令^由于指令的执行阶段所需时间最长（为 4Δt)，因此指令开始流水执行后，每 4At 将完成一条指令，所需时间为 3Δt +2Δt 十 4Δt +4ΔtX9=45Δt。

甲和乙要进行通信，甲对发送的消息附加了数字签名，乙收到该消息后利用（7）验证该消息的真实性。

(7)A.甲的公钥　　B.甲的私钥　　C.乙的公钥　　D.乙的私钥

【答案】A

【解析】本题考查数字签名的概念。

数字签名（Digital Signature)技术是不对称加密算法的典型应用：数据源发送方使用自己的私钥对数据校验和(或)其他与数据内容有关的变量进行加密处理，完成对数据的合法“签名”，数据接收方则利用对方的公钥来解读收到的“数字签名“，并将解读结果用于对数据完整性的检验，以确认签名的合法性。数字签名主要的功能是保证信息传输的完整性、发送者的身份认证、防止交易中的抵赖发生。

在 Windows 系统中，默认权限最低的用户组是（8）

(8)A.everyone　　B.administrators　　C.power users　　D.users

【答案】A

【解析】本题考查 Windows 用户权限方面的知识。

在以上 4 个选项中，用户组默认权限由高到低的顺序是 administrators—power users —users—everyone

IIS6.0 支持的身份验证安全机制有 4 种验证方法，其中安全级别最高的验证方法是(9)。

(9)A.匿名身份验证　　B.集成 Windows 身份验证

C.基本身份验证　　D.摘要式身份验证

【答案】B

【解析】本题考查 Windows IIS 服务中身份认证的基础知识^

Windows IIS 服务支持的身份认证方式有.NET Passport 身份验证、集成 Windows 身份验证、摘要式身份验证和基本身份验证。

•集成 Windows 身份验证：以 Kerberos 票证的形式通过网络向用户发送身份验证信息，并提

供较高的安全级别。Windows 集成身份验证使用 Kerberos 版本 5NTLM 身份验证。

•摘要式身份验证：将用户凭据作为 MD5 哈希或消息摘要在网络中进行传输，这样就无法根据哈希对原始用户名和密码进行解码。

• .NET Passport 身份验证：对 IIS 的请求必须在查询字符串或 Cookie 中包含有效的.NET Passport 凭据，提供了单一登录安全性，为用户提供对 Internet 上各种服务的访问权限。

•基本身份验证：用户凭据以明文形式在网络中发送。这种形式提供的安全级别很低，因为几乎所有协议分析程序都能读取密码。

软件著作权的客体不包括（10）。

(10)A.源程序　　B.目标程序　　C.软件文档　　D.软件开发思想

【答案】D

【解析】

软件著作权的客体是指著作权法保护的计算机软件，包括计算机程序及其相关文档。

计算机程序通常包括源程序和目标程序。

源程序（又称为源代码、源码）是采用计算机程序设计语言（如 C、Java 语言）编写的程序，需要转换成机器能直接识别和执行的形式才能在计算机上运行并得出结果。 它具有可操作性、间接应用性和技术性等特点。

目标程序以二进制编码形式表示，是计算机或具有信息处理能力的装置能够识别和执行的指令序列，能够直接指挥和控制计算机的各部件（如存储器、处理器、I/O 设备等） 执行各项操作，从而实现一定的功能。它具有不可读性、不可修改性和面向机器性等特点。

源程序与目标程序就其逻辑功能而言不仅内容相同，而且表现形式相似，二者可以互相转换，最终结果一致。源程序是目标程序产生的基础和前提，目标程序是源程序编 译的必然结果；源程序和目标程序具有独立的表现形式，但是目标程序的修改通常依赖于源程序。同一程序的源程序文本和目标程序文本应当视为同一程序。无论是用源程序形式还是目标程序形式体现，都可能得到著作权法保护。

计算机软件包含了计算机程序，并且不局限于计算机程序，还包括与之相关的程序描述和辅助资料。我国将计算机程序文档（软件文档）视为计算机软件的一个组成部分。 计算机程序文档与计算机程序不同，计算机程序是用编程语言，如汇编语言、C 语言、 Java 语言等编写而成，而计算机程序文档是由自然语言或由形式语言编写而成的。计算机程序文档是指用自然语言或者形式化语言所编写的文字资料和图表，用来描述程序的内容、组成、设计、

功能、开发情况、测试结果及使用方法等。计算机程序文档一般以程序设计说明书、流程图、数据流图和用户手册等表现。

我国《计算机软件保护条例》第6条规定：“本条例对软件著作权的保护不延及开发软件所用的思想、处理过程、操作方法或者数学概念等。”也就是说，软件开发的思想、 处理过程、操作方法或者数学概念等与计算机软件分别属于主客观两个范畴。思想是开发软件的设计方案、构思技巧和功能，设计程序所实现的处理过程、操作方法、算法等， 表现是完成某项功能的程序。

我国著作权法只保护作品的表达，不保护作品的思想、原理、概念、方法、公式、 算法等，因此对计算机软件来说，只有程序的作品性能得到著作权法的保护，而体现其工具性的程序构思、程序技巧等却无法得到保护。实际上计算机程序的技术设计，如软件开发中对软件功能、结构的构思，往往是比程序代码更重要的技术成果，通常体现了软件开发中的主要创造性贡献。

中国企业M与美国公司L进行技术合作，合同约定M使用一项在有效期内的美国专利，但该项美国专利未在中国和其他国家提出申请。对于M销售依照该专利生产的产品，以下叙述正确的是（11）

(11)A.在中国销售，M需要向L支付专利许可使用费

B.返销美国，M不需要向L支付专利许可使用费

C.在其他国家销售，M需要向L支付专利许可使用费

D.在中国销售，M不需要向L支付专利许可使用费

【答案】D

【解析】本题考查知识产权知识，涉及专利权的相关概念。

知识产权受地域限制，只有在一定地域内知识产权才具有独占性。也就是说，各国依照其本国法律授予的知识产权，只 能在其本国领域内受其法律保护，而其他国家对这种权利没有保护的义务，任何人均可在自己的国家内自由使用外国人的知识产品，既无需取得权利人的同意（授权），也不必向权利人支付报酬。例如，中国专利局授予的专利权或中国商标局核准的商标专用权只 能在中国领域内受保护，在其他国家则不给予保护。外国人在我国领域外使用中国专利 局授权的发明专利不侵犯我国专利权，如美国人在美国使用我国专利局授权的发明专利不侵犯我国专利权。

通过缔结有关知识产权的国际公约或双边互惠协定的形式，某一国家的国民（自然 人或法人）的知识产权在其他国家（缔约国）也能取得权益。参加知识产权国际公约的 国家（或者签订双边互惠协定的国家）会相互给予成员国国民的知识产权保护。所以， 我国公民、法人完成的发明创造要想在外国受保护，必须在外国申请专利。商标要想在 外国受保护，必须在外国申请商标注册。著作权虽然自动产生，但它受地域限制，我国 法律对外国人的作品并不是都给予保护，只保护共同参加国际条约国家的公民作品。同样，参加公约的其他成员国也按照公约规定，对我国公民和法人的作品给予保护。虽然 众多知识产权国际条约等的订立使地域性有时会变得模糊，但地域性的特征不但是知识产权最“古老”的特征，也是最基础的特征之一。目前知识产权的地域性仍然存在，是否授予权利、如何保护权利仍须由各缔约国按照其国内法来决定。

本题涉及的依照该专利生产的产品在中国或其他国家销售，中国 M 企业不需要向美国 L 公司支付这件美国专利的许可使用费。这是因为 L 公司未在中国及其他国家申请该专利，不受中国及其他国家专利法的保护，因此依照该专利生产的产品在中国及其他国 家销售，M 企业不需要向 L 公司支付这件专利的许可使用费。如果返销美国，需要向 L 公司支付这件专利的许可使用费。这是因为这件专利已在美国获得批准，因而受到美国专利法的保护，M 企业依照该专利生产的产品要在美国销售，则需要向 L 公司支付这件专利的许可使用费。

使用(12) DPI 的分辨率扫描一幅 2x4 英寸的照片，可以得到一幅 300x600 像素的图像。

(12)A.100　　B.150　　C.300　　D.600

【答案】B

【解析】本题考查多媒体基础知识。

我们经常遇到的分辨率有两种，即显示分辨率和图像分辨率。显示分辨率是指显示屏上能够显示出的像素数目。例如，显示分辨率为 1024x768 表示显示屏分成 768 行（垂直分辨率），每（水平分辨率）显示 1024 个像素，整个显示屏就含有 796 432 个显像点。屏幕能够显示的像素越多，说明显示设备的分辨率越高， 显示的图像质量越高。图像分辨率是指组成一幅图像的像素密度，也是用水平和垂直的像素表示，即用每英寸多少点（dpi)表示数字化图像的大小。例如，用 200dpi 来扫描一 幅 2x2.5 英寸的彩色照片，那么得到一幅 400x500 个像素点的图像。它实质上是图像数字化的采样间隔，由它确立组成一幅图像的像素数目。对同样大小的一幅图，如果组成该图的图像像素数目越多，则说明图像的分辨率越高，图像

看起来就越逼真。相反，图像显得越粗糙。因此，不同的分辨率会造成不同的图像清晰度。

计算机数字音乐合成技术主要有（13）两种方式,其中使用（14）合成的音乐，其音质更好。

(13)A.FM 和 AM　　B.AM 和 PM　　C.FM 和 PM　　D.FM 和 Wave Table

(14)A.FM　　B.AM　　C.PM　　D.Wave Table

【答案】D　D

【解析】本题考查多媒体基础知识。

计算机和多媒体系统中的声音，除了数字波形声音之外，还有一类是使用符号表示的，由计算机合成的声音包括语音合成和音乐合成。音乐合成技术主要有调频（FM)音乐合成、波形表(WaveTable)音乐合成两种方式。调频音乐合成是使高频振荡波的频率按调制信号规律变化的一种调制方式。采用不同调制波频率和调制指数就可以方便地合成具有不同频谱分布的波形，再现某些乐器的音色。可以采用这种方法得到具有独特效果的“电子模拟声”，创造出丰富多彩的声音，是真实乐器所不具备的音色。波形表音乐合成是将各种真实乐器所能发出的所有声音（包括各个音域、 声调）录制下来，存储为一个波表文件。播放时，根据 MIDI 文件记录的乐曲信息向波表发出指令，从“表格”中逐一找出对应的声音信息，经过合成、加工后回放出来。应用调频音乐合成技术的乐音已经很逼真，波形表音乐合成技术的乐音更真实。目前这两种音乐合成技术都应用于多媒体计算机的音频卡中。

数据流图(DFD)对系统的功能和功能之间的数据流进行建模，其中顶层数据流图描述了系统的（15)。

(15)A.处理过程　　B.输入与输出　　C.数据存储　　D.数据实体

【答案】B

【解析】本题考查数据流图的基本概念。

数据流图从数据传递和加工的角度,以图形的方式刻画数据流从输入到输出的移动变换过程,其基础是功能分解。对于复杂一些的实际问题，在数据流图中常常出现许多加工，这样看起来不直观，也不易理解，因此用分层的 数据流图来建模。按照系统的层次结构进行逐步分解，并以分层的数据流图反映这种结构关系。

在分层的数据流图中，各层数据流图之间应保持“平衡”关系，即输入和输出数据流在各层

应该是一致的。

模块 A 执行几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，则该模块具有（16)内聚。

(16)A.顺序　　B.过程　　C.逻辑　　D.功能

【答案】C

【解析】本题考查软件设计的相关内容。

模块独立性是创建良好设计的一个重要原则，一般采用模块间的耦合和模块的内聚两个准则进行度量。内聚是指模块内部各元素之间联系的紧密程度，内聚度越高，则模块的独立性越好。内聚性一般有以下几种：

①偶然内聚：指一个模块内的各个处理元素之间没有任何联系。

②逻辑内聚：指模块内执行几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。

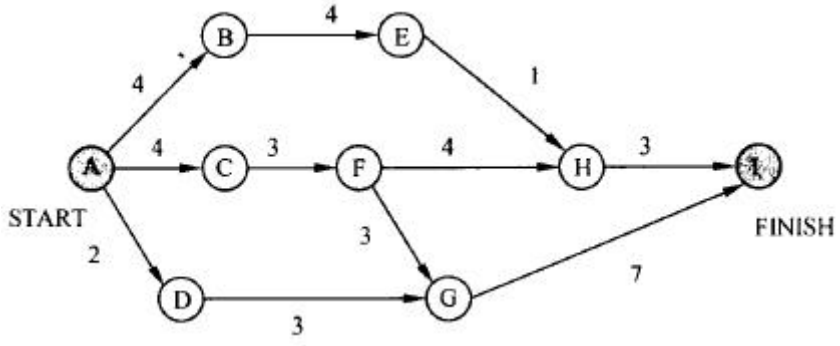
③时间内聚：把需要同时执行的动作组合在一起形成的模块。

④通信内聚：指模块内所有处理元素都在同一个数据结构上操作，或者指各处理使用相同的输入数据或者产生相同的输出数据。

⑤顺序内聚：指一个模块中各个处理元素都密切相关于同一功能且必须顺序执行， 前一个功能元素的输出就是下一个功能元素的输入。

⑥功能内聚：是最强的内聚，指模块内所有元素共同完成一个功能，缺一不可。

下图是一个软件项目的活动图，其中顶点表示项目里程碑，连接顶点的边表示活动， 边上的值表示完成活动所需要的时间，则(17)在关键路径上。



(17)A.B　　B.C　　C.D　　D.H

【答案】B

【解析】本题考查项目管理及工具技术。

根据关键路径法，计算出关键路径为A-C—F—0—1，关键路径长度为17。因此里程碑C在关键路径上，而里程碑B、D和H不在关键路径上。

(18)最不适于采用无主程序员组的开发人员组织形式。

(18)A.项目开发人数少（如3～4人）的项目　　B.采用新技术的项目

C.大规模项目　　D.确定性较小的项目

【答案】C

【解析】本题考查项目管理的人员管理。

程序设计小组的组织形式一般有主程序员组、无主程序员组和层次式程序员组。其中无主程序员组中的成员之间相互平等，工作目标和决策都由全体成员民主讨论。对于项目规模较小、开发人员少、采用新技术和确定性较小的项目比较合适，而对大规模项目不适宜采用。

