三总线结构的计算机总线系统由（1）组成。

(1)A.CPU 总线、内存总线和 IO 总线　　B.数据总线、地址总线和控制总线

C.系统总线、内部总线和外部总线　　D.串行总线、并行总线和 PCI 总线

【答案】B

【解析】本题考查计算机系统基础知识。

总线上传输的信息类型分为数据、地址和控制，因此总线由数据总线、地址总线和控制总线组成。

计算机采用分级存储体系的主要目的是为了解决（2）问题。

(2)A.主存容量不足　　B.存储器读写可靠性

C.外设访问效率　　D.存储容量、成本和速度之间的矛盾

【答案】D

【解析】本题考查计算机系统基础知识。

计算机系统中，高速缓存一般用 SRAM,内存一般用 DRAM,外存一般采用磁存储器。SRAM 的集成度低、速度快、成本高。DRAM 的集成度高，但是需要动态刷新。磁存储器速度慢、容量大、价格便宜。因此，不同的存储设备组成分级存储体系，来解决速度、存储容量和成本之间的矛盾。

属于 CPU 中算术逻辑单元的部件是（3）。

(3)A.程序计数器　B.加法器　C.指令寄存器　D.指令译码器

【答案】B

【解析】本题考查计算机系统基础知识。

程序计数器、指令寄存器和指令译码器都是 CPU 中控制单元的部件，加法器是算术逻辑运算单元的部件。

内存按字节编址从 A5000H 到 DCFFFH 的区域其存储容量为（4）。

(4)A.123KB　B.180KB　C.223KB　D.224KB

【答案】D

【解析】本题考查计算机系统基础知识。

从地址 A5000H 到 DCFFFH，存储单元数目为 37FFFH (即 224*1024)个，由于是字节编址，从而得到的存储容景为 224KB。

以下关于 RISC 和 CISC 的叙述中，不正确的是（5）。

(5)A. RISC 通常比 CISC 的指令系统更复杂

B. RISC 通常会比 CISC 配置更多的寄存器

C. RISC 编译器的子程序库通常要比 CISC 编译器的子程序库大得多

D. RISC 比 CISC 更加适合 VLSI 工艺的规整性要求

【答案】A

【解析】本题考查计算机系统基础知识。

计算机工作时就是取指令和执行指令。一条指令往往可以完成一串运算的动作，但却需要多个时钟周期来执行。随着需求的不断增加，设计的指令集越来越多，为支持这些新增的指令，计算机的体系结构会越来越复杂，发展成 CISC 指令结构的计算机。而在 CISC 指令集的各种指令中，其使用频率却相差悬殊，大约有 20%的指令会被反复使用，占整个程序代码的 80%。而余下的 80%的指令却不经常使用，在程序中常用的只占 20%，显然这种结构是不太合理的。

RISC 和 CISC 在架构上的不同主要有：

①在指令集的设计上，RISC 指令格式和长度通常是固定的（如 ARM 是 32 位的指令)、且寻址方式少而简单、大多数指令在一个周期内就可以执行完；CISC 构架下的指令长度通常是可变的、指令类型也很多、一条指令通常要若干周期才可以执行完。由于指令集多少与复杂度上的差异，使 RISC 的处理器可以利用简单的硬件电路设计出指令解码功能，这样易于流水线的实现。相对的 CISC 则需要通过只读存储器里的微码来进行解码，CISC 因为指令功能与指令参数变化较大，执行流水线作业时有较多的限制。

②RISC 架构中只有载入和存储指令可以访问存储器，数据处理指令只对寄存器的内容进行操作。为了加速程序的运算，RISC 会设定多组的寄存器，并且指定特殊用途的寄存器。CISC 构架则允许数据处理指令对存储器进行操作，对寄存器的要求相对不高。

Flynn 分类法基于信息流特征将计算机分成 4 类，其中（6）只有理论意义而无实例。

(6)A. SISD　　B. MISD　　C. SIMD　　D. MIMD

【答案】B

【解析】本题考查计算机系统基础知识。

Flynn 主要根据指令流和数据流来分类，分为四类：

①单指令流单数据流机器（SISD）

SISD 机器是一种传统的串行计算机，它的硬件不支持任何形式的并行计算，所有的指令都是串行执行，并且在某个时钟周期内，CPU 只能处理一个数据流。因此这种机器被称作单指令流单数据流机器。早期的计算机都是 SISD 机器。

②单指令流多数据流机器（SIMD）

SIMD 是采用一个指令流处理多个数据流。这类机器在数字信号处理、图像处理以及多媒体信息处理等领域非常有效。

Intel 处理器实现的 MMXTM、SSE (Streaming SIMD Extensions)、SSE2 及 SSE3 扩展指令集，都能在单个时钟周期内处理多个数据单元。也就是说人们现在用的单核计算机基本上都属于 SIMD 机器。

③多指令流单数据流机器（MISD）

MISD 是采用多个指令流来处理单个数据流。在实际情况中，采用多指令流处理多数据流才是更有效的方法。因此 MISD 只是作为理论模型出现，没有投入实际应用。

④多指令流多数据流机器（MIMD）

MIMD 机器可以同时执行多个指令流，这些指令流分别对不同数据流进行操作。例如，intel 和 AMD 的双核处理器就属于 MIMD 的范畴。

网络系统中，通常把（7）置于 DMZ 区。

(7)A.网络管理服务器 B.Web 服务器 C.入侵检测服务器 D.财务管理服务器

【答案】B

【解析】本题考查防火墙的基础知识。

DMZ 是指非军事化区，也称周边网络，可以位于防火墙之外也可以位于防火墙之内。非军事化区一般用来放置提供公共网络服务的设备，这些设备由于必须被公共网络访问，所以无法提供与内部网络主机相等的安全性。

分析四个备选答案，Web 服务器是为一种为公共网络提供 Web 访问的服务器；网络管理服务器和入侵检测服务器是管理企业内部网和对企业内部网络中的数据流进行分析的专用设备，一般不对外提供访问；而财务服务器是一种仅针对财务部门内部访问和提供服务的设

备，不提供对外的公共服务。

以下关于拒绝服务攻击的叙述中，不正确的是（8）。

(8)A.拒绝服务攻击的目的是使计算机或者网络无法提供正常的服务

B.拒绝服务攻击是不断向计算机发起请求来实现的

C.拒绝服务攻击会造成用户密码的泄漏

D.DDoS 是一种拒绝服务攻击形式

【答案】C

【解析】本题考查拒绝服务攻击的基础知识。

拒绝服务攻击是指不断对网络服务系统进行干扰，改变其正常的作业流程，执行无关程序使系统响应减慢直至瘫痪，从而影响正常用户的使用。当网络服务系统响应速度减慢或者瘫痪时，合法用户的正常请求将不被响应，从而实现用户不能进入计算机网络系统或不能得到相应的服务的目的。