若软件项目组对风险采用主动的控制方法，则（19）是最好的风险控制策略。

(19)A.风险避免　B.风险监控　C.风险消除　D.风险管理及意外事件计划

【答案】A

【解析】本题考查项目管理的风险管理。

风险控制的目的是辅助项目组建立处理风险的策略。有效的策略必须考虑以下三个问题，即风险避免、风险监控和风险管理及意外事件计划，而其中风险避免是最好的风险控制策略。

对于逻辑表达式“x and y or not z”，and、or、not分别是逻辑与、或、非运算，优先级 从高到低为not、and、or, and、or为左结合，not为右结合，若进行短路计算，则(20)。

(20)A.x为真时，整个表达式的值即为真，不需要计算y和z的值

B.x为假时，整个表达式的值即为假，不需要计算y和z的值

C.x为真时，根据y的值决定是否需要计算z的值

D.x为假时，根据y的值决定是否需要计算z的值

【答案】C

【解析】本题考查程序语言基础知识。

对逻辑表达式可以进行短路计算，其依据是：a and b的含义是a和b同时为“真”，则a and b为“真”，因此，若a为“假”，则无论b的值为“真”或“假”，a and b必然 为“假”；a or b的含义是a和b同时为“假”，则a or b为“假”，因此，若a为“真”， 则无论b

的值为“真”或“假”，a or b必然为“真”。

在优先级和结合性规定下，对逻辑表达式“x and y or not z”求值时，应先计算“x and y”的值，若为“假”，才去计算“not z”的值。因此,若x的值为“假”，则“x and y” 的值为“假”，需要计算“not z”来确定表达式的值而不管y是“真”是“假”。当x的 值为“真”，则需要计算y的值：若y的值为“真”，则整个表达式的值为“真”(从而不需再计算“not z”)；若y的值为“假”，则需要计算“not z”来确定表达式的值。

对于二维数组a[1..N，1..N]中的一个元素a[ij] (1≤i,j≤N)，存储在a[ij]之前的元素个数 (21)。

(21)A.与按行存储或按列存储方式无关

B.在i=j时与按行存储或按列存储方式无关

C.在按行存储方式下比按列存储方式下要多

D.在按行存储方式下比按列存储方式下要少

【答案】B

【解析】

本题考查数组元素的存储。

二维数组a[1..N，1..N]的元素布局如下：

a[1,1]	a[1,2]	…	a[1,j]	…	a[1,N]
a[2,1]	a[2,2]	…	a[2,j]	…	a[2,N]
⋮	⋮		⋮		⋮
a[i,1]	a[i,2]	…	a[i,j]	…	a[i,N]
⋮	⋮		⋮		⋮
a[N,1]	a[N,2]	…	a[N,j]	…	a[N,N]

在按行存储方式下，a[ij]之前的元素个数为(i-1)*N+j-1;在按列存储方式下，a[ij] 之前的元素个数为(j-1)*N+i-1。若i=j，则a[ij]是主对角线上的元素，显然(i-1)*N+j-1 与(j-1)*N+i-1相等。若i<j，则a[ij]是上三角区域的元素；若i>j，则a[i,j]是下三角区域的元素，这两种情况下，存储在a[i,j]之前的元素个数分别为和(j-1)*N+i-1， 其大小关系依赖于i和j的具体取值。

算术表达式 x-(y+c)*8 的后缀式是（22）(-、+、*表示算术的减、加、乘运算，运算符的优先级和结合性遵循惯例)。

(22)A. xyc8-+*　　B. xy-c+8*　　C. xyc8*+-　　D. xyc+8*-

【答案】D

【解析】本题考查程序语言基础知识。

后缀表达式（也叫逆波兰式，Reverse Polish notation)是将运算符写在操作数之后的表达式表示方法。

表达式“x-(y+c)*8”的后缀式为“xyc+8*-”。

若某企业拥有的总资金数为 15,投资 4 个项目 PI、P2、P3、P4,各项目需要的最大资金数分别是 6、8、8、10,企业资金情况如图 a 所示。P1 新申请 2 个资金，P2 新申请 1 个资金，若企业资金管理处为项目 P1 和 P2 分配新申请的资金，则 P1、P2、P3、 P4 尚需的资金数分别为（23),假设 P1 已经还清所有投资款，企业资金使用情况如图 b 所示，那么企业的可用资金数为（24)。若在图 b 所示的情况下，企业资金管理处为 P2、P3、P4 各分配资金数 2、2、3,则分配后 P2、P3、P4 已用资金数分别为（25)。

项目	最大资金	已用资金	尚需资金
P1	6	2	4
P2	8	3	5
P3	8	2	6
P4	10	3	7

图 a

项目	最大资金	已用资金	尚需资金
P1	—	—	—
P2	8	3	5
P3	8	2	6
P4	10	3	7

图 b

(23)A. 1、3、6、7,可用资金数为 0,故资金周转状态是不安全的

B. 2、5、6、7,可用资金数为 1，故资金周转状态是不安全的

C. 2、4、6、7，可用资金数为 2,故资金周转状态是安全的

D. 3、3、6、7，可用资金数为 2,故资金周转状态是安全的

(24)A. 4　　B. 5　　C. 6　　D. 7

(25)A. 3、2、3，尚需资金数分别,5、6、7，故资金周转状态是安全的

B. 5、4、6，尚需资金数分别,3、4、4，故资金周转状态是安全的

C. 3、2、3，尚需资金数分别为 5、6、7，故资金周转状态是不安全的

D. 5、4、6 尚需资金数分别为,3、4、4，故资金周转状态是不安全的

【答案】C　D　D

【解析】本题考查操作系统进程管理方面的基础知识。

在图 a 的情况下，项目 P1 申请 2 个资金，P2 申请 1 个资金，则企业资金管理处分配资金后项目 P1、P2、P3、P4 已用的资金数分别为 4、4、2、3,可用资金数为 2,故尚需的资金数分别为 2、4、6、7。由于可用资金数为 2,能保证项目 P1 完成。假定项目 P1 完成释放资源后，可用资金数为 6,能保证项目 P2 或 P3 完成。同理，项目 P2 完成释放资源后，可用资金数为 10,能保证项目 P3 或 P4 完成，故资金周转状态是安全的。

(24) 对于图 b，因为企业的总资金数是 15，企业资金管理处为项目 P2、P3、P4 已分配资金数为 3、2、3,故可用资金数为 7。

(25) 在图 b 的情况下，企业资金管理处为项目 P2、P3、P4 已分配资金数为 3、2、3,若 企业资金管理处又为项目 P2、P3、P4 分配资金数为 2、2、3,则企业分配后项目 P2、 P3、P4 已用资金数分别为 5、4、6,可用资金为 0，尚需资金数分别为 3、4、4,故资金周转状态是不安全的。

假设一台按字节编址的 16 位计算机系统，采用虚拟页式存储管理方案，页面的大小为 2K，且系统中没有使用快表（或联想存储器）。某用户程序如图 a 所示，该程序的页面变换表如图 b 所示，表中状态位等于 1 和 0 分别表示页面在内存或不在内存。

图 a

页面变换表

页号	状态
0	1
1	0
2	0
3	0
4	0
5	0

图 b

图 a 中 MOVE Data1, Data2 是一个 4 字节的指令，Data1 和 Data2 表示该指令的两个 32 位操作数。假设 MOVE 指令存放在 2047 地址开始的内存单元中，Data1 存放在 6143 地址开始的内存单元中，Data2 存放在 10239 地址开始的内存单元中，那么执行 MOVE 指令将产生 (26)次缺页中断,其中：取指令产生 (27)次缺页中断，取 Data1 和 Data2 操作数分别产生 (28)次缺页中断。

(26)A.3　　B.4　　C.5　　D.6

(27)A.0　　B.1　　C.2　　D.3

(28)A.1、1　　B.1、2　　C.2、2　　D.2、3

【答案】C　B　C

【解析】本题考查操作系统中系统内存管理方面的知识。

从题图中可见，程序的MOVE指令跨两个页面，且源地址Datal和目标地址Data2所涉及的区域也跨两个页面的页内地址， 根据题意1、2、3、4和5号页面不在内存，系统取“MOVE Datal，Data2”的指令时， 由于该指令跨越两个页面0、1，查页面变换表可以发现1号页面不在内存，故需要产生一次缺页中断；取地址为Datal的操作数时，由于该操作数不在内存且跨两个页面2、3, 需要将2、3页面装入内存，所以产生两次缺页中断；同理，取地址为Data2的操作数时， 由于该操作数不在内存且跨两个页面4、5,需要将4、5页面装入内存，所以产生两次缺页中断，共产生5次缺页中断。

软件开发的增量模型（29）。

(29)A.最适用于需求被清晰定义的情况

B.是一种能够快速构造可运行产品的好方法

C.最适合于大规模团队开发的项目

D.是一种不适用于商业产品的创新模型 试题

【答案】B

【解析】本题考查软件开发过程模型。

增量模型是一种阶段化的软件开发过程模型。在该过程模型中，客户提出系统需求，并指出哪些需求是最重要的。开发团队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构成， 并且能完成特定的功能。其优点包括：能较短时间内向用户提交可完成一些有用的工作产品；逐步增加产品的功能，使用户有较充裕的时间学习和适应新产品；项目失败的风险较低；优先级最高的服务首先交付，然后依次将其他构件集成进来，这意味着最重要的服务将接受最多的测试。因此增量模式是一种能够快速构造可运行产品的方法，也适用于今天竞争激烈，需要快速发布产品的市场环境。

假设某软件公司与客户签订合同开发一个软件系统，系统的功能有较清晰的定义， 且客户对交付时间有严格要求，则该系统的开发最适宜采用（30）。

(30)A.瀑布模型　　B.原型模型　　C.V模型　　D.螺旋模型

【答案】A

【解析】本题考查软件过程模型。

软件过程是软件生存周期中的一系列相关活动，即用于开发和维护软件及相关产品的一系列活动。瀑布模型从一种非常高层的角度描述了软件开发过程中进行的活动，并且提出了要求开发人员经过的事件序列。该模型适用于项目开始时需求已确定的情况。V 模型是瀑布模型的变种，它说明测试活动是如何与分析和设计相联系的。原型模型允许开发人员快速地构造整个系统或系统的一部分以理解或澄清问题。原型的用途是获知用户的真正需求，因此原型模型可以有效地引发系统需求。螺旋模型把开发活动和风险管理结合起来，以将风险减到最小并控制风险。本题中系统功能有较清晰定义意味着需求较确定，且对交付时间有严格要求，因此最适宜用瀑布模型。

某企业由于外部市场环境和管理需求的变化对现有软件系统提出新的需求，则对该软件系统进行的维护属于 (31)维护。

(31)A.正确性　　B.完善性　　C.适应性　　D.预防性

【答案】C

【解析】

在软件开发完成交付用户使用后，就进入软件运行/维护阶段。软件维护活动根据其内容可以分为 4 种类型：

①正确性维护。为了识别和纠正软件错误，改正软件性能上的缺陷，排除实施的误使用，应进行的诊断和改正错误的过程。

②适应性维护。由于信息技术飞速发展，软件运行的外部环境或数据环境可能发生变化，为了使软件适应这种变化而修改软件的过程。

③完善性维护。在软件使用过程中，用户往往会对软件提出新的功能与性能要求， 为了满足这些要求，需要修改或再开发软件，以扩充软件功能、增强软件性能、改进加工效率、提高软件的可维护性而进行的维护活动。

④预防性维护。为了提高软件的可维护性和可靠性等，为以后进一步改进软件打下良好基础而进行的维护工作。

McCall 软件质量模型从软件产品的运行、修正和转移三个方面确定了 11 个质量特性，其中 (32)不属于产品运行方面的质量特性。

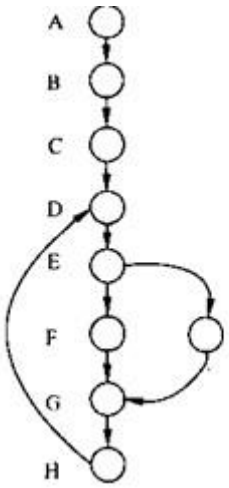
(32)A.正确性　　B.可靠性　　C.效率　　D.灵活性

【答案】D

【解析】本题考查软件质量的相关知识。

McCall 软件质量模型从软件产品的运行、修正和转移三个方面确定了 11 个质量特性。其中产品运行方面包括正确性、可靠性、易使用性、效率和完整性；产品修正方面包括可维护性、灵活性和可测试性；产品转移方面包括可移植性、 复用性和互用性。

采用 McCabe 度量法计算下列程序图的环路复杂性为 (33)。



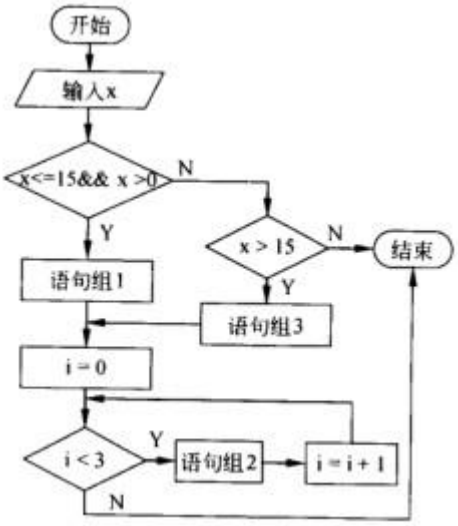
(33)A.2　　B.3　　C.4　　D.5

【答案】B

【解析】本题考查软件复杂性度量。

McCabe 度量法计算程序的环路复杂性为 V(G) = m-n + 2p,其中 V(G)是有向图 G 中的环路数，m 是图 G 中弧的个数，n 是图 G 中顶点的个数，p 为图 G 中的强连通分量数。上图中，弧的个数为 10,顶点的个数为 9, p=1，因此有 V(G) = m-n+2p=10-9+2=3。

在白盒测试法中，(34)是最弱的覆盖准则。下图至少需要 (35)个测试用例，才可以完成路径覆盖语句组 2 不对变量 i 进行操作。



(34)A.语句　　B.条件　　C.判定　　D.路径

(35)A.1　　B.2　　C.3　　D.4

【答案】A　C

【解析】本题考查软件测试的基本概念。

白盒测试也称为结构测试，根据程序的内部结构和逻辑来设计测试用例，对程序的路径和过程进行测试，检查是否满足设计的需要。在白盒测试中，语句覆盖是指选择足够的测试用例，使被测程序中每条语句至少执行一次。它对程序执行逻辑的覆盖很低，因此一般认为是很弱的逻辑覆盖。判定覆盖是指设计足够的测试用例，使得被测程序中每个判定表达式至少获得一次“真”植和“假”值。条件覆盖是指设计足够的测试用例，使得每一个判定语句中每个逻辑条件的各种可能的值至少满足一次。路径覆盖是指覆盖被测程序中所有可能的路径。在这些覆盖技术中，从弱到强依次为语句覆盖、判定覆盖、条件覆盖和路径覆盖。在上图中，要完成路径覆盖，至少需要 3 个测试用例才可以，如测试用例（0)、(8)和（16)即可完成路径覆盖，测试用例格式为（x 的值)。

根据 ISO/IEC 9126 软件质量模型中对软件质量特性的定义，可维护性质量特性的（36)子特性是指与为确认经修改软件所需努力有关的软件属性。

(36)A.易测试性　　B.易分析性　　C.稳定性　　D.易改变性

【答案】A

【解析】本题考查软件质量特性的基础知识。

根据 ISO/IEC9126 软件质量模型的定义，可维护性质量特性包含易分析性、易改变性、稳定性和易测试性 4 个子特性。其中易分析性是指为诊断缺陷或失效原因，或为判定待修改的部

分所需努力有关的软件属性；易改变性是指与进行修改、排错或适应环境变换所需努力有关的软件属性；稳定性是指与修改造成未预料效果的风险有关的软件属性；易测试性是指为确认经修改软件所需努力有关的软件属性。

面向对象技术中，组合关系表示（37）。

(37)A.包与其中模型元素的关系　　B.用例之间的一种关系

C.类与其对象的关系　　D.整体与其部分之间的一种关系

【答案】D

【解析】本题考查面向对象的基本知识。

在面向对象技术中，包用于将关系紧密的模型元素组织在一起，提供一个命名空间， 以提供访问控制。用例之间有继承、包含和扩展关系。类是在对象之上的抽象，对象是类的具体化，对定义好的类的属性的不同赋值就可以得到该类的对象实例。组合关系表示整体与其部分之间的一种关系。

以下关于封装在软件复用中所充当的角色的叙述，正确的是（38）。

(38)A.封装使得其他开发人员不需要知道一个软件组件内部如何工作

B.封装使得软件组件更有效地工作

C.封装使得软件开发人员不需要编制开发文档

D.封装使得软件组件开发更加容易

【答案】A

【解析】本题考查面向对象的基本知识。

封装是一种信息隐藏技术，其目的是使对象（组件）的使用者和生产者分离，也就是使其他开发人员无需了解所要使用的软件组件内部的工作机制，只需知道如何使用组件，即组件提供的功能及其接口。

在有些程序设计语言中，过程调用和响应调用需执行的代码的绑定直到运行时才进行，这种绑定称为（39）。

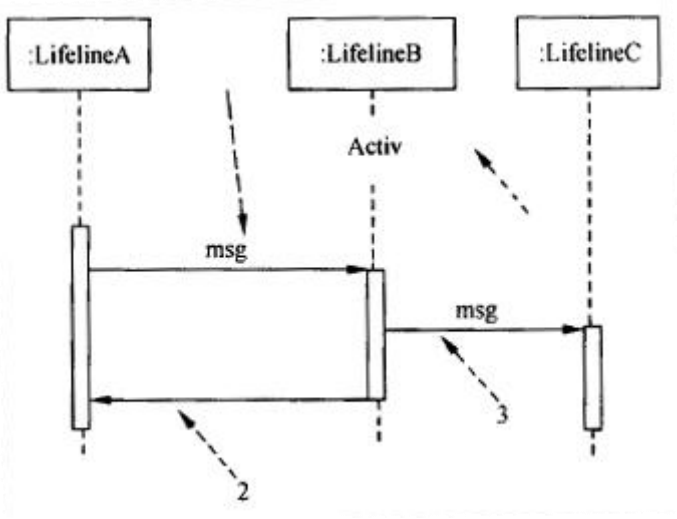
(39)A.静态绑定　　B.动态绑定　　C.过载绑定　　D.强制绑定

【答案】B

【解析】本题考查面向对象的基本知识。

在面向对象系统中，绑定是一个把过程调用和响应调用需要执行的代码加以结合的过程。在有些程序设计语言中，绑定是在编译时进行的，叫做静态绑定。在有些程序设 计语言中，绑定则是在运行时进行的，即一个给定的过程调用和响应调用需执行的代码的结合直到调用发生时才进行。

UML 序列图是一种交互图，描述了系统中对象之间传递消息的时间次序。其中，异步消息与同步消息不同，(40)。下图中（41）表示一条同步消息，(42)表示一条异步消息，(43)表示一条返回消息。



(40)A.异步消息并不引起调用者终止执行而等待控制权的返回

B.异步消息和阻塞调用有相同的效果

C.异步消息是同步消息的响应

D.异步消息和同步消息一样等待返回消息

(41)A.1　　B.2　　C.3　　D.4

(42)A.1　　B.2　　C.3　　D.4

(43)A.1　　B.2　　C.3　　D.4

【答案】A　A　C　B

【解析】本题考查统一建模语言（UML)的基本知识。

UML2.0 中提供了多种图形，序列图是场景的图形化表示，描述了以时间顺序组织的对象之间的交互活动。其中消息定义了交互中生命线之间的特定交互，有同步消息、异步消息和返回消息三类。同步消息指进行阻塞调用，调用者中止执行，等待控制权返回，需要等待返回消息；而异步消息的调用者发出消息后继续执行，不引起调用者阻塞， 也不等待返回消息。

消息由名称进行标识，还描述出消息的发出者和接收者。异步消息由空心箭头表示，如上图中 3 所示，同步消息用实心三角箭头表示，如上图中 1 所示，返回消息。

设计模式根据目的进行分类；可以分为创建型、结构型和行为型三种。其中结构型模式用于处理类和对象的组合。(44)模式是一种结构型模式。

(44)A.适配器（Adapter）　　B.命令（Command）

C.生成器（Builder）　　D.状态（State）

【答案】A

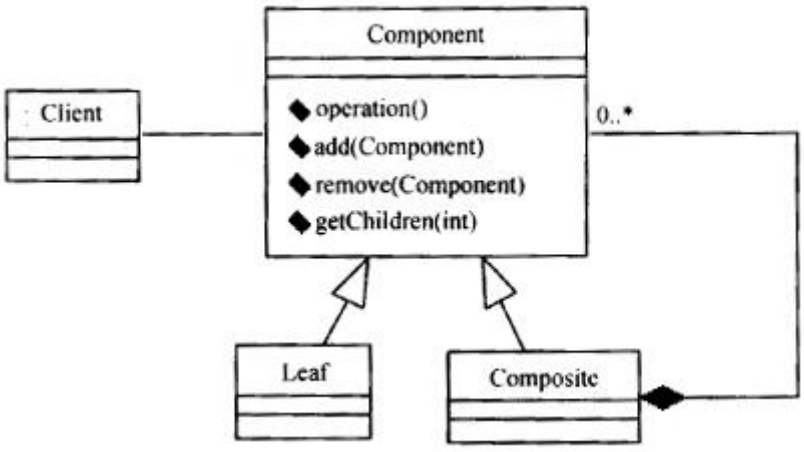
【解析】本题考查设计模式的基本知识。

每一个设计模式描述了一个在我们周围不断重复发生的问题，以及该问题的解决方案的核心，使该方案能够重用而不必做重复劳动。

设计模式根据目的进行分类，可以分为创建型、结构型和行为型三种。其中创建型模式与对象的创建有关；结构型模式用于处理类和对象的组合；行为型模式描述类或对象怎样交互和怎样分配职责。

适配器（Adapter)模式是一种结构型模式；命令（Command)模式和状态（State）模式是行为型模式；生成器（Builder)模式是一种创建型模式。

设计模式中的(45)模式将对象组合成树形结构以表示“部分-整体”的层次结构， 使得客户对单个对象和组合对象的使用具有一致性。下图为该模式的类图，其中，(46) 定义有子部件的那些部件的行为；组合部件的对象由 (47)通过 Component 提供的接口操作。



(45)A.代理（Proxy）　　B.桥接器（Bridge）

C.组合（Composite）　　D.装饰器（Decorator）

(46)A.Client　　B.Component　　C.Leaf　　D.Composite

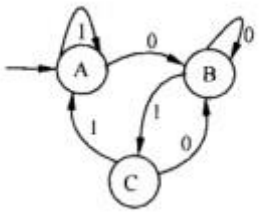
(47)A.Client　　B.Component　　C.Leaf　　D.Composite

【答案】C　D　A

【解析】本题考查设计模式的基本知识。

代理(Proxy)模式是为其他对象提供一种代理以控制对这个对象的访问，使得只有在确实需要这个对象时才对其进行创建和初始化。桥接器(Bridge)模式将对象的抽象和其实现分离，从而可以独立地改变它们，当一个抽象可能有多个实现时，抽象类定义对该抽象的接口，而具体的子类则用不同方式加以实现。组合(Composite)模式描述了如何将对象组合成树形结构以构造一个层次结构来表示“部分-整体”，使得客户对单个对象和组合对象的使用具有一致性，这一结构由两种类型的对象所对应的类构成，使得可以组合基元对象以及其他的组合对象，从而形成任意复杂的结构。上阁中的 Composite 定义有子部件的那些部件的行为；组合部件的对象由 Client 通过 Component 提供的接 LI 操作。装饰器 (Decorator)模式则描述动态地给一个对象添加一些额外的职责。

下图所示为一个有限自动机（其中，A 是初态、C 是终态），该自动机所识别的字符串的特点是 (48)。



(48)A.必须以 11 结尾的 0、1 串　　B.必须以 00 结尾的 0、1 串

C.必须以 01 结尾的 0、1 串　　D.必须以 10 结尾的 0、1 串

【答案】C

【解析】本题考查程序语言处理基础知识。

从有限自动机的初态到终态的路径上的标记形成其可识别的字符串。

对于题中的自动机，从 A 出发到达 C 结束的所有路径中必然包含 BC 这条弧(标记为 1)，同时到达 B 的弧上都标记了 0,所以其识别的字符串必须以 01 结尾。

E-R 模型向关系模型转换时，三个实体之间多对多的联系 m:n:p 应该转换为一个独立的

关系模式，且该关系模式的关键字由（49)组成。

(49)A.多对多联系的属性　　B.三个实体的关键字

C.任意一个实体的关键字　　D.任意两个实体的关键字

【答案】B

【解析】本题考查数据库设计方面的基础知识。

E-R 模型向关系模型转换时，两个以上实体之间多对多的联系应该转换为一个独立的关系模式，且该关系模式的关键字由这些实体的关键字组成。

函数（过程）调用时，常采用传值与传地址两种方式在实参与形参间传递信息。以下叙述中，正确的是（50)。

(50)A.在传值方式下，将形参的值传给实参，因此，形参必须是常量或变量

B.在传值方式下，将实参的值传给形参，因此，实参必须是常量或变量

C.在传地址方式下，将形参的地址传给实参，因此，形参必须有地址

D.传地址方式下，将实参的地址传给形参，因此，实参必须有地址

【答案】D

【解析】本题考查程序语言处理基础知识。

一个函数被调用时，可能需要接收从外部传入的数据信息，传值调用与引用调用（传地址）是函数调用时常采用的信息传递方式。传值调用是将实参的值传给被调用函数的形参，因此实参可以是常量、变量、表达式或函数调用，而引用调用的实质是将实参的地址传给被调用函数的形参，因此实参必须具有地址。

编译和解释是实现高级程序设计语言翻译的两种基本形式。以下关于编译与解释的叙述中，正确的是（51)。

(51)A.在解释方式下，对源程序不进行词法分析和语法分析，直接进行语义分析

B.在解释方式下，无需进行词法、语法和语义分析，而是直接产生源程序的目标代码

C.在编译方式下，必须进行词法、语法和语义分析，然后再产生源程序的目标代码

D.在编译方式下，必须先形成源程序的中间代码，然后再产生与机器对应的目标代码

【答案】C

【解析】本题考查程序语言处理基础知识。、

程序的翻译通常有两种基本方式：一种是编译方式，另一种是解释方式。在编译方式下，首先将源程序翻译为等价的目标程序，源程序的翻译和目标程序的运行是完全独立的两个阶段；而解释方式下，对源程序的翻译和运行是结合在一起进行的，并不生成目标代码。

编译过程基本上可以划分为词法分析、语法分析、语义分析、中间代码生成、代码优化和目标代码生成等几个阶段，其中，中间代码生成和代码优化不是必须的。在词法、语法、语义分析方面，编译方式和解释方式没有区别。

若对关系 R(A，B，C,D)进行化π1，3(R)运算，则该关系运算与 (52)等价，表示 (53)。

(52)A. πA=1,C=3(R)　　B. πA=1^C=3(R)　　C. πA,C(R)　　D. πA=1∨C=3(R)

(53)A.属性 A 和 C 的值分别等于 1 和 3 的元组为结果集

B.属性 A 和 C 的值分别等于 1 和 3 的两列为结果集

C.对 R 关系进行 A=1、C=3 的投影运算

D.对 R 关系进行属性 A 和 C 的投影运算

【答案】C　D

【解析】本题考查关系代数运算方面的基础知识。

投影运算 n 是向关系的垂直方向进行运算，其含义为在关系 R 中选择出若干属性列 组成新的关系，记作： πAi，Ai2,…，Aii(R) = {t[Ai1,Ai2…，Ain]| t∈R]。本试题中，关系 R(A, B, C, D)共有 4 个属性，属性 A、B、C 和 D 分别位于第 1 列、第 2 列、第 3 列 和第 4 列， π1,3(R)相当于在关系 R 的第 1 列和第 3 列上进行投影运算，即在关系 R 的属 性 A 和 C 上进行投影运算，故π1,3(R)与πA,C(R)是等价的。