DDoS 是分布式拒绝服务的英文缩写。分布式拒绝服务的攻击方式是通过远程控制大量的主机向 H 标主机发送大量的干扰消息的一种攻击方式。

（9）不是蠕虫病毒。

(9)A.熊猫烧香　　B.红色代码　　C.冰河　　D.爱虫病毒

【答案】C

【解析】本题考查计算机病毒的基础知识。

“蠕虫”(Worm)是一个程序或程序序列。它利用网络进行复制和传播，传染途径是通过网络、移动存储设备和电子邮件。最初的蠕虫病毒定义是在 DOS 环境下，病毒发作时会在屏幕上出现一条类似虫子的东西，胡乱吞吃屏幕上的字母并将其改形，蠕虫病毒因此而得名。常见的蠕虫病毒有红色代码、爱虫病毒、熊猫烧香、Nimda 病毒、爱丽兹病毒等。

冰河是木马软件，主要用于远程监控，冰河木马后经其他人多次改写形成多种变种，并被用于入侵其他用户的计算机的木马程序。

甲公司接受乙公司委托开发了一项应用软件，双方没有订立任何书面合同。在此情形下（10）享有该软件的著作权。

(10)A.甲公司　　B.甲、乙公司共同　　C.乙公司　　D.甲、乙公司均不

【答案】A

【解析】

委托开发软件著作权关系的建立，通常由委托方与受委托方订立合同而成立。委托开发软件关系中，委托方的责任主要是提供资金、设备等物质条件，并不直接参与开发软件的创作开发活动。受托方的主要责任是根据委托合同规定的目标开发出符合条件的软件。关于委托开发软件著作权的归属，《计算机软件保护条例》第十二条规定："受他人委托开发的软件，其著作权的归属由委托者与受委托者签定书面协议约定，如无书面协议或者在协议中未作明确约定，其著作权属子受委托者。"根据该条的规定，确定委托开发的软件著作权的归属应当掌握两条标准：

①委托开发软件系根据委托方的要求，由委托方与受托方以合同确定的权利和义务的关系而进行开发的软件，因此软件著作权归属应当作为合同的重要条款予以明确约定。对于当事人已经在合同中约定软件著作权归属关系的，如事后发生纠纷，软件著作权的归属仍应当根据委托开发软件的合同来确定。

②对于在委托开发软件活动中，委托者与受委托者没有签定书面协议，或者在协议中未对软件著作权归属作出明确的约定，其软件著作权属于受委托者，即属于实际完成软件的开发者。

甲、乙软件公司于 2013 年 9 月 12 日就其财务软件产品分别申请"大堂"和"大唐"商标注册。两财务软件相似，且经协商双方均不同意放弃使用其申请注册的商标标识。此情形下，（11）获准注册。

(11)A."大堂"　　B."大堂"与"大唐"都能

C."大唐"　　D.由甲、乙抽签结果确定谁能

【答案】D

【解析】

我国商标注册采取"申请在先"的审查原则，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，商标主管机关根据申请时间的先后，决定商标权的归属，申请在先的人可以获得注册。对于同日申请的情况，使用在先的人可以获得注册。如果同日使用或均未使用，则采取申请人之间协商解决，协商不成的，由各申请人抽签决定。类似商标是指在同一种或类似商品上用作商标的文字、图形、读音、含义或文字与图形的整体结构上等要素大体相同的商标，即易使消费者对商品的来源产生误认的商标。甲、乙两公

司申请注册的商标，“大堂”与“大唐”读音相同、文字相近似，不能同时获准注册。在协商不成的情形下，由甲、乙公司抽签结果确定谁能获准注册。

以下媒体中（12）是表示媒体，（13）是表现媒体。

(12)A.声音　　B.声音编码　　C.超声波　　D.喇叭

(13)A.声音　　B.声音编码　　C.超声波　　D.喇叭

【答案】B　D

【解析】本题考查多媒体基础知识。

传输媒体指传输表示媒体的物理介质，如电缆、光缆、电磁波等；表示媒体指传输感觉媒体，如声音、图像等的中介媒体，即用于数据交换的编码，如文本编码、声音编码和图像编码等；表现媒体是指进行信息输入和输出的媒体，如键盘、鼠标、话筒以及显示器、打印机、喇叭等；存储媒体指用于存储表示媒体的物理介质，如硬盘、光盘等。

声音信号的两个基本参数是幅度和频率。幅度是指声波的振幅，通常用动态范围表示，一般用分贝（dB)为单位来计量。频率是指声波每秒钟变化的次数，用 Hz 表示。人们把频率小于 20Hz 声波信号称为亚音信号(也称次音信号);频率范围为 20Hz～20kHz 的声波信号称为音频信号；高于 20kHz 的信号称为超音频信号（也称超声波)。

显示深度、图像深度是图像显示的重要指标。当（14）时，显示器不能完全反映数字图像电使用的全部颜色。

(14)A.显示深度=图像深度　　B.显示深度>图像深度

C.显示深度≥图像深度　　D.显示深度<图像深度

【答案】D

【解析】本题考查考生多媒体基础知识。

图像深度是指存储每个像素所用的位数，它是用来度量图像的色彩分辨率的，即确定彩色图像的每个像素可能有的颜色数，或者确定灰度图像的每个像素可能有的灰度级数。显示深度是显示器上每个点用于显示颜色的二进制位数。使用显示器显示数字图像时，应当使显示器的显示深度大于或等于数字图像的深度，这样显示器就可以完全反映数字图像中使用的全部颜色。

以下关于结构化开发方法的叙述中，不正确的是（15）。

(15)A.总的指导思想是自顶向下，逐层分解

B.基本原则是功能的分解与抽象

C.与面向对象开发方法相比，更适合于大规模、特别复杂的项目

D.特别适合于数据处理领域的项目

【答案】C

【解析】本题考查结构化开发方法的基础知识。

结构化开发方法由结构化分析、结构化设计和结构化程序设计构成，是一种面向数据流的开发方法。结构化方法总的指导思想是自顶向下、逐层分解，基本原则是功能的分解与抽象。它是软件工程中最早出现的开发方法，特别适合于数据处理领域的问题，但是不适合解决大规模的、特别复杂的项目，而且难以适应需求的变化。

模块 A、B 和 C 都包含相同的 5 个语句，这些语句之间没有联系。为了避免重复把这 5 个语句抽取出来组成一个模块 D，则模块 D 的内聚类型为（16）内聚。

（16）A.功能　　B.通信　　C.逻辑　　D.巧合

【答案】D

【解析】本题考查软件设计的相关知识。

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来进行度量。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巧合内聚，指一个模块内的各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。

某个项目在开发时采用了不成熟的前沿技术，由此而带来的风险属于（17）风险。

（17）A.市场　　B.技术　　C.经济　　D.商业

【答案】B

【解析】本题考查软件开发风险的基本概念。

风险是一种具有负面后果的、人们不希望发生的事件。从不同的角度可以对风险进行不同的分类。如从风险涉及的范围，风险可以分为项目风险、技术风险和商业风险等。技术风险涉及设计方案、实现、接口、验证以及维护等方面的问题。此外，包括需求规格说明的不确定性、技术的不确定性、技术的陈旧以及采用不成熟的前沿技术等可能会带来技术风险。技术风险威胁着开发产品的质量和交付产品的时间。

属于面向对象、解释型程序设计语言的是（18）。

（18）A.XML　　B.Python　　C.Prolog　　D.C++

【答案】B

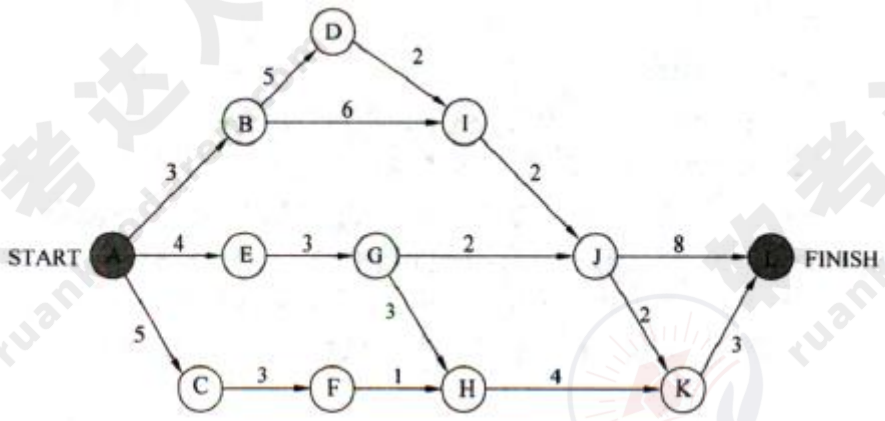
【解析】本题考查程序语言基础知识。

XML（可扩展标记语言）是标准通用标记语言的子集，是一种用于标记电子文件使其具有结构性的标记语言。

Python 是一种面向对象、解释型计算机程序设计语言。

Prolog 是逻辑型程序设计语言。

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，边的权重表示活动的持续时间，则里程碑（19）在关键路径上。活动 GH 的松弛时间是（20）。



(19)A.B　　B.E　　C.C　　D.K

(20)A.0　　B.1　　C.2　　D.3

【答案】A D

【解析】本题考查活动图的基础知识。

根据关键路径法，计算出关键路径为 A—B—D—I—J—L，关键路径长度为 20,因此里程碑 B 在关键路径上，而里程碑 E、C 和 K 不在关键路径上。包含活动 GH 的最长路径是 A—E—G—H—K—L，长度为 17，因此该活动的松弛时间为 20-17=3。

算术表达式“(a-b)*(c+d)”的后缀式是（21）。

(21)A. ab-cd+* B. abcd-*+ C. ab-*cd+ D. ab-c+d*

【答案】A

【解析】本题考查程序语言基础知识。

后缀式（逆波兰式）是波兰逻辑学家卢卡西维奇发明的一种表示表达式的方法。这种表示方式把运算符写在运算对象的后面，例如把 a+b 写成 ab+,所以也称为后缀式。算术表达式“(a-b)*(c+d)“的后缀式是“ab-cd+*”。

将高级语言源程序翻译成机器语言程序的过程中，常引入中间代码。以下关于中间代码的叙述中，不正确的是（22）。

(22)A. 中间代码不依赖于具体的机器 B. 使用中间代码可提高编译程序的可移植性

C. 中间代码可以用树或图表示 D. 中间代码可以用栈和队列表示

【答案】D

【解析】本题考查程序语言基础知识。

从原理上讲，对源程序进行语义分析之后就可以直接生成目标代码，但由于源程序与目标代码的逻辑结构往往差别很大，特别是考虑到具体机器指令系统的特点，要使翻译一次到位很困难，而且用语法制导方式机械生成的目标代码往往是繁琐和低效的，因此有必要设计一种中间代码，将源程序首先翻译成中间代码表示形式，以利于进行与机器无关的优化处理。由于中间代码实际上也起着编译器前端和后端分水岭的作用，所以使用中间代码也有助于提高编译程序的可移植性。常用的中间代码有后缀式、三元式、 四元式和树（图）等形式。

假设系统采用 PV 操作实现进程同步与互斥。若 n 个进程共享两台打印机，那么信号量 S 的取值范围为（23）。

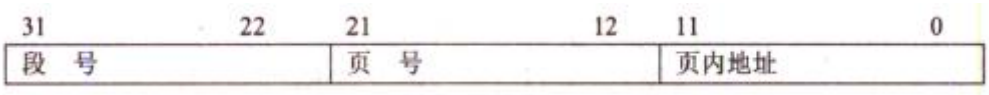
(23)A. -2～n B. -(n-1)～1 C. -(n-1)～2 D. -(n-2)～2

【答案】D

【解析】本题考查操作系统 PV 操作方面的基本知识。

系统采用 PV 操作实现进程同步与互斥，若有 n 个进程共享两台打印机，那么信号量 S 初值应为 2。当第 1 个进程执行 P(S)操作时，信号量 S 的值减去 1 后等于 1;当第 2 个进程执行 P(S)操作时，信号量 S 的值减去 1 后等于 0;当第 3 个进程执行 P(S)操作时，信号量 S 的值减去 1 后等于-1;当第 4 个进程执行 P(S)操作时，信号量 S 的值减去 1 后等于-2;……;当第 n 个进程执行 P(S)操作时，信号量 S 的值减去 1 后等于-(n-2)。可见，信号量 S 的取值范围为-(n-2)～2。

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（24）。



(24)A. 最多可有 2048 个段，每个段的大小均为 2048 个页，页的大小为 2K

B. 最多可有 2048 个段，每个段最大允许有 2048 个页，页的大小为 2K

C. 最多可有 1024 个段，每个段的大小均为 1024 个页，页的大小为 4K

D. 最多可有 1024 个段，每个段最大允许有 1024 个页，页的大小为 4K

【答案】D

【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 12 位，212=4096,即 4K;页号部分的地址长度是 10 位，每个段最大允许有 210=1024 个页；段号部分的地址长度是 10 位，210=1024,最多可有 1024 个段。

假设磁盘块与缓冲区大小相同，每个盘块读入缓冲区的时间为 10μs，由缓冲区送至用户区的时间是 5μs，系统对每个磁盘块数据的处理时间为 2μs。若用户需要将大小为 10 个磁盘块的 Doc1 文件逐块从磁盘读入缓冲区，并送至用户区进行处理，那么采用单缓冲区需要花费的时间为（25）μs；采用双缓冲区需要花费的时间为（26）μs。

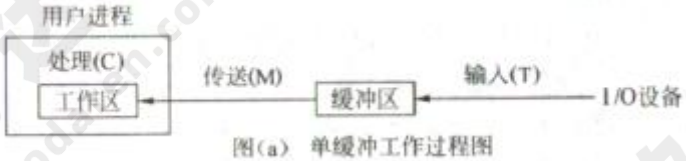
(25)A. 100　　B. 107　　C. 152　　D. 170

(26) A. 100　　B. 107　　C. 152　　D. 170

【答案】C　B

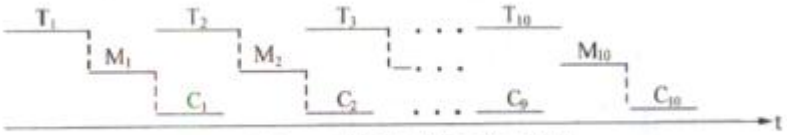
【解析】

在块设备输入时，假定从磁盘把一块数据输入到缓冲区的时间为 T，缓冲区中的数据传送到用户工作区的时间为 M，而系统处理（计算）的时间为 C，如图（a)所示。



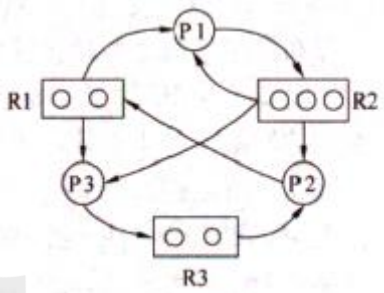
图(a) 单缓冲工作过程图

当第一块数据送入用户工作区后，缓冲区是空闲的，可以传送第二块数据。这样第一块数据的处理 C1 与第二块数据的输入 T2 是可以并行的，如图（b)所示，依次类推。系统对每一块数据的处理时间为：Max(C，T)+M。因为当 T>C 时，处理时间为 M+T;当 T<C 时，处理时间为本题每一块数据的处理时间为 10+5=15，Doc1 文件的处理时间为 15*10+2。