某销售公司数据库的零件关系 P (零件号，零件名称，供应商，供应商所在地，库存量)，函数依赖集 F={零件号一>零件名称，(零件号，供应商）一>库存量，供应商一>供应商所在地}。零件关系模式 P 属于 (54)。

查询各种零件的平均库存量、最多库存量与最少库存量之间差值的 SQL 语句如下：

SELECT 零件号，零件名称， (55)

FROM　P

(56) ；

(54)A.1NF　　B.2NF　　C.3NF　　D.4NF

(55)A.AVG(库存量)AS 平均库存量，MAX(库存量)-MIN(库存量)AS 差值

B.平均库存量 ASAVG(库存量)，差值 AS MAX(库存量)-MIN(库存量)

C.AVG 库存量 AS 平均库存量，MAX 库存量-MIN 库存量 AS 差值

D.平均库存量 ASAVG 库存量，差值 AS MAX 库存量-MIN 库存量

(56)A.ORDER BY 供应商　　B.ORDER BY 零件号

C.GROUP BY 供应商　　D.GROUP BY 零件号

【答案】A　A　D

【解析】本题考查关系数据库及 SQL 方面的基础知识。

根据题意，零件 P 关系中的（零件号，供应商）可决定零件 P 关系的所有属性，所以零件 P 关系的主键为（零件号，供应商）；又因为，根据题意（零件号，供应商）—> 零件名称，而零件号—>零件名称，供应商—>供应商所在地，可以得出零件名称和供应商所在地都部分依赖于码，所以该关系模式属于 1NF。

(55,56) 查询各种零件的平均库存量、最高库存量与最低库存量之间差距时，首先需要在结果列中的空（55)处填写“AVG (库存量）AS 平均库存量，MAX(库存量)-MIN(库存量)AS 差值”。其次必须用分组语句按零件号分组，故空（56)应填写“GROUP BY 零件号”。

对于一个长度大于 1 且不存在重复元素的序列，令其所有元素依次通过一个初始为空的队列后，再通过一个初始为空的找。设队列和栈的容量都足够大，一个序列通过队列（栈）的含义是序列的每个元素都入队（栈）且出队列（栈）一次且仅一次。对于 该序列在上述队列和栈上的操作，正确的叙述是（57)

(57)A.出队序列和出栈序列一定相同

B.出队序列和出栈序列一定互为逆序

C.入队序列与出队序列一定相同，入栈序列与出栈序列不一定相同

D.入栈序列与出栈序列一定互为逆序，入队序列与出队序列不一定互为逆序

【答案】C

【解析】本题考查数据结构基础知识。

栈和队列是两种常用的数据结构。栈的特点是后进先出，队列的特点是先进先出。 因此，入队序列与出队序列一定相同。在入栈序列一定的情况下，由于元素的出栈时机不同，

会形成不同的出栈序列，入栈序列与出栈序列可以相同，也可以不同。

在字符串的 KMP 模式匹配算法中，需要求解模式串 p 的 next 函数值，其定义如下所示。若模式串 p 为“aaabaaa”，则其 next 函数值为 (58)。

$$next[j]=\begin{cases}0 & j=1\\ \max\{k \mid 1<k<j,'p_1p_2\cdots p_{k-1}'='p_{j-k+1}p_{j-k+2}\cdots p_{j-1}'\} & \\ 1 & \text{其他情况}\end{cases}$$

(58)A.0123123　　B.0123210　　C.0123432　　D.0123456

【答案】A

【解析】本题考查字符串的模式匹配运算。

KMP 模式匹配算法是对基本模式匹配算法的改进，其改进之处在于：每当匹配过程中出现相比较的字符不相等时，不需要回溯主串的字符位置指针，而是利用已经得到的“部分匹配”结果将模式串向右“滑动”尽可能远的距离，再继续进行比较。

在 KMP 算法中，依据模式串的 next 函数值实现子串的滑动。若令 next[j]=k，则 next[j]表示当模式串中的 Pj 与主串中相应字符不相等时，令模式串的 pk 与主串的相应字符进行比较。

根据 next 的定义，模式串“aaabaaa”的 next 函数值为 0123123。

若 n2、n1、n0 分别表示一个二叉树中度为 2、度为 1 和叶子结点的数目（结点的度定义为结点的子树数目），则对于任何一个非空的二叉树，(59)。

(59)A.n2 一定大于 n1　　B.n1一定大于 n0　　C.n2一定大于 n0。　D.n0一定大于 n2

【答案】D

【解析】本题考查数据结构中二叉树的基础知识。

对任何一棵二叉树，若其终端节点数为 n0,度为 2 的节点数为 n2,则 n0=n2+1。证明如下：

设一棵二叉树上叶结点数为 n0,单分支结点数为 m，双分支结点数为 n2,则总结点数=n0+n1+n2。

在一棵二叉树中，所有结点的分支数（即度数）应等于单分支结点数加上双分支结点数的 2 倍，即总的分支数=n1+2n2。

由于二叉树中除根结点以外，每个结点都有唯一的一个分支指向它，因此二叉树中：总的分支数=总结点数-1。因此，n1+2n2= n0+n1+ n2-1,即 n0=n2+1。

从存储空间的利用率角度来看，以下关于数据结构中图的存储的叙述，正确的是(60)。

(60)A.有向图适合采用邻接矩阵存储，无向图适合采用邻接表存储

B.无向图适合采用邻接矩阵存储，有向图适合采用邻接表存储

C.完全图适合采用邻接矩阵存储

D.完全图适合采用邻接表存储

【答案】C

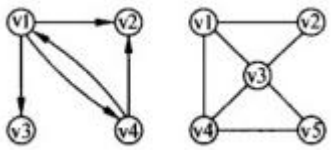
【解析】本题考查数据结构中图结构基础知识。

图的基本存储结构有邻接矩阵表示法和邻接链表表示法。图的邻接矩阵表示利用一个矩阵来表示图中顶点之间的关系。对于具有 n 个顶点的图 G=(V，E),其邻接矩阵是一个 n 阶方阵，且满足：

$$A[i][j]=\begin{cases}1 若(v_i,v_j)或<v_i,v_j>是E中的\\0 若(v_i,v_j)或<v_i,v_j>不是E中的\end{cases}$$

邻接表存储是指为图的每个顶点建立一个单链表，第 i 个单链表中的结点表示依附 于顶点 Vi 的边（对于有向图是以 Vi 为尾的弧)。

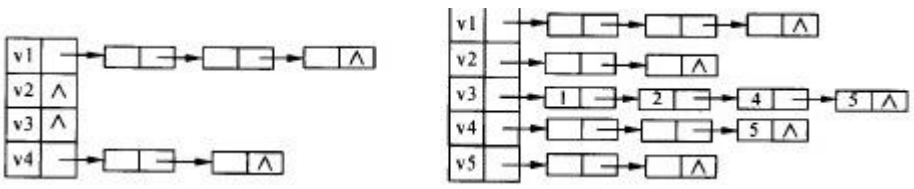
例如，下面分别为一个有向图 a 和一个无向图 b。



有向图 a 和无向图 b 的邻接矩阵如下所示

$$A=\begin{bmatrix}0&1&1&1\\0&0&0&0\\0&0&0&0\\1&1&0&0\end{bmatrix}\qquad B=\begin{bmatrix}0&1&1&1&0\\1&0&1&0&0\\1&1&0&1&1\\1&0&1&0&1\\0&0&1&1&0\end{bmatrix}$$

有向图 a 和无向图 b 的邻接表如下图所示。



图中的顶点数决定了邻接矩阵的阶和邻接表中的单链表数目，无论是对有向图还是无向图，

边数的多少决定了单链表中的结点数，而不影响邻接矩阵的规模，因此完全图适合采用邻接矩阵存储。

递增序列 A(a1,a2…，an)和 B 的（b1,b2，…,bn）元素互不相同，若需将它们合并为一个长度为 2n 的递增序列，则当最终的排列结果为(61)时，归并过程中元素的比较次数最多。

(61)A. a1,a2…，an, b1,b2，…,bn

B. b1,b2，…,bn,a1,a2…，an

C. a1,b1,a2,b2,…,ai,bi,…,an,bn

D. a1,a2,…,ai/2,b1,b2,…，bi/2,ai/2+1,…,an,bi/2+1，bi/2+2,bn

【答案】C

【解析】本题考查归并排序算法。

归并的过程是：取序列 A 的一个元素+和序列 B 的一个元素 bj，若 ai >bj，则输出 bj,接下来令七与 1^+1 比较，否则输出 ai;接下来令 ai+1 与 bj 比较，重复以上过程直至将所有元素输出。

对于最终排列 a1,a2…，an, b1,b2，…,bn 的情况，归并过程中进行了 n 次比较，分别是 a1<b1，a2<b1…，an<b1 最后依次输出 b1,b2,…,bn。

对于最终排列为 b1,b2，…,bn,a1,a2…，an 的情况，归并过程中进行了 n 次比较，分别是 b1<a1，b2<a1,…，bn<a1,最后依次输出 a1,a2,…an

对于最终排列为 a1,b1,a2,b2,…,ai,bi,…,an,bn 的情况，归并过程中进行了 2n-1 次比较，分别是 a1<b1，b1<a2，a2<b2，b2<a3…，an<bn.

若最终排列为 a1,a2, …， ai/2,b1,b2, … ,bi/2,ai/2+1,ai/2+2, …，an,bi/2+1,bi/2+2,…，bn 则在归并过程中，分别是 a1,a2,…,a1/2,各进行一次比较，共 i/2 次；然后是 b1,b2，…，bi/2 分别与 ai/2+1 各进行一次比较，共(n-i/2)次；接下来是 ai/2+1,ai/2+2, …， an 各进行一次比较， 共（n-i/2)次，合计比较次数为 i/2+i/2+n-i/2=n+i/2。

以下关于渐进符号的表示中，不正确的是(62)。

(62)A. $n^2=\Theta(n^2)$ B. n2=O(n2) C. n2=O(n) D. n2=O(n3)

【答案】C

【解析】本题考查算法分析技术的相关知识

几个算法时间复杂度符号的定义分别如下：

O 记号：给出一个函数的渐进上界。给定一个函数 g(n)，O(g(n))表示为一个函数集合的{f(n):#在正常数 c 和 no，使得对所有的 n>no，有 0≤f(n)≤cg(n)}

记号：给出一个函数的渐进下界。给定一个函数 g(n)，n(g(n))表示为一个函数集合{f(n):存在正常数 c 和 no,使得对所有的 n>no，有 0≤cg(n)≤f(n)}

记号：给出一个函数的渐进上界和下界，即渐进确界。给定一个函数 g(n)，0(g(n)) 表示为一个函数集合{f(n):存在正常数 C1、C2 和 n0,使得对所有的 n≥n0，有 n≥(0≤c1g(n)≤f(n)≤c2g(n)}

根据定义，可知 C 不正确。

某货车运输公司有一个中央仓库和 n 个运输目的地，每天要从中央仓库将货物运输到所有的运输目的地，到达每个运输目的地一次且仅一次，最后回到中央仓库。在两个地点 i 和 j 之间运输货物存在费用 cij。为求解旅行费用总和最小的运输路径，设计如下算法：首先选择离中央仓库最近的运输目的地 1，然后选择离运输目的地 1 最近的运输目的地 2，……，每次在未访问过的运输目的地中选择离当前运输目的地最近的运输目的地，最后回到中央仓库。则该算法采用了 (63)算法设计策略，其时间复杂度为 (64)。

(63)A.分治　　B.动态规划　　C.贪心　　D.回溯

(64)A. $\Theta(n^2)$　　B. $\Theta(n)$　　C. $\Theta(n\lg n)$　　D. $\Theta(1)$

【答案】C　A

【解析】本题考查算法设计策略。

由于每次选择下一个要访问的城市时都是基于与当前最近的城市来进行，是一种贪心的选择策略，因此采用的是贪心策略。而货车从中央仓库出发，第一个要到达的目的地是在 n 个目的地中选择一个，第二个要到达的目的地是在 n-1 个目的地中选择一个，……，第 n 个要到达的目的地是在 1 个目的地中选择一个，因此时间复杂度为 n+(n-1)+…+1=n*(n-1)/2= $\Theta(n^2)$

现要对 n 个实数（仅包含正实数和负实数）组成的数组 A 进行重新排列，使得其中所有

的负实数都位于正实数之前。求解该问题的算法的伪代码如下所示，则该算法的时间和空间复杂度分别为(65)。

```
i = 0; j = n-1;
while i<j do
      while A[i] < 0 do
          i = i+1;
      while A[j] > 0 do
          j = j-1;
      if i < j do
```

交换 A[i]和 A[j];

(65)A. $\Theta(n)$ 和 $\Theta(n)$　　B. $\Theta(1)$ 和 $\Theta(n)$　　C. $\Theta(n)$ 和 $\Theta(1)$　　D. $\Theta(1)$ 和 $\Theta(1)$

【答案】C

【解析】本题考查算法分析方法。

根据伪代码可知算法的基本思想，从前往后检查元素，若为负数继续向前检杳；若遇到正数，则开始从后往前检查元素 ，若为正数则继续往前检查若遇到负数则与前面遇到的正数进行交换。重复检查元素，所有元素检查完毕，根据该思想，可知每个元素因此算法的时间复杂度为线性时间，即 $\Theta(n)$ 在该过程中，仅需要一个额外的辅助存储空间，以便进行元素的交换，因此空间复杂度为常数，即 $\Theta(1)$

以下关于网络中各种交换设备的叙述中，错误的是(66)。

(66)A.以太网交换机根据 MAC 地址进行交换

B.帧中继交换机只能根据虚电路号 DLCI 进行交换

C.三层交换机只能根据第三层协议进行交换

D.ATM 交换机根据虚电路标识进行信元交换

【答案】C

【解析】

以太网交换机根据数据链路层MAC地址进行帧交换帧中继网和ATM网都是面向连接的通信网，交换机根据预先建立的虚电路标识进行交换。帧中继的虚电路号是DLCI， 进行交换的协议数据单元为“帧”；而 ATM 网的虚电路号为 VPI 和 VCI，进行交换的协议数据单元为“信元”。三层交换机是指因特网中使用的高档交换机，这种设备把 MAC 交换的高带宽和低延迟优势与网络层分组路由技术结合起来，其工作原理可以概括为：一次路由、多次交换、