图(b) 单缓冲并行工作示意图

在如下所示的进程资源图中，（27）；该进程资源图是（28）。



(27)A. P1、P2、P3 都是阻塞节点　　B.P1 是阻塞节点、P2、P3 是非阻塞节点

C.P1、P2 是阻塞节点、P3 是非阻塞节点　D.P1、P2 是非阻塞节点、P3 是阻塞节点

(28)A.可以化简的，其化简顺序为 P1→P2→P3

B.可以化简的，其化简顺序为 P3→P1→P2

C.可以化简的，其化简顺序为 P2→P1→P3

D.不可以化简的，因为 P1、P2、P3 申请的资源都不能得到满足

【答案】C　B

【解析】

图中 R1 资源只有 2 个，P2 进程申请该资源得不到满足，故 P2 进程是阻塞节点；R2 资

源只有 3 个，P1 申请该资源得不到满足，故 P1 进程也是阻塞节点；R3 资源只有 2 个，分配给 P1 进程 1 个，P3 申请 1 个该资源可以得到满足，故 P3 是非阻塞节点。

以下关于增量模型的叙述中，正确的是（29）。

(29)A.需求被清晰定义　　B.可以快速构造核心产品

C.每个增量必须要进行风险评估　　D.不适宜商业产品的开发

【答案】B

【解析】本题考查软件开发过程的基础知识。

软件开发过程以系统需求作为输入，以要交付的产品作为输出，涉及活动、约束和资源使用的一系列工具和技术。瀑布模型、快速原型化模型、增量模型、螺旋模型等都是典型的软件开发过程模型。增量模型是 Mills 等于 1980 年提出来的。在使用该模型开发软件时，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，并能够完成特定的功能。

其优点包括能在较短时间内向用户提交可完成一些有用的工作产品；用户有充裕的时间来学习和适应不断增加的产品功能；项目失败风险较低；优先级最高的服务首先交付，然后再逐步增加新的构件，这样最重要的构件被测试得最充分。在四个选项中，只有选项 B 是描述增量模型的，要求需求被清晰定义是瀑布模型的一个典型特点，风险评估是螺旋模墦的特点。在当今市场竞争激烈的条件下，用增量模型可以快速的交付一部分产品，是适于商业产品的开发的。

以下关于 CMM 的叙述中，不正确的是（30）。

(30)A.CMM 是指软件过程能力成熟度模型

B.CMM 根据软件过程的不同成熟度划分了 5 个等级，其中，1 级被认为成熟度最高，5 级被认为成熟度最低

C.CMMI 的任务是将已有的几个 CMM 模型结合在一起，使之构造成为“集成模型”

D.采用更成熟的 CMM 模型，一般来说可以提高最终产品的质量

【答案】B

【解析】本题考查软件过程的基础知识。

CMM (Capability Maturity Model)是指软件过程能力成熟度模型，该模型按照软件过程的不同成熟度划分了 5 个等级，1 级被认为成熟度最低，5 级则为成熟度最高。一般来说，

采用更成熟的软件过程模型，往往可以得到更高质量的软件产品。1997 年美国卡内基•梅隆大学软件工程研究所 SEI 将已有的几个 CMM 模型结合在一起，构造成“集成模型”即 CMMI (Capability Maturity Model Integration) 。

在 ISO/IEC 软件质量模型中，可靠性是指在规定的一段时间内和规定的条件下，软件维持在其性能水平的能力；其子特性不包括（31）。

(31)A.成熟性　　B.容错性　　C.易恢复　　D.可移植性

【答案】D

【解析】本题考查软件质量的基础知识。

ISO/IEC9126 软件质量模型由三个层次组成：第一层是质量特性，第二层是质量子特性，第三层是度量指标。可靠性是一个重要的质量特性，其子特性包括成熟性、容错性和易恢复性。

在软件开发过程中，系统测试阶段的测试目标来自于（32）阶段。

(32)A.需求分析　　B.概要设计　　C.详细设计　　D.软件实现

【答案】A

【解析】本题考查软件测试的基础知识。

软件测试的基本目标是为了发现软件中的错误，但软件测试分为几个不同的阶段，每个阶段的侧重点是有所不同的。单元测试主要是发现程序代码中的问题，针对详细设计和软件实现阶段的工作进行的；集成测试验证系统模块是否能够根据系统和程序设计规格说明的描述进行工作，即模块以及模块之间的接口的测试：而系统测试则是验证系统是否确实执行需求规格说明中描述的功能和非功能要求，因此测试目标在需求分析阶段就已经定义。

以下关于文档的叙述中，不正确的是（33）。

(33)A.项目相关人员可以通过文档进行沟通　　B.编写文档会降低软件开发的效率

C.编写高质量文档可以提高软件开发的质量　　D.文档是软件的不可或缺的部分

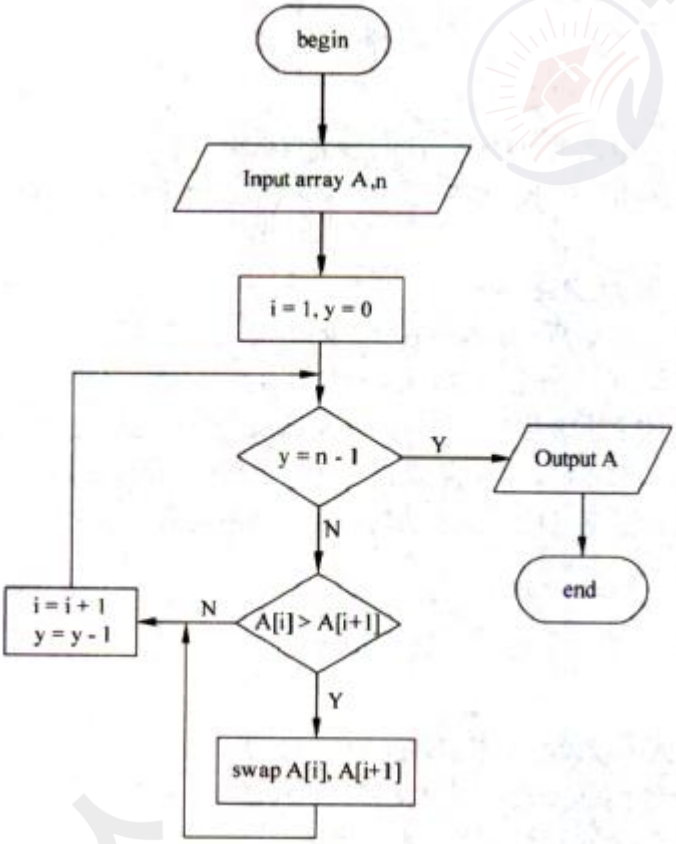
【答案】B

【解析】本题考查软件文档的相关知识。

软件由程序、数据和相关文档构成，因此文档是软件不可或缺的重要组成部分。软件开发各个阶段都需要撰写相关文档，如开发计划、需求分析文档、设计文档等，这些文档是开

发人员之间以及和其他人员之间进行沟通的重要依据，高质量的文档对于提高软件开发质量具有重要的意义。尽管在开发过程中编写文档需要占用开发时间，但是相对于没有文档而言，编写文档使得开发人员对各个阶段的工作都进行周密思考，全盘权衡，从而减少返工。并且可以在开发早期发现错误和不一致性，便于及时加以纠正，因此可以提高软件开发效率。