就是说，当三层交换机第一次收到一个数据包时必须通过路由功能寻找转发端口，同时记住目标 MAC 地址和源 MAC 地址，以及其他相关信息，当再次收到目标地址和源地址相同的帧时就直接进行交换了，不再调用路由功能。所以三层交换机不但具有路由功能，而且比通常的路由器转发得更快。

SMTP 传输的邮件报文采用（67)格式表示

(67)A.ASCII　　B.ZIP　　C.PNP　　D.HTML

【答案】A

【解析】

本题考查 SMTP 协议及相关服务。SMTP 传输的邮件报文需采用 ASCII 进行编码。

网络的可用性是指（68)

(68)A.网络通信能力的大小　　B.用户用于网络维修的时间

C.网络的可靠性　　D.用户可利用网络时间的百分比

【答案】D

【解析】

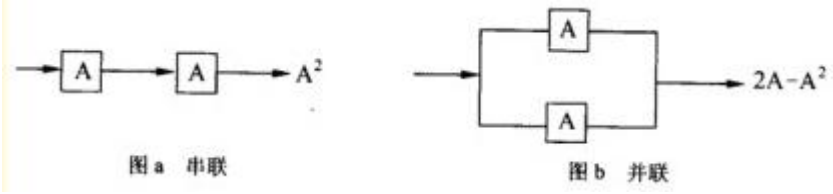
可用性是指网络系统、网络元素或网络应用对用户可利用的时间的百分比。有些应用对可用性很敏感，例如飞机订票系统若宕机一小时，就可能减少几十万元的票款；而股票交易系统如果中断运行一分钟，就可能造成几千万元的损失。实际上，可用性是网络元素可靠性的表现，而可靠性是指网络元素在具体条件下完成特定功能的概率。

如果用平均无故障时间（Mean Time Between Failure, MTBF)来度量网络兀素的故障率，则可用性 A 可表示为 MTBF 的函数：

$$A=\frac{\mathrm{MTBF}}{\mathrm{MTBF}+\mathrm{MTTR}}$$

其中 MTTR (Mean Time To Rqwir)为发生失效后的平均维修时间。由于网络系统由许多网络元素组成，.因此系统的可靠性不但与各个元素的可靠性有关，而且还与网络元素的组织形式有关。根据可靠性理论，由元素串并联组成的系统的可用性与网络元素的可用性之间的关系如下图所示。从图 a 中可以看出，若两个元素串联，则可用性减少。 例如两个 Modem 串联在链路的两端，若单个 Modem 的可用性 A=0.98，并假定链路其他部分的可用性为 1，贝 ij 整个链路的可用性 A=0.98x0.98=0.9604。从图 b 中可以看出，若两个元素并联，则可用

性增加。例如终端通过两条链路连接到主机，若一条链路失效，另外一条链路自动备份。假定单个链路的可用性A=0.98,则双链路的可用性A=2x 0.98-0.98x0.98=1.96-0.9604=0.9996。



图a 串联　　图b 并联

建筑物综合布线系统中的园区子系统是指（69）。

(69)A.由终端到信息插座之间的连线系统　　B.楼层接线间到工作区的线缆系统

C.各楼层设备之间的互连系统　　D.连接各个建筑物的通信系统

【答案】D

【解析】

结构化综合布线系统（Structure Cabling System)是基于现代计算机技术的通信物理平台，集成了语音、数据、图像和视频的传输功能，消除了原有通信线路在传输介质上的差别。结构化布线系统分为6个子系统：工作区子系统、水平子系统、干线子系统、设备间子系统、管理子系统和建筑群子系统。

工作区子系统（Work Location)。

工作区子系统是由终端设备到信息插座的整个区域。一个独立的需要安装终端设备的区域划分为一个工作区。工作区应支持电话、数据终端、计算机、电视机、监视器以及传感器等多种终端设备。

水平布线子系统（Horizontal)。

各个楼层接线间的配线架到工作区信息插座之间所安装的线缆属于水平子系统。水平子系统的作用是将干线子系统线路延伸到用户工作区。

管理子系统（Administration)。

管理子系统设置在楼层的接线间内，由各种交连设备（双绞线跳线架、光纤跳线架）以及集线器和交换机等交换设备组成，交连方式取决于网络拓扑结构和工作区设备的要求。交连设备通过水平布线子系统连接到各个工作区的信息插座，集线器或交换机与交连设备之间通过短线缆互连，这些短线被称为跳线。通过跳线的调整，可以在工作区的信息插座和交换机端口之间进行连接切换。

干线子系统（Backbone）

干线子系统是建筑物的主干线缆，实现各楼层设备间子系统之间的互连。干线子系统通常由垂直的大对数铜缆或光缆组成，一头端接于设备间的主配线架上，另一头端接在楼层接线间的管理配线架上。

设备间子系统（Equipment)。

建筑物的设备间是网络管理人员值班的场所，设备间子系统由建筑物的进户线、交换设备、电话、计算机、适配器以及保安设施组成，实现中央主配线架与各种不同设备（如PBX、网络设备和监控设备等）之间的连接。

建筑群子系统（Campus)。

建筑群子系统也叫园区子系统，它是连接各个建筑物的通信系统。大楼之间的布线方法有三种:一种是地下管道敷设方式,管道内敷设的铜缆或光缆应遵循电话管道和入孔的各种规定，安装时至少应预留1～2个备用管孔，以备扩充之用。第二种是直埋法，要在同一个沟内埋入通信和监控电缆，并应设立明显的地面标志。最后一种是架空明线，这种方法需要经常维护。

如果子网172.6.32.0/20被划分为子网172.6.32.0/26,则下面的结论中正确的是(70)。

(70)A.被划分为62个子网　　B.每个子网有64个主机

C.被划分为32个子网　　D.每个子网有62个主机地址

【答案】D

【解析】

子网172.6.32.0/20被划分为子网172.6.32.0/26,网络掩码增加了 6位，被划分成了64个子网，每个子网的主机ID部分为6位，可以提供主机地址个数为62。

At a basic level, cloud computing is simply a means of delivering IT resources as (71). Almost all IT resources can be delivered as a cloud service: applications, compute power, storage capacity, networking, programming tools, even communication services and collaboration (72).

Cloud computing began as large-scale Internet service providers such as Google, Amazon, and others built out their infrastructure. An architecture emerged: massively scaled, (73) distributed system resources, abstracted as virtual IT services and managed as continuously configured, pooled resources. In this

architecture, the data is mostly resident on (74) “somewhere on the Internet” and the application runs on both the “cloud servers” and the user’s browser.

Both clouds and grids are built to scale horizontally very efficiently. Both are built to withstand failures of (75) elements or nodes. Both are charged on a per-use basis. But while grids typically process batch jobs, with a defined start and end point, cloud services can be continuous. What’s more, clouds expand the types of resources available — file storage, databases, and Web services — and extend the applicability to Web and enterprise applications.

(71)A.hardware	B.entire	C.services	D.software
(72)A.computers	B.disks	C.machine	D.tools
(73)A.horizontally	B.vertically	C.inclined	D.decreasingly
(74)A.clients	B.middleware	C.servers	D.hard disks
(75)A.entire	B.individual	C.general	D.separate

【答案】C D A C B

【解析】本题考查对英语资料的阅读理解。

本段英文简要介绍云计算的概念。云计算主要是将资源看作云服务，包括应用程序、计算能力、存储容量、网络、编程工具，以及通信和协作工具。云计算最初由一些大的 Internet 服务提供商构建的基础设施而起步，其架构呈现出大规模、水平分布式系统资源、抽象的 1T 服务、管理持续配置、资源池等特性，数据大多存储于 Internet 上的某个地方的服务器上，应用程序运行于云服务器和用户浏览器中。

云和网格都针对有效的水平可扩展性，避免节点的单点失效对系统的影响，都按使用付费。它们的区别是网格通常是处理一批有明确定义起点和终点的作业，而云服务是可以连续不断的。另外，云扩展了资源的类型，包括文件存储、数据库和 Web 服务等，也将适用性扩展到 Web 和企业应用。

试题一

某学校欲开发图书管理系统，以记录图书馆所藏图书及其借出和归还情况，提供给借阅者借阅图书功能，提供给图书馆管理员管理和定期更新图书表功能。主要功能的具体描述如下：

处理借阅。借阅者要借阅图书时，系统必须对其身份（借阅者 ID)进行检查。通过与教务处维护的学生数据库、人事处维护的职工数据库中的数据进行比对，以验证借阅者 ID 是否合法。若合法，则检查借阅者在逾期未还图书表中是否有逾期未还图书，以及罚金表中的罚金是否超过限额。如果没有逾期未还图书并且罚金未超过限额，则允许借阅图书，更新图书表，并将借阅的图书存入借出图书表。借阅者归还所借图书时， 先由图书馆管理员检查图书是否缺失或损坏，若是，则对借阅者处以相应罚金并存入罚金表；然后，检查所还图书是否逾期，若是，执行“处理逾期”操作；最后，更新图书表，删除借出图书表中的相应记录。

维护图书。图书馆管理员查询图书信息；在新进图书时录入图书信息，存入图书表；在图书丢失或损坏严重时，从图书表中删除该图书记录。

处理逾期。系统在每周一统计逾期未还图书，逾期未还的图书按规则计算罚金，并记入罚金表，并给有逾期未还图书的借阅者发送提醒消息。借阅者在借阅和归还图书时，若罚金超过限额，管理员收取罚金，并更新罚金表中的罚金额度。

现采用结构化方法对该图书管理系统进行分析与设计，获得如图 1-1 所示的顶层数据流图和图 1-2 所示的 0 层数据流图。

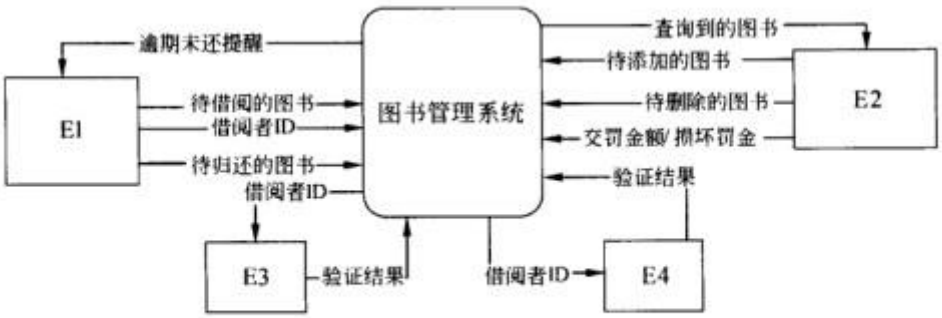


图 1-1　顶层数据流图

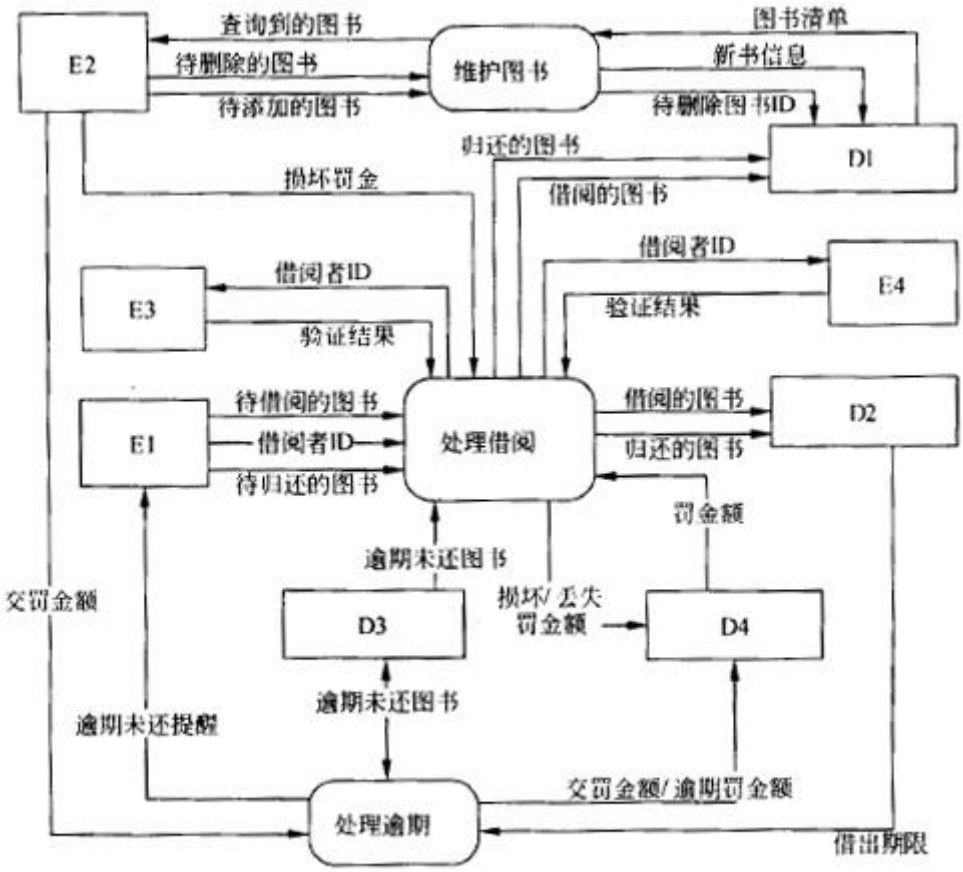


图 1-2　0 层数据流图

【问题 1】

使用说明中的词语，给出图 1-1 中的实体 E1～E4 的名称。

E1：借阅者 E2:图书管理员 E3/E4:学生数据库/职工数据库

注：E3 和 E4 不分顺序，但必须不同。

本题考查顶层 DFD。顶层 DFD一般用来确定系统边界，将待开发系统看作一个加工，图中只有唯一的一个处理和一些外部实体，以及这两者之间的输入输出数据流。题目要求根据描述确定图中的外部实体。分析题目中描述，并结合已经在顶层数据流图中给出的数据流进行分析。从题目的说明中可以看出：和系统的交互者包括图书管理员、借阅者两类人，图书管理员需要维护图书信息、得到查询所得的图书信息，借阅者提供借阅者 ID、借阅与归还的图书。还有通过与教务处维护的学生数据库、人事处维护的职工数据库中的数据进行比对以验证借阅者 ID 是否合法的两个数据库作为外部实体。