下图所示的程序流程图中有（34）条不同的简单路径。采用 McCabe 度量法计算该程序图的环路复杂性为（35）。



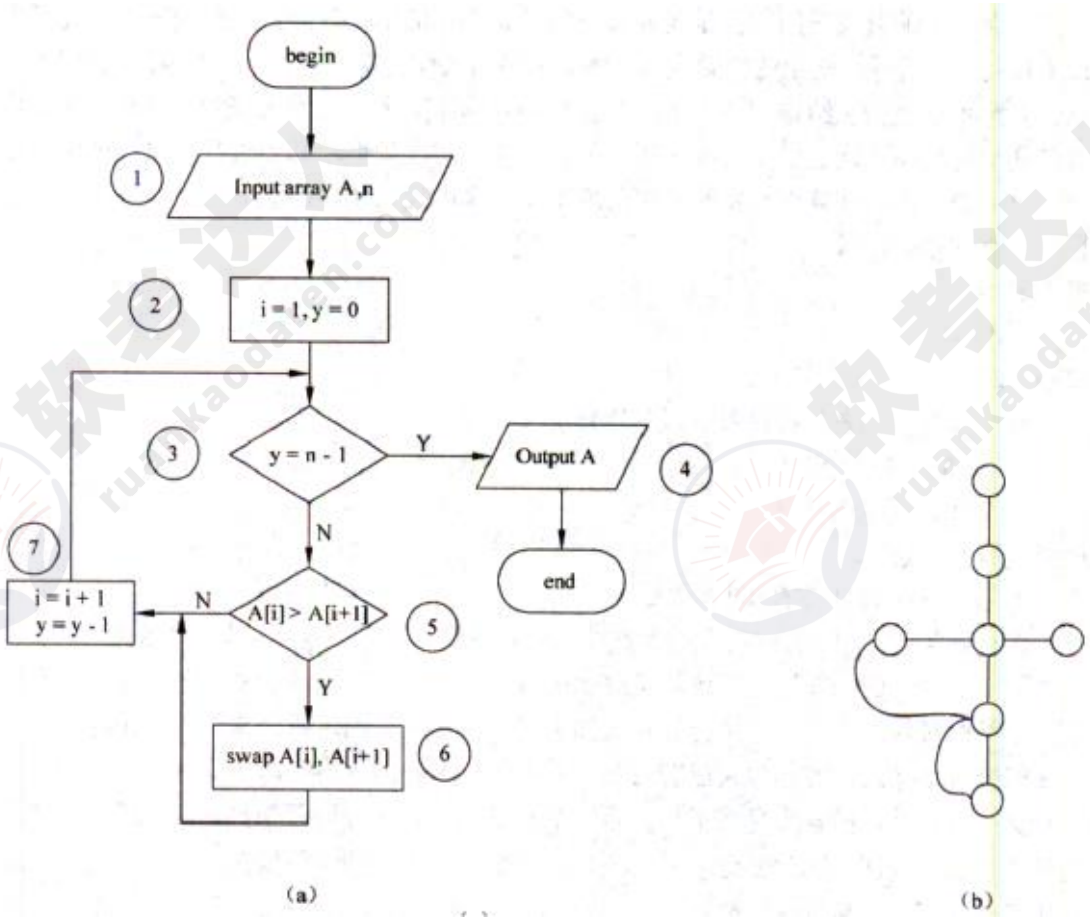
(34) A. 3　　B. 4　　C. 5　　D. 6

(35) A. 3　　B. 4　　C. 5　　D. 6

【答案】A　A

【解析】本题考查程序结构和复杂性的基本概念和相关知识。

重新画出上述流程图，给相关的操作加上编号，并给出对应的等价图，如下图（a）和（b）所示。



从图(a)可知，共有 3 条简单路径，即①②③④；①②③⑤⑦③；①②③⑤⑥⑦③。 根据图（b)计算程序复杂性，得到 n=7， e=8,因此复杂性为 e-n+2=3。

以下关于软件维护和可维护性的叙述中，不正确的是（36）。

(36)A. 软件维护要解决软件产品交付用户之后运行中发生的各种问题

B. 软件的维护期通常比开发期长得多，其投入也大得多

C. 进行质量保证审查可以提高软件产品的可维护性

D. 提高可维护性是在软件维护阶段考虑的问题

【答案】D

【解析】本题考查软件维护的相关知识。

软件产品在交付给用户之后，就进入了维护阶段。在该阶段针对系统改变所做的任何工作，都属于维护活动。软件维护期通常比开发期要长得多，根据统计数据一般项目花费 1 到 2 年的开发时间,但是需要额外的 5 到 6 年的维护时间。维护的成本也比开发成本高得多，另外一些企业数据表明，平均 39%的工作量花在开发上，其余的在维护上。软件可维护性是指在给定的条件下，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。通

过多个方面来提高软件产品的可维护性，其中进行质量保证审查是一个重要的手段。在软件开发的各个阶段都需要考虑提高软件产品的可维护性，而不仅仅是在软件维护阶段。

类（37）之间存在着一般和特殊的关系。

(37)A.汽车与轮船　　B.交通工具与飞机　　C.轮船与飞机　　D.汽车与飞机

【答案】B

【解析】本题考查面向对象的基本知识。

在进行类设计时，有些类之间存在一般和特殊关系，即一些类是某个类的特殊情况，某个类是一些类的一般情况，这就是继承关系。继承是类之间的一种关系，在定义和实现一个类的时候，可以在一个已经存在的类（一般情况）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，并加入若干新的内容，即子类比父类更加具体化。交通工具是泛指各类交通工具，而汽车、飞机和轮船分别都是具体的交通工具类，且具有自己的特性。因此交通工具是汽车、飞机和轮船类的一般情况，分别与汽车、轮船和飞机存在一般与特殊的关系。

多态分为参数多态、包含多态、过载多态和强制多态四种不同形式，其中（38）多态在许多语言中都存在，最常见的例子就是子类泛型化。

(38)A.参数　　B.包含　　C.过载　　D.强制

【答案】B

【解析】本题考查面向对象的基本知识。

面向对象系统中，在收到消息时，对象要予以相应。多态（polymorphism)是不同的对象收到同一消息可以进行不同的响应，产生完全不同的结果，用户可以发送一个通用的消息，而实现细节则由接收对象自行决定，使得同一个消息就可以调用不同的方法，即一个对象具有多种形态。Cardelli 和 Wegner 将多态分为 4 种不同的形式：参数多态、包含多态、过载多态和强制多态。其中参数多态是应用比较广的多态，包含多态在许多语言中都存在，最常见的例子就是子类型化。过载多态是同一个名字在不同的上下文中所代表的含义。