对应图 1-1 中数据流和实体的对应关系，可知 E1 为借阅者，E2 为图书管理员，E3 和 E4 为学生数据库和职工数据库。

【问题 2】

使用说明中的词语，给出图 1-2 中的数据存储 D1～D4 的名称

D1：图书表 D2:借出图书表 D3:逾期未还图书表 D4:罚金表

本题考查 0 层 DFD 中数据存储的确定。说明中描述维护图书信息主要存储或者更新图书表；借阅时需要检查逾期未还图书表，确定是否有逾期未还图书以及罚金表中的罚金限额，归还时出现缺失和损坏需要处以罚金并存入罚金表；借阅与归还图书时需要存入借出图书表和更新借出图书表。在处理逾期时需要将罚金记入罚金表，要检查和更新罚金限额。根据描述和图 1-2 中的数据存储的输入输出数据流提示，可知 D1 为图书表，D2 为借出图书表，D3 为逾期未还图书表，D4 为罚金表。

【问题 3】

在 DFD 建模时，需要对有些复杂加工（处理）进行进-步精化，绘制下层数据流图。针对图 1-2 中的加工“处理借阅”，在 1 层数据流图中应分解为哪些加工？（使用说明中的术语）

检查借阅者身份或检查借阅者 ID；检查逾期未还图书；检查罚金是否超过限额；借阅图书；归还图书。

本题考查将 0 层 DFD 中的处理进一步精化建模，绘制下层数据流图。从说明中对“处理借阅”的描述和图 1-2 可知，处理借阅需要检查借阅者身份、检查逾期未还图书、检查罚金是否超过限额、借阅图书和归还图书。描述中：检查所还图书是否逾期，若是，执行“处理逾期”操作。这里处理逾期明确说明是一个操作，而且在描述（3)中单独描述，图 1-2 中已经建模为单独一个处理，所以在本问题中仍然不分解。

【问题 4】

说明【问题 3】中绘制 1 层数据流图时要注意的问题。

保持父图与子图平衡。父图中某加工的输入输出数据流必须与它的子图的输入输出数据流在数量和名字上相同。如果父图的一个输入（或输出）数据流对应于子图中几个输入（或输出）数据流，而子图中组成这些数据流的数据项全体正好是父图中的这一个数据流，那么它们仍然算是平衡的。

本题考查在绘制下层数据流图时需要注意的问题。问题 3 明确给出是对复杂处理进行进一步精化，绘制下层数据流图，因此需要注意的问题是绘制下层数据流图时要保持父图与子图平

衡。父图中某加工的输入输出数据流必须与它的子图的输入输出数据流在数量和名字上相同。如果父图的一个输入（或输出）数据流对应于子图中几个输入（或输出）数据流，而子图中组成这些数据流的数据项全体正好是父图中的这一个数据流，那么它们仍然算是平衡的。

试题二

某医院拟开发一套住院病人信息管理系统，以方便对住院病人、医生、护士和手术等信息进行管理。

【需求分析】

(1)系统登记每个病人的住院信息，包括：病案号、病人的姓名、性别、地址、身份证号、电话号码、入院时间及病床等信息，每个病床有唯一所属的病 K 及病房，如表 2-1 所示。其中病案号唯一标识病人本次住院的信息。

表 2-1 住院登记表

病案号	071002286	姓名	张三	性别	男
身份证号	0102196701011234	入院时间	2011-03-03	病床号	052401
病房	0524 室	病房类型	三人间	所属病区	05 II 区

(2)在一个病人的一次住院期间，由一名医生对该病人的病情进行诊断，并填写一份诊断书，如表 2-2 所示。对于需要进行一次或多次手术的病人，系统记录手术名称、手术室、手术日期、手术时间、主刀医生及多名协助医生，每名医生在手术中的责任不同，如表 2-3 所示，其中手术室包含手术室号、楼层、地点和类型等信息

表 2-2 诊断书

诊断时间：2011 年 03 月

病案号	071002286	姓名	张三	性别	男	医生	李**
诊断							

(3)护士分为两类：病床护士和手术室护士。每个病床护士负责护理一个病区内的所有病人，每个病区由多名护士负责护理。手术室护士负责手术室的护理工作。每个手术室护士负责多个手术室，每个手术室由多名护士负责，每个护士在手术室中有不同的责任，并由系统记录其责任。

表 2-3 手术安排表

手术名称	***手术	病案号	071002286	姓名	张三	性别	男
手术室	032501	手术日期	2011-03-15	手术时间	8:30~10:30	主刀医生	李**
协助医生	王**(协助)，周**(协助)，刘**(协助)，高**(麻醉)						

【概念模型设计】

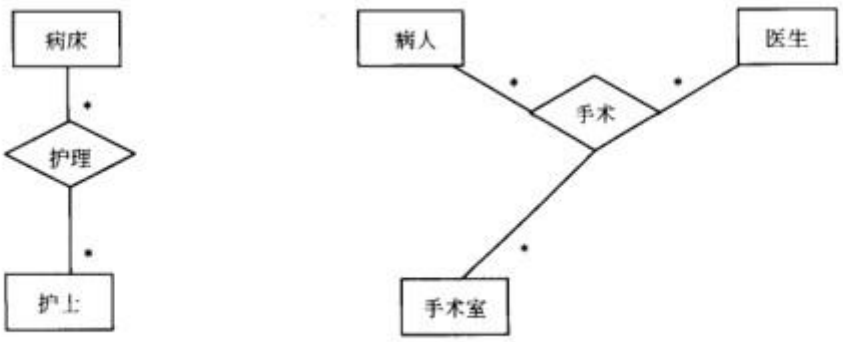


图 2-1　实体联系图

【概念模型设计】

根据概念模型设计阶段完成的实体联系图，得出如下关系模式（不完整）：

病床（病床号，病房，病房类型，所属病区）

护士（护士编号，姓名，类型，性别，级别）

病床护士（　　（1）　　）

手术室（手术室号，楼层，地点，类型）

手术室护士（　　（2）　　）

病人（　　（3）　　，姓名，性别，地址，身份证号，电话号码，入院时间）

医生（医生编号，姓名，性别，职称，所属科室）

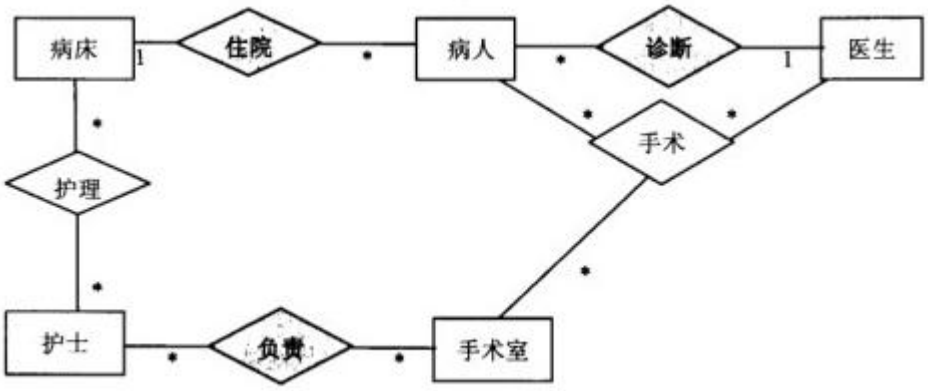
诊断书（　　（4）　　，诊断，诊断时间）

手术安排（病案号，手术室号，手术时间，手术名称）

手术医生安排（　　（5）　　，医生责任）

【问题 1】

补充图 2-1 中的联系和联系的类型。



本题考查数据库的概念结构设计，题目要求补充完整实体联系图中的联系和联系的类型。根据题目的需求描述可知，一名病人在一次住院期间对应一张病床，而一个病床可以有多名病人曾经住过。所以，病床实体和病人实体之间存在“住院”联系，联系的类 型为多对一，表示为*:1。

根据题目的需求描述可知，一名病人在一次住院期间，由一名医生做出诊断，并给出一份诊断书。所以，病人实体和医生实体之间存在“诊断”联系，联系的类型为多对多，表示为*:*:*。根据题目的需求描述可知，一名病人在一次住院期间可以进行多次手术，一次手术安排在一个手术室，由多名医生参与。所以，病人实体与医生实体和手术室实体三者之间'存在“手术”联系，三者之间联系的类型为多对多对多，表示为根据题目的需求描述可知，一名手术室护士负责多个手术室，每个手术室由多名护士负责。所以，护士实体和手术室实体之间存在“负责”联系，联系的类型为多对多，表示为*:*。

【问题 2】

根据图 2-1,将逻辑结构设计阶段生成的关系模式中的空（1)～（5)补充完整，并用下划线指出主键。

1：病区，护士号。2：手术室号，护士号，责任。3：病案号，病床号。4：病案号，医生编号。5：病案号，手术室号，手术时间，医生编号。

本题考查数据库的逻辑结构设计,题目要求补充完整各关系模式,并给出各关系模式的主键。

根据实体联系图和需求描述，每个病床护士负责护理一个病区内的所有病人，每个病区由多名护士负责护理。系统记录每个病床护士所负责护理的病区。所以，对于“病 床护士”关系模式需填写的属性为：病区，护士号。

根据实体联系图和需求描述，每个手术室护士负责多个手术室，每个手术室由多名 护士负责，每个护士在手术室中有不同的责任。因此，对于“手术室护士”关系模式，需填写的属性为：手术室号，护士号，责任。

根据实体联系图和需求描述，病案号唯一标识病人本次住院的信息。病人的住院信息包括病床信息。所以，对于“病人”关系模式需补充的属性为：病案号，病床号。

根据实体联系图和需求描述，一名病人在一次住院期间，由一名医生做出诊断，并给出一份诊断书。所以，对于“诊断”关系模式需补充的属性为：病案号，医生编号。

根据实体联系图和需求描述，一名病人在一次住院期间，可能需要进行一次或多次手术，每次手术安排在一间手术室，由多名医生(包括主刀医生)参与。所以，对于“手术医生安排”关系模式需补充的属性为：病案号，手术室号，手术时间，医生编号。

病床护士关系模式的主键：病区，护士号

手术室护士关系模式的主键：手术室号，护士号

病人关系模式的主键：病案号

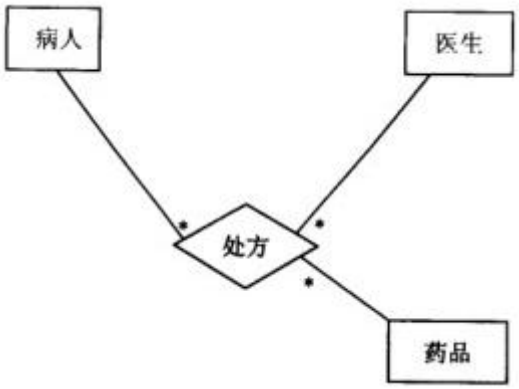
诊断书关系模式的主键：病案号

采购订单关系模式的主键：订单编码

手术医生安排关系模式的主键：病案号，手术室号，手术时间，医生编号码

【问题 3】

如果系统还需要记录医生给病人的用药情况，即记录医生给病人所开处方中药品的名称、用量、价格、药品的生产厂家等信息。请根据该要求，对图 2-1 进行修改，画出补充后的实体、实体间联系和联系的类型。



本题考查数据库的概念结构设计，根据新增的需求新增实体联系图中的实体及联系和联系的类型。

根据问题描述，系统需记录医生给病人开处方的药品信息，则需新增“药品”实体，并在病人实体与医生实体和药品实体三者之间存在“处方”联系，联系的类型是多对多对多(*:*:*)。

试题三

某网上购物平台的主要功能如下：

1 创建订单。顾客（Customer)在线创建订单（Order),主要操作是向订单中添加项目、从订单中删除项目。订单中应列出所订购的商品（Product)及其数量（quantities ）。

2 提交订单。订单通过网络来提交。在提交订单时，顾客需要提供其姓名（name)、 收货地址（address)以及付款方式（form of payment)(预付卡、信用卡或者现金)。为了制定送货计划以及安排送货车辆，系须确定订单量（volume)。除此之外，还必须记录每种商品的名称（name)、进价（cost price)、售价（sale price)以及单件商品的包装体积（cubic volume)。

3 处理订单。订单处理人员接收来自系统的订单；根据订单内容，安排配货，制定送货计划。在送货计划中不仅要指明发货日期（delivery date)，还要记录每个订单的限时发送要求（Delivery Time Window)。

4 派单。订单处理人员将已配好货的订单转交给派送人员。

5 送货/收货。派送人员将货物送到顾客指定的收货地址。当顾客收货时，需要在运货单（delivery slip)上签收。签收后的运货单最终需交还给订单处理人员。

6 收货确认。当订单处理人员收到签收过的运货单后，会和顾客进行一次再确认。 现采用面向对象方法开发上述系统，得到如图 3-1 所示的用例图和图 3-2 所示的类图。

【问题 1】

根据说明中的描述,给出图 3-1 中 A1～A3 所对应的参与者名称和 U1～U2 处所对应的用例名称

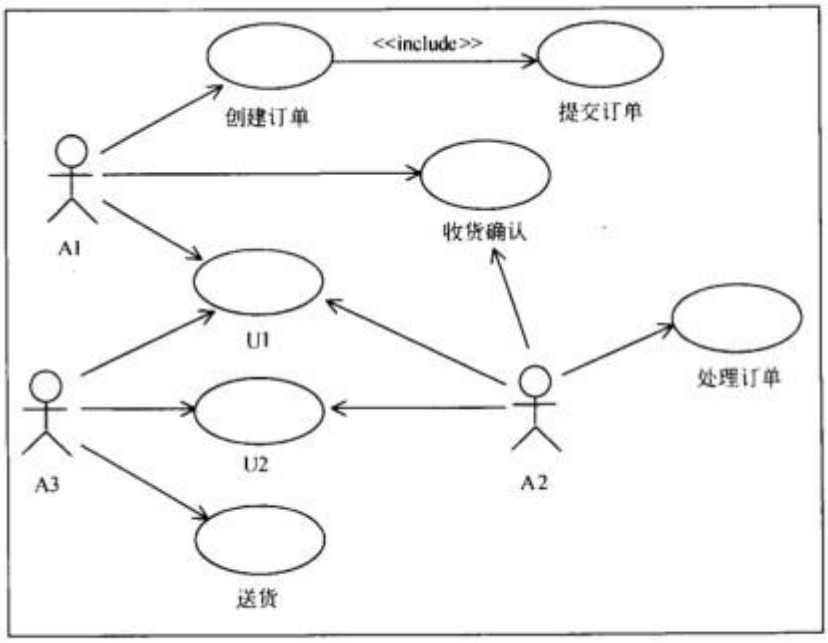


图 3-1　用例图

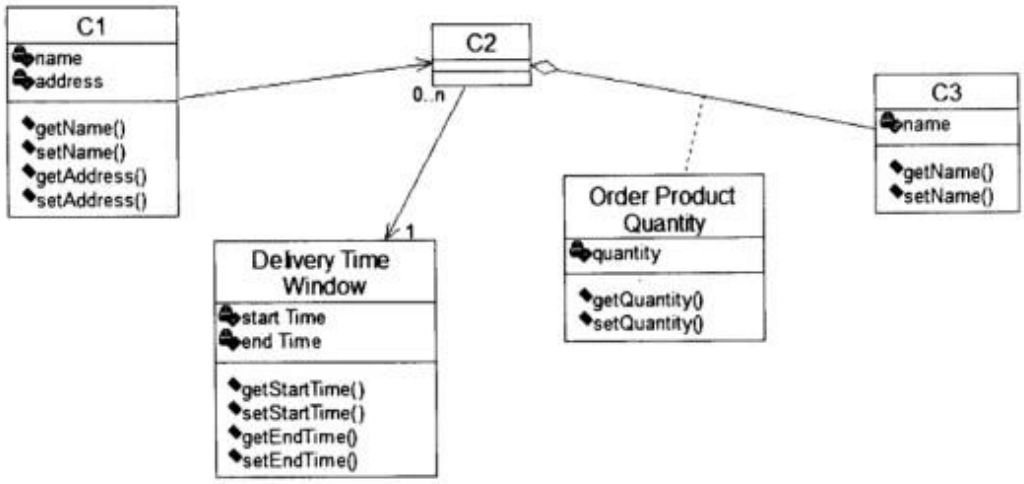


图 3-2　类图

A1：顾客 A2:订单处理人员 A3:派送人员 U1：收货 U2:派单

本题要求将图 3-1 所给出的用例图补充完整。用例图的构成要素有参与者(Actor)、用例(Use Case)以及用例之间的关系。由图 3-1 可知，题目中“网上购物平台”的用例模型中共有 3 个参与者、7 个用例。图中给出了其中的 5 个用例，需要补充所缺少的两个用例和三个参与者。解答此题时，应首先确定参与者，然后再找到与每个参与者对应的用例。

参与者表示要与本系统发生交互的一个角色单元。与系统交互的外部人员、角色、其他的计算机系统、物理实体等通常都可以看作是参与者。从“说明”中可以看出，在本题的描述中

出现了 “顾客”、“订单处理人员”、“派送人员”这三种角色。而用例图中恰好缺少了三个参与者，所以可以确定这三种角色就应该与 A1～A3 对应。接下来要确定的就是它们之间的对应关系，这就需要明确参与者与用例之间的关系了。解答时可以依据用例图逐个考查其中的参与者。

先从 A1 开始。A1 参与的用例分别为“创建订单”、“收货确认”以及需要补充的用例 U1。用例总是由参与者启动的。以“创建订单”、“收货确认”为关键词，在“说明” 中查找出现这两个关键词的语句的主语。这个主语就是该用例的参与者。由“说明”可见，A1 应该表示的是“顾客”。采用同样的方法，可以确定 A2 表示“订单处理人员”，A3 表示“派送人员”。由于缺少的用例 U1 和 U2 与参与者 A1 和 A3 相关，因此重点考查“说明”中出现了 “顾客”和“派送人员”的语句。比对“说明”和用例图可知，功能“派单”、“收货”没有出现在用例图中。“订单处理人员”和“派送人员”都参与了“派单”功能，所以 U2 处应该是用例“派单”。而“收货”功能涉及了三种角色：派送人员、顾客（两者完成货物的交接）、订单处理人员（收回顾客签字后的运货单，此时一次完整的收货活动才算结束），所以 U1 处应该是“收货”。

【问题 2】

根据说明中的描述，给出图 3-2 中 C1～C3 所对应的类名以及（1）～（4）处所对应的多重度（类名使用说明中给出的英文词汇）

C1：Customer C2：Order C3：Product

（1）1　　（2）0..n 或 0..*　　（3）0..n 或 0..*　　（4）1..n 或 1..*

本题考查的是类图建模。解题的重点在于根据类图中提供的类、类的属性以及类之 间的关联关系推断出需要补充的类。

先看类 C1，C1 的关键属性都已经给出了。由“说明”可知，属性 address 表示的是“收货地址”。而收货地址的最初始来源是顾客，所以类 C1 代表的应该是 Customer(顾客)。

另外，从“说明”中可知，“订单”和“商品”是“网上购物平台”系统的关键概念，所以应该在类图上有相应的类来表示。由此可以推断出，C2 和 C3 就应该与这两个概念相关。观察类图，在 C2 和 C3 之间存在着一个关联类 Order Product，而 C3 和 C2 之间又存在着一个聚集关系，C3 是构成 C2 的部分对象。根据“说明”，订单中包含了订购的商品，所以 C2 应

该是 Order（订单），C3 应该是 Product（商品）。

三个类都确定之后，下面来识别它们之间的多重度。

首先顾客（C1)和订单（C2)之间的关联，一名顾客可以在线创建多个订单，但是一个订单只能由一名顾客来创建。所以（1）处应填写“1”，(2)处应填写“0..n”。

订单（C2)和商品（C3)，一个订单上可以订购多件商品，而一件商品可以出现在一个或多个订单中，也可能没有任何顾客订购某种商品。因此 C2 和 C3 之间是一种多对多联系。在图 3-2 中，采用了关联类的方法来表示多对多联系。这里（3）处应填写“0..n”，

(4)处应填写“1..n”。这里下限 1 表示一个订单中至少应包含一件商品。

【问题 3】

根据说明中的描述,将类C2和C3的属性补充完整(属性名使用说明中给出的英文词汇)。

C2：volume、delivery date、form of payment

C3: cubic volume、cost price、sale price

本题考查类的关键属性的识别。由“说明”中给出的描述可知，类 C2 的属性至少应包括 volume、delivery date、form of payment；类 C3 的属性至少应包括 cubic volume、cost price、sale price。

试题四

用两台处理机 A 和 B 处理 n 个作业。设 A 和 B 处理第 i 个作业的时间分别为叫 ai 和 bi。由于各个作业的特点和机器性能的关系，对某些作业，在 A 上处理时间长，而对某些作业在 B 上处理时间长。一台处理机在某个时刻只能处理一个作业，而且作业处理是不可中断的，每个作业只能被处理一次。现要找出一个最优调度方案，使得 n 个作业被这两台处理机处理完毕的时间（所有作业被处理的时间之和）最少。

算法步骤：

（1）确定候选解上界为最短的单台处理机处理所有作业的完成时间 m，

$$m = \min\left(\sum_{i=1}^{n} a_i, \sum_{i=1}^{n} b_i\right)。$$

（2）用 P(x，y，k)=1 表示前 k 个作业可以在 A 用时不超过 x 且在 B 用时不超过 y 时间内处理完成，则 p(x，y，k)=p(x-ak，y，k-1) || p(x，y-bk，k-1)（||表示逻辑或操作）。

（3）得到最短处理时间为 min(max(x，y))。

【C 代码】

下面是该算法的 C 语言实现。

（1）常量和变量说明

n：作业数

m：候选解上界

a：数组，长度为 n，记录 n 个作业在 A 上的处理时间，下标从 0 开始

b：数组，长度为 n，记录 n 个作业在 B 上的处理时间，下标从 0 开始

k：循环变量

p：三维数组，长度为(m+1)* (m+1)* (n+1)

temp：临时变量

max：最短处理时间

（2）C 代码

```
#include <stdio.h>
int n, m;
int a[60], b[60], p[100][100][60];
void read(){  /*输入 n、a、b，求出 m，代码略*/  }
void schedule(){          /*求解过程*/
    int x,y,k;
    for(x = 0; x <= m; x++){
        for (y = 0; y <= m; y++){
            ____(1)____;
            for(k = 1; k <= n; k++)
                p[x][y][k] = 0;
        }
    }
    for(k = 1;k <= n;k++){
        for (x = 0;x <= m;x++){
            for(y = 0; y <= m; y++){
                if(x - a[k - 1] >= 0)  ____(2)____;
                if(____(3)____)  p[x][y][k] = (p[x][y][k] || p[x][y - b[k -

                    1]][k - 1]);
            }
        }
    }
}
void write(){          /*确定最优解并输出*/
    int x,y, temp,max = m;
    for(x = 0;x <= m;x++){
        for (y = 0;y <= m;y++){
            if( ____(4)____ ){
                temp = ____(5)____ ;
                if(temp < max)     max = temp;
            }
        }
    }
    printf("\n%d\n", max);
}
void main(){    read();    schedule();    write();    }
```

【问题 1】

根据以上说明和 C 代码，填充 C 代码中的空（1）～（5）。

（1）p[x][y][0] = 1

（2）p[x][y][k] = p[x – a[k – 1]][y][k – 1]

（3）y – b[k – 1] >= 0

（4）p[x][y][n] == 1 或 p[x][y][n] 或 p[x][y][n] != 0

（5）(x >= y)? x:y

schedule 函数计算 p 数组中元素的值。第一个三重 for 循环进行 p 数组的初始化，当 k = 0

时，p[i][j][k]=1;而当 k 关 0 时，p[i][j][k]=0,因此(1)处应填 “p[x][y][0]=1 ”。在接下来的三重 for 循环中，自底向上计算 p[i][j]M 的值，计算根据题中定义来进行，(2)处应填 “p[x][y][k]=p[x-a[k-1]][y][k-1]”。对应于上一行的条件，(3)处应填“y- b[k—1]>=0”。

write 函数中，已经计算出 p 数组元素的值，因此此时每一个 p[i][j][n] = 1 表示一个可行解，因次（4）处应填“p[x][y][n]=1”或等价的表达式。其中该解对应的处理时间为 i 和 j 中的较大值，因此(5)处应填“(x>=y)?x:y”。在所有的可行解中，确定最小的处理时间即为问题的最优解。

【问题 2】

根据以上 C 代码，算法的时间复杂度为（6)(用 0 符号表示)

$O(m^2n)$

根据上述 C 代码，函数 schedule 有两处三重 for 循环，时间复杂度为 0(m2n)。函数 write 有一处两重 for 循环，时间复杂度为 0(m2)。因此整个算法的时间复杂度为 0(m2n + m2)，由于 m2 在数量级上小于 m2n,因此算法的时间复杂度可以表示为 0(m2n)。

【问题 3】

考虑 6 个作业的实例，各个作业在两台处理机上的处理时间如表 4-1 所示。该实例的最优解为（7),最优解的值（即最短处理时间）为(8)。最优解用（x1,x2,x3,x4,x5,x6）表示，其中若第 i 个作业在 A 上处理，则 Xi=1，否则 Xi=2。如（1，1，1，1，2,2)表示作业 1，2,3 和 4 在 A 上处理，作业 5 和 6 在 B 上处理。

表 4-1

	作业 1	作业 2	作业 3	作业 4	作业 5	作业 6
处理机 A	2	5	7	10	5	2
处理机 B	3	8	4	11	3	4

(7)(1,1,2,2,1,1)

(8)15

根据题意和算法，可以得到若作业 1、2、5 和 6 在处理机 A，而作业 3 和 4 在处理 机 B 上处理，可以得到最优解。此时在处理机 A 上的处理时间为 14,而在处理机 B 上 的处理时间为 15,因此最优解的值，即最短的处理时间为 15,而最优解为（1,1,2,2,1,1)。

试题五

某咖啡店售卖咖啡时，可以根据顾客的要求在其中加入各种配料，咖啡店会根据所加入的配料来计算费用。咖啡店所供应的咖啡及配料的种类和价格如下表所示。

咖啡	价格/杯（￥）	配料	价格/份（￥）
蒸馏咖啡（Espresso）	25	摩卡（Mocha）	10
深度烘焙咖啡（DarkRoast）	20	奶泡（Whip）	8

现采用装饰器（Decorator）模式来实现计算费用的功能，得到如图 5-1 所示的类图。

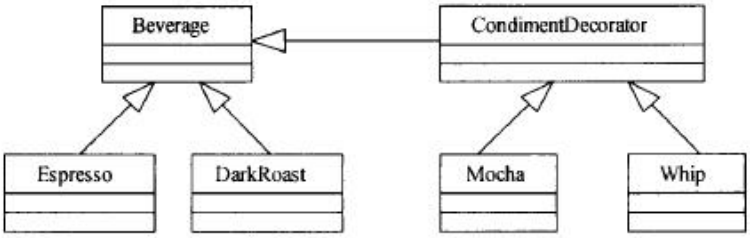


图 5-1　类图

【C++代码】

```
#include <iostream>
#include <string>
using namespace std;
const int ESPRESSO_PRICE = 25;
const int DRAKROAST_PRICE = 20;
const int MOCHA_PRICE = 10;
const int WHIP_PRICE = 8;
class Beverage {    // 饮料
 (1)  :    string description;
public:
 (2)  () {  return description;  }
    (3)  ;
};
class CondimentDecorator : public Beverage {    // 配料
protected:
    (4)  ;
};
class Espresso : public Beverage {    // 蒸馏咖啡
public:
    Espresso() { description = "Espresso";  }
    int cost() { return ESPRESSO_PRICE; }
};
class DarkRoast : public Beverage {    // 深度烘焙咖啡
public:
```