在面向对象程序设计语言中，对象之间通过（39）方式进行通信。以下关于好的面向对象程序设计语言的叙述中，不正确的是（40）。

(39) A.消息传递　　B.继承　　C.引用　　D.多态

(40)A.应该支持被封装的对象　　B.应该支持类与实例的概念

C.应该支持通过指针进行引用　　D.应该支持继承和多态

【答案】A　C

【解析】本题考查面向对象的基本知识。

在面向对象技术中，继承关系是一种模仿现实世界中继承关系的一种类之间的关系，是超类（父类）和子类之间共享数据和方法的机制。面向对象语言中对象通过消息传递的方式进行相互通信。在继承的支持下，不同的对象收到同一消息可以进行不同的响应，并且会产生完全不同的结果，这种现象称为多态。

好的面向对象程序设计语言一般应该支持被封装的对象、类与实例的概念、支持继承和多态等面向对象技术中的概念。

UML 中有 4 种事物：结构事物、行为事物、分组事物和注释事物。类、接口、构建属于（41）事物；依附于一个元素或一组元素之上对其进行约束或解释的简单符号为（42）事物。

(41)A.结构　B.行为　C.分组　D.注释

(42) A.结构　B.行为　C.分组　D.注释

【答案】A　D

【解析】本题考查统一建模语言（UML)的基本知识。

UML 是面向对象软件的标准化建模语言，由三个要素构成：UML 的基本构造块、支配这些构造块如何放置在一起的规则和运用与整个语言的一些公共机制。UML 的词汇表包含三种构造块：事物、关系和图。事物是对模型中最具有代表性的成分的抽象；关系把事物结合在一起；图聚集了相关的事物。UML 中有 4 种事物：结构事物、行为事物、分组事物和注释事物。结构事物是 UML 模型中的名词，它们通常是模型的静态部分，描述概念或物理元素。结构事物包括类（class)、接口（interface)、协作（collaboration)、用例（usecase)、主动类（activeclass)、构件（component)、制品（artifact)和结点（node)。行为事物是 UML 模型的动态部分，它们是模型中的动词，描述了跨越时间和空间的行为。行为事物包括：交互（interaction)、 状态机（state machine)和活动（activity)。分组事物是 UML 模型的组织部分，是一些由模型分解成的“盒子”。在所有的分组事物中最主要的分组事物是包（package)。注释事物是 UML 模型的解释部分。这些注释事物用来描述、说明和标注模型的任何元素。注解（note)是一种主要的注释事物。注解是一个依附于一个元素或者一组元素之上，对它进行约束或解释的简单符号。

一组对象以定义良好但是复杂的方式进行通信，产生的相互依赖关系结构混乱且难以理解。采用（43）模式，用一个中介对象来封装一系列的对象交互，从而使各对象不需要显式地相互引用，使其耦合松散，。而且可以独立地改变它们之间的交互。此模式与（44）模式是相互竞争的模式，主要差别是：前者的中介对象封装了其它对象间的通信，而后者通过引入其它对象来分布通信。

(43)A.解释器(Interpreter)　　B.策略(Strategy)

C.中介者(Mediator)　　D.观察者(Observer)

(44)A.解释器(Interpreter)　　B.策略(Strategy)

C.中介者(Mediator)　　D.观察者(Observer)

【答案】C　D

【解析】本题考查设计模式的基本概念。

解释器（Interpreter)设计模式是给定一个语言，定义它的文法的一种表示，并定义一个解释器，这个解释器使用该表示来解释语言中的句子。策略（Strategy)设计模式定义一系列算法，把它们一个个封装起来，并且使它们可相互替换。这一模式使得算法可独立于它的客户而变化。中介者（Mediator)用一个中介对象来封装一系列的对象交互。中介者使各对象不需要显式地相互引用，从而使其耦合松散，而且可以独立地改变它们之间的交互。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。

以上四种设计模式都是行为设计模式。行为设计模式大多注重封装变化，当一个程序的某个方面的特征经常发生改变时，这些模式就定义一个封装这个方面的对象。这样，当该程序的其他部分依赖于这个方面时，它们都可以与此对象协作。这些模式通常定义一个抽象类来描述这些封装变化的对象，并且通常该模式依据这个对象来命名。例如：一个 Strategy 对象封装一个算法，一个 Mediator 对象封装对象间的协议。Mediator 和 Observer 是相互竞争的模式，之间的差别是：Observer 通过引入 Observer 和 Subject 对象来分布通信，而 Mediator 对象则封装了其他对象间的通信。

UML 图中，一张交互图显示一个交互。由一组对象及其之间的关系组成，包含它们之间可能传递的消息。（45）不是交互图。

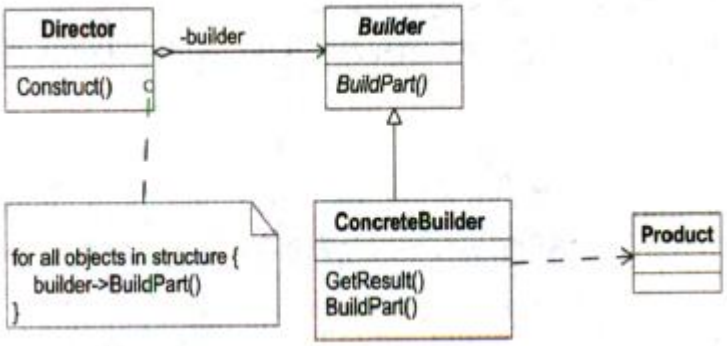
(45)A.序列图　　B.对象图　　C.通信图　　D.时序图

【答案】B

【解析】本题考查统一建模语言（UML)的基本知识。

UML 中提供了多种建模系统的图，体现系统的静态方面和动态方面。对象图（object diagram)展现了某一时刻一组对象以及它们之间的关系。对象图描述了在类图中所建立的事物的实例的静态快照，给出系统的静态设计视图或静态进程视图。序列图(sequence diagram)是场景（scenario)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。通信图（communication diagram)强调收发消息的对象的结构组织。时序图（Timing Diagram)关注沿着线性时间轴、生命线内部和生命线之间的条件改变，描述对象状态随着时间改变的情况，很像示波器，适合分析周期和非周期性任务。交互概览图(Interaction Overview Diagram)是 UML 2.0 新增的交互图之一，它是活动图的变体，描述业务过程中的控制流概览，软件过程中的详细逻辑概览，以及将多个图进行连接，抽象掉了消息和生命线。序列图、通信图、交互概览图和时序图均被称为交互图。

图所示为（46）设计模式，适用于（47）。



(46)A.抽象工厂(Abstract Factory)　　B.生成器(Builder)

C.工厂方法(Factory Method)　　D.原型(Prototype)

(47)A.一个系统要由多个产品系列中的一个来配置时

B.当一个类希望由它的子类来指定它所创建的对象时

C.当创建复杂对象的算法应该独立于该对象的组成部分及其装配方式时

D.当一个系统应该独立于它的产品创建、构成和表示时

【答案】B　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，描述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。

抽象工厂（AbstractFactory)模式提供一个创建一系列相关或相互依赖对象的接口，而无需指定他们具体的类。抽象工厂模式适用于一个系统要独立于它的产品的创建、组合和表示时；一个系统要由多个产品系列中的一个来配置时；当要强调一系列相关的产品对象的设计以便进行联合使用时：当提供一个产品类库，而只想显示它们的接口而不是实现时。

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：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（Factory Method)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。工厂方法模式适用于当一个类不知道它所必须创建的对象的类的时候：当一个类希望由它的子类来指定它所创建的对象的时候；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：当一个系统应该独立于它的产品创建、构成和表示时；当要实例化的类是在运行时刻指定时，例如通过动态装载，为了避免创建一个与产品类层次平行的工厂类层次时；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。

对高级语言源程序进行编译的过程可以分为多个阶段，分配寄存器的工作在（48）阶段进行。

(48)A.词法分析　　B.语法分析　　C.语义分析　　D.目标代码生成

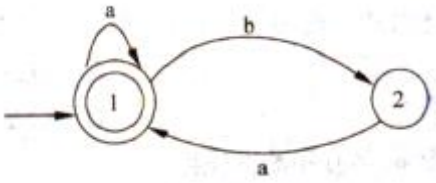
【答案】D

【解析】本题考查程序语言基础知识。

编译程序的功能是把某高级语言书写的源程序翻译成与之等价的目标程序(汇编语言或机器语言)。编译程序的工作过程可以分为词法分析、语法分析、语义分析、中间代码生成、代码优化、目标代码生成、符号表管理和出错处理等部分，如下图所示。

目标代码生成是编译器工作的最后一个阶段。这一阶段的任务是把中间代码变换成特定机器上的绝对指令代码、可重定位的指令代码或汇编指令代码，这个阶段的工作与具体的机器密切相关。因此在目标代码生成阶段分配寄存器。

以下关于下图所示有限自动机的叙述中，不正确的是（49）。



(49)A. 该自动机识别的字符串中 a 不能连续出现

B. 该自动机识别的字符串中 b 不能连续出现

C. 该自动机识别的非空字符串必须以 a 结尾

D. 该自动机识别的字符串可以为空串

【答案】A

【解析】本题考查程序语言基础知识。

自动机识别字符串的过程是：从初态出发，根据字符串的当前字符实现状态转移。如果存在从初态到终态的状态转移路径与字符串中的各个字符相匹配，那么就说该自动机可以识别该字符串。题中所给自动机的初态和终态都是编号为 1 的状态，从其状态图 可知，从状态 1 开始，识别出字符“a”时仍然转移到状态 1，而识别出字符“b”时才离开状态 1 进入状态 2,状态 2 仅对字符“a”有状态转移，且转回状态 1。因此，该自动机识别的字符串仅包含 a、b 字符，但是字符“b”不能连续出现，连续出现“a”是可以的。

对于大多数通用程序设计语言，用（50）描述其语法即可。

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

【答案】B

【解析】本题考查程序语言的基础知识。

乔姆斯基（Chomsky)把文法分成四种类型，即0型、1型、2型和3型。0型文法也称为短语文法，其能力相当于图灵机，任何0型语言都是递归可枚举的；反之，递归可枚举集也必定是一个0型语言。1型文法也称为上下文有关文法，这种文法意味着对非终结符的替换必须考虑上下文。2型文法就是上下文无关文法，非终结符的替换无需考虑上下文。3型文法等价于正规式，因此也被称为正规文法或线性文法。通用程序设计语言的大多数语法可由上下文无关文法表示。

在数据库逻辑结构设计阶段，需要（51）阶段形成的（52）作为设计依据。

(51)A.需求分析　　B.概念结构设计　　C.物理结构设计　D.数据库运行和维护

(52)A.程序文档、数据字典和数据流图　　B.需求说明文档、程序文档和数据流图

C.需求说明文档、数据字典和数据流图　D.需求说明文档、数据字典和程序文档

【答案】A　C

【解析】本题考查数据库系统基本概念方面的基础知识。

数据库设计主要分为用户需求分析、概念结构、逻辑结构和物理结构设计四个阶段。其中，在用户需求分析阶段中，数据库设计人员采用一定的辅助工具对应用对象的功能、性能、限制等要求所进行的科学分析，并形成需求说明文档、数据字典和数据流程图。 用户需求分析阶段形成的相关文档用以作为概念结构设计的设计依据。

给定关系模式R(A,B,C,D)、S(C,D,E)，与 $\pi_{1,3,5}(\sigma_{2='软件工程'}(R \bowtie S))$ 等价的SQL语句如下：SELECT (53) FROM R, S WHERE (54)下列查询B=“信息”且E=“北京”的A、B、E的关系代数表达式中，查询效率最高的是（55）。

（53）A.A,C,S.C　　B.A,B,E　　C.A,R.C,E　　D.A,R.C,S.D

(54)A.B=软件工程 OR R.C=S.C AND R.D=S.D

B.B=’软件工程’OR R.C=S.C AND R.D=S.D

C.B=’软件工程’OR R.C=S.C OR R.D=S.D

D. B='软件工程' AND R.C=S.C AND R.D=S.D

(55) A. $\pi_{1,2,7}(\sigma_{2='信息'\wedge 3=5\wedge 4=6\wedge 7='北京'}(R\times S))$

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

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

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

【答案】C D B

【解析】本题考查关系代数运算与 SQL 查询方面的基础知识。

试题（53）的正确答案为选项 C。$\pi_{1,3,5}(\sigma_{2='软件工程'}(R\bowtie S))$ 的含义是从 R⋈S 结果集中选取 B ='软件工程'的元组，再进行 R.A、R.C 和 S.E 投影。

试题（54）的正确答案为选项 D。自然联结 R⋈S 中的公共属性为 C、D，所以在 SQL 中可以用条件“WHERE R.C =S.C AND R.D =S.D”来限定。对于选取运算 $\sigma_{2='信息化'}$ 在 SQL 中可以用条件“WHERE B ='软件工程'”来限定。

(55) 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，减少运算量和从外存读块的次数。

②合并乘积与其后的选择运算为连接运算。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，需再扫描一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间比计算的时间少时，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，正确的选项是 B。