【问题 1】

```
        DarkRoast() { description = "DardRoast"; }
        int cost() { return DRAKROAST_PRICE; }
    };
    class Mocha : public CondimentDecorator {  // 摩卡
    public:
        Mocha(Beverage* beverage) {  this->beverage = beverage; }
        string getDescription()  {  return beverage->getDescription()+ ",
Mocha";  }
        int cost() { return MOCHA_PRICE + beverage->cost(); }
    };
    class Whip : public CondimentDecorator {  // 奶泡
    public:
        Whip(Beverage* beverage) {  this->beverage = beverage; }
        string getDescription()  {  return beverage->getDescription()+ ",
Whip";  }
        int cost() { return WHIP_PRICE + beverage->cost(); }
    };

    int main() {
        Beverage* beverage = new DarkRoast();
        beverage = new Mocha( (5) );
        beverage= new Whip( (6) );
        cout << beverage->getDescription() << "￥" << beverage->cost() << endl;
        return 0;
    }
```

编译运行上述程序，其输出结果为：

```
DarkRoast, Mocha, Whip ￥38
```

1.protected

2.virtual string getDescription

3.virtualintcost() = 0

4.Beverage* beverage

5.beverage

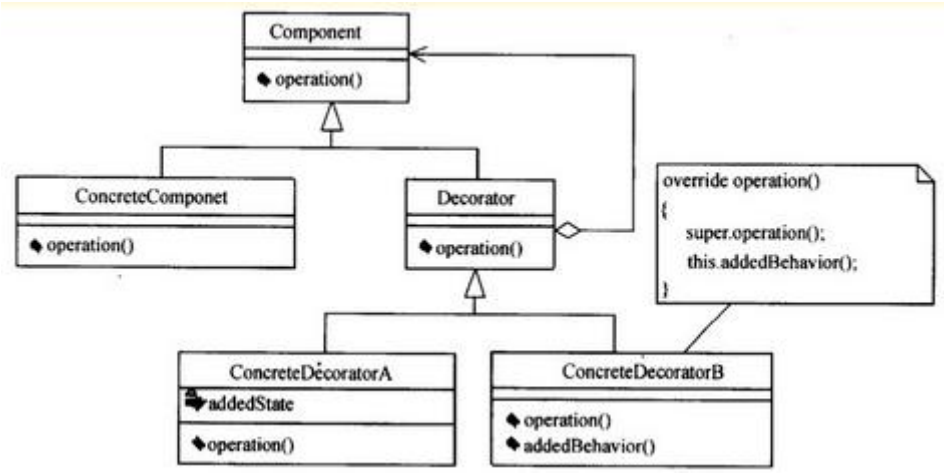
本题考查装饰器（Decorator)模式的概念及应用。

Decorator 模式动态地给一个对象添加一些额外的职责，就增加功能来说，装饰模式提供了比继承更有弹性的替代方案。

Decorator 模式的优点是有效避免了使用继承方式扩展对象功能而带来的灵活性差、子类无限制扩展的问题；装饰者与被装饰者之间虽然都是同一类型，但是它们彼此是完全独立并可以独立任意改变的。

Decorator 模式的适用场合是：想透明并且动态地给对象增加新的职责；给对象增加的职责，在未来存在增加或减少的可能。

Decorator 模式的类图如下所示：



题目利用 Decorator 模式来计算各种配料组合的咖啡的价格。Beverage 相当于抽象的 Component 类，最终要计算出 Beverage 的价钱。Espresso 和 DarkRoast 是 4 个具体的组件，代表一种咖啡类型。Macha 和 Whip 是配料装饰者，可以添加到不同类型的咖啡中。CondimentDecorator 相当于 Decorator,是装饰者共同实现的接口。在本题中，确定装饰者共同实现的接口是什么，是一个重要的考查点。

下面来分析程序。

第(1)空要求给出类 Beverage 的数据成员 description 的访问控制。C++语言提供了 3 种访问控制 private、public 和 protected。private 成员只能在定义它的类中或其友元访问；public 成员可以被任意访问；protected 成员只能在定义它的类及其子类中访问。从程序上下文可知，在 Beverage 的子类 Espresso 和 DarkRoast 中都需要访问 description, 所以第(1)空应填“protected”。

第(2)、(3)空要求确定 Beverage 中提供的公共接口。解答时应全面阅读程序，通过子类的代码来推断父类所提供的接 U 是什么。

首先来观察 Beverage 的两个子类。在 Espresso 和 DarkRoat 中都出现的成员函数是 cost，其功能是给出咖啡的价格；而在这两个类中，cost 的实现代码又不相同。这就提示我们这里采用了 C++语言的动态多态机制，意味着需要在这两个类的父类中定义一个虚拟函数，其函数原型就应该是 intcost0。现在回到第(2)、(3)空。第(2)空给出了成员函数的实现体，说明(2)处的成员函数是与 description 相关的。也就是说，第(2) 空处的成员函数不可能是 cost。这样就可以确定第(3)空应该是 cost 成员函数。仔细观察第(3)空，这条语句的结束符是“;”，即第(3)空中只能出现函数的接口声明。只有接口声明的虚函数，在 C++中只能是纯虚拟函数。所以第(3)空应填写“virtualint cost() = 0”。

如何来确定第(2)空？在类 Espresso 和 DarkRoast 中已经找不到相关信息了，我们考查剩余的类。

类 CondimentDecorator 是 Beverage 的子类，那么第(2)和(3)空处的成员函数都会被它继承。而类 Mocha 和 Whip 又是 CondimentDecorator 的子类，第(2)和(3)空处的成员函数同样也会被这两个类继承。观察 Mocha 和 Whip 的代码，可以发现在这两个子类中也重置了纯虚拟函数 cost。除此之外还可以发现，在这两个子类中都出现了成员函数 string getDescription()，并且其在两个类中的实现代码也不相同。这说明，getDescription 也应该是父类中所定义的虚拟函数。从子类沿着继承路径追袖到父类，可以确定第(2)空就应该是虚拟函数 getDescription 最初的定义之处。所以第(2)空应填写 “virtual string getDescription”。

第（4)空考查的是类 CondimentDecorator 中定义的数据成员，要求确定其类型和名称。由于声明成了 protected,因此在其子类 Mocha 和 Whip 中可以找到答案。在这两个类中都出现了 “this->beverage”表达式，说明 beverage 应该是该类的数据成员。但在这两个类中都没有定义数据成员，很显然，这个数据成员应该是从父类继承而来的。所以第（4)空应填“Beverage* beverage”。

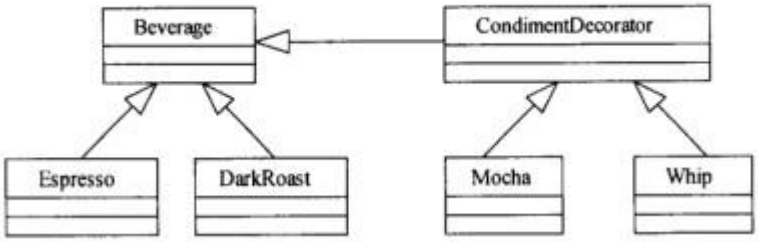
第（5)和（6)空考查的是 Decorator 模式的使用方法。从主函数可知，其基本过程是现制作一杯烘焙咖啡（DarkRoast),然后再向这杯咖啡中加入 1 份 Mocha 和 1 份 Whip,最后这杯咖啡的价格应该是烘焙咖啡的价格+1 份 Mocha 的价格+1 份 Whip 的价格。(5)和（6)处都应该填写“beverage”。

试题六

某咖啡店售卖咖啡时，可以根据顾客的要求在其中加入各种配料，咖啡店会根据所加入的配料来计算费用。咖啡店所供应的咖啡及配料的种类和价格如下表所示。

咖啡	价格/杯（￥）	配料	价格/份（￥）
蒸馏咖啡（Espresso）	25	摩卡（Mocha）	10
深度烘焙咖啡（DarkRoast）	20	奶泡（Whip）	8

现采用装饰器（Decorator）模式来实现计算费用的功能，得到如图 6-1 所示的类图。



【Java 代码】

```
import java.util.*;
 (1)  class Beverage {      // 饮料
  String description = "Unknown Beverage";
  public  (2)  () {      return description;  }
  public  (3)  ;
}
abstract class CondimentDecorator extends Beverage {   // 配料
  (4)  ;
}
class Espresso extends Beverage {     // 蒸馏咖啡
  private final int ESPRESSO_PRICE = 25;
  public Espresso() {   description = "Espresso";  }
  public int cost() {    return ESPRESSO_PRICE;  }
}

class DarkRoast extends Beverage {    // 深度烘焙咖啡
  private final int DARKROAST_PRICE = 20;
  public DarkRoast() {  description = "DarkRoast";  }
  public int cost() {    return DARKROAST_PRICE;      }
}
class Mocha extends CondimentDecorator {   // 摩卡
  private final int MOCHA_PRICE = 10;
  public Mocha(Beverage beverage) {
     this.beverage = beverage;
  }
  public String getDescription() {
     return beverage.getDescription() + ", Mocha";
  }
```

【问题 1】

```
  public int cost() {
     return MOCHA_PRICE + beverage.cost();
  }
}
class Whip extends CondimentDecorator {  // 奶泡
  private final int WHIP_PRICE = 8;
  public Whip(Beverage beverage) {  this.beverage = beverage; }
  public String getDescription()  {
      return beverage.getDescription()+ ", Whip";
  }
  public int cost() { return WHIP_PRICE + beverage.cost(); }
}

public class Coffee {
  public static void main(String args[]) {
    Beverage beverage = new DarkRoast();
    beverage = new Mocha( __(5)__ );
    beverage = new Whip( __(6)__ );
    System.out.println(beverage.getDescription() + " ￥" +
    beverage.cost());
  }
}
```

编译运行上述程序，其输出结果为：

```
DarkRoast, Mocha, Whip ￥38
```

1.abstract.2.String getDescription.3.abstract int cost().4.Beverage beverage.5.beverage.6.beverage

本题考查装饰器（Decorator)模式的概念及应用。

Decorator 模式动态地给一个对象添加一些额外的职责，就增加功能来说，装饰模式提供了比继承更有弹性的替代方案。

Decorator 模式的优点是有效避免了使用继承方式扩展对象功能而带来的灵活性差、子类无限制扩展的问题；装饰者与被装饰者之间虽然都是同一类型，但是它们彼此是完全独立并可以独立任意改变的。

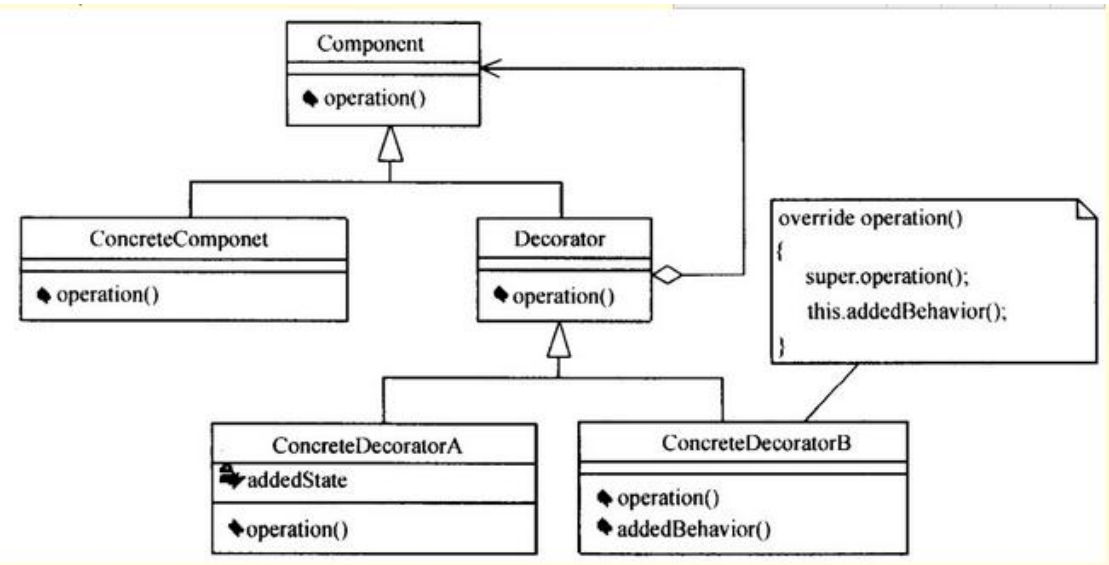
Decorator 模式的适用场合是：想透明并且动态地给对象增加新的职责；给对象增加的职责，在未来存在增加或减少的可能。

Decorator 模式的类图如下所示：

题目利用 Decorator 模式来计算各种配料组合的咖啡的价格。Beverage 相当于抽象的

Component 类，最终要计算出 Beverage 的价钱。Espresso 和 DarkRoast 是 4 个具体的组件，代 k 一种咖啡类型。Macha 和 Whip 是配料装饰者，可以添加到不同类型的咖啡中。

CondimentDecorator 相当于 Decorator,是装饰者共同实现的接口。在本题中，确定装饰者共同实现的接口是什么，是一个重要的考查点。



下面来分析程序。

Decorator 模式中的 Component 通常都用抽象类来实现。所以第(1)空应填写 “abstract”。第（2)、(3)空要求确定 Beverage 中提供的公共接口。解答时应全面阅读程序，通过子类的代码来推断父类所提供的接口是什么。

首先来观察 Beverage 的两个子类。在 Espresso 和 DarkRoat 中都出现的成员函数是 cost,其功能是给出咖啡的价格；而在这两个类中，cost 的实现代码又不相同。这意味着需要在这两个类的父类中定义一个抽象函数，其函数原型就应该是 int cost()。现在回到第(2)、(3)空。第(2)空给出了成员函数的实现体，说明(2)处的成员函数是与 description 相关的。也就是说，第(2)空处的成员函数不可能是 cost。这样就可以确定第(3)空应该是 cost 成员函数。所以第（3)空应填写“abstract int cost()’’。

如何来确定第（2)空？在类 Espresso 和 DarkRoast 中已经找不到相关信息了，我们考查剩余的类。

类 CondimentDecorator 是 Beverage 的子类，那么第（2)和（3)空处的成员函数都会被它继承。而类 Mocha 和 Whip 又是 CondimentDecorator 的子类，第（2)和（3)空处的成员函数同样也会被这两个类继承。观察 Mocha 和 Whip 的代码，可以发现在这两个子类中也重置了抽象函数 cost。除此之外还可以发现，在这两个子类中都出现了成员函数 String

getDescription0。从子类沿着继承路径追溯到父类，可以确定第（2)空就应 该是成员函数getDescription 最初的定义之处。所以第（2)空应填写“ String getDescription’’。