给定关系模式 R(U,F)，U={A,B,C,D,E,H}，函数依赖集 F={A→B,A→C,C→D,AE→H}。关系模式 R 的候选关键字为（56）。

（56）A. AC B. AB C. AE D. DE

【答案】C

【解析】本题考查关系数据库基础知识。

试题（56)的正确答案为选项C。关系模式R中，属性AE仅出现在函数依赖集F左部，而其余属性都不是左右都未出现的属性，所以AE必为R的唯一候选码。

对于线性表，相对于顺序存储，采用链表存储的缺点是（57）。

（57）A.数据元素之间的关系需要占用存储空间，导致存储密度不高

B.表中结点必须占用地址连续的存储单元，存储密度不高

C.插入新元素时需要遍历整个链表，运算的时间效率不高

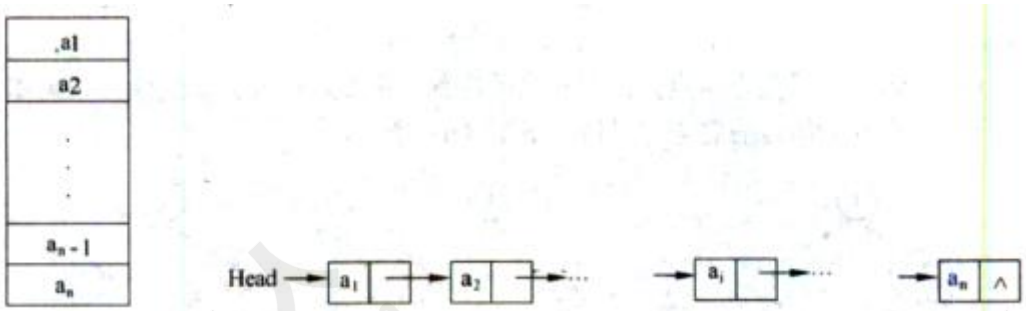
D.删除元素时需要遍历整个链表，运算的时间效率不高

【答案】A

【解析】本题考查数据结构基础知识。

对于线性表(a1，a2,…，an)，顺序存储时表中元素占用的存储单元地址是连续的，因此逻辑上相邻的元素，其物理位置也相邻，如下图（a）所示。

线性表采用链式存储有单链表、双向链表、循环链表等形式，单链表如下图（b）所示。链式存储的基本特点是逻辑上相邻的元素不要求物理位置上相邻,所以需要在元素的存储单元中专门表示下一个（或上一个）元素的存储位置信息，从而可以得到元素间的顺序信息。



若一个栈初始为空，其输入序列是1，2，3，…，n-1，n，其输出序列的第一个元素为k(1≤k≤「n/2」)，则输出序列的最后一个元素是（58）。

（58）A.值为n的元素　　B.值为1的元素　　C.值为n-k的元素　　D.不确定的

【答案】D

【解析】本题考查数据结构基础知识。

以n等于4举例说明。输入序列为1 2 3 4,输出序列的第一个元素可以为1或2。若为1，则输出序列可能为1 234、1 243、1 3 42、1 3 24、1 432；若为2,则输出序 列为 2134、

2143、2314、234 1、243 1。

以上序列都可由合法的入栈、出栈操作序列给出，从中可知无法确定输出序列中最后 1 个元素的值。

某个二叉查找树(即二叉排序树)中进行查找时，效率最差的情形是该二叉查找树是(59)。

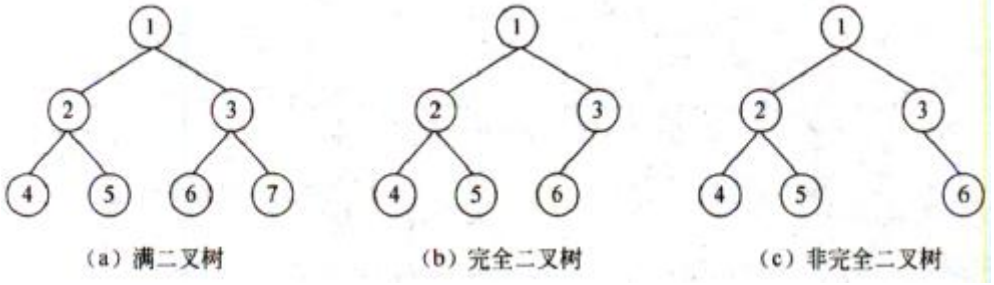
（59）A.完全二叉树　　B.平衡二叉树　　C.单枝树　　D.满二叉树

【答案】C

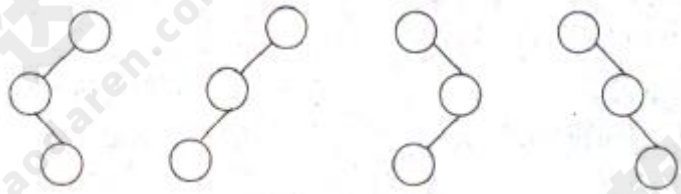
【解析】本题考查数据结构基础知识。

非空二叉查找树中的结点分布特点是左子树中的结点均小于树根，右子树中的结点均大于树根。因此，在二叉查找树中进行查找时，走了一条从树根出发到所找到结点的路径，到达一个空的子树则表明查找失败。

根据定义，高度为 h 的满二叉树中有 2h-1 个结点，每一层上的结点数都达到最大值。完全二叉树的最高层只要求结点先占据左边的位置。例如，高度为 3 的满二叉树如下图(a)所示，具有 6 个结点的完全二叉树如下图（b)所示。



在平衡二叉树中，任何一个结点的左子树高度与右子树高度之差的绝对值不大于 1 单枝树中给每个结点只有 1 个子树。例如，具有 3 个结点的单枝树如下图所示。



显然，在结点数确定后，二叉查找树的形态为单枝树时查找效率最差。

在字符串的 KMP 模式匹配算法中，需先求解模式串的 next 函数值，其定义如下式所示，j 表示模式串中字符的序号（从 1 开始)。若模式串 p 为“abaac”，则其 next 函数值为(60)。

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

（60）A.01234　　B.01122　　C.01211　　D.01111

【答案】B

【解析】本题考查字符串的模式匹配运算知识。

KMP 是进行字符串模式匹配运算效率较高的算法。根据对 next 函数的定义，模式串前两个字符的 next 值为 0、1。对于第 3 个字符“a”，其在模式串中的前缀为“ab”， 从该子串找不出前缀和后缀相同的部分，因此，根据定义，该位置字符的 next 值为 1。对于第 4 个字符“a,,，其在模式串中的前缀为“aba”，该子串只有长度为 1 的前缀“a”和后缀“a”相同，根据定义，该位置字符的 next 值为 2。

对于第 5 个字符“c”，其在模式串中的前缀为“abaa”，该子串只有长度为 1 的前缀“a”和后缀“a”相同，根据定义，该位置字符的 next 值为 2。

综上可得，模式串“abaac”的 next 函数值为 01122。

快速排序算法在排序过程中，在待排序数组中确定一个元素为基准元素，根据基准元素把待排序数组划分成两个部分，前面一部分元素值小于等于基准元素，而后面一部分元素值大于基准元素。然后再分别对前后两个部分进一步进行划分。根据上述描述，快速排序算法采用了（61）算法设计策略。可知确定基准元素操作的时间复杂度为Θ (n)，则快速排序算法的最好和最坏情况下的时间复杂度为（62）。

（61）A.分治　　B.动态规划　　C.贪心　　D.回溯

(62) A. Θ(n)和Θ(nlgn)　　B. Θ(n)和Θ(n2)

C. Θ(nlgn)和Θ(nlgn)　　D. Θ(nlgn)和Θ(n2)

【答案】A　D

【解析】本题考查快速排序算法。

快速排序算法是应用最为广泛的排序算法之一。其基本思想是将 n 个元素划分为两个部分：一部分元素值小于某个数；另一部分元素值大于某个数，该数的位置确定：然后进一步划分前面部分和后面部分。根据该叙述，可以知道，这里采用的是分治算法设计策略。

由于已知划分操作的时间复杂度为Θ(n)，不需要合并子问题的答案。对于最好的情况，应该是每次划分都把 n 个元素划分为大约 2 个 n/2 个元素的子数组，此时

$T(n) = 2T(n/2) + \Theta(n)$

解该递归式，可得时间复杂度为Θ(nlgn)。若刚好划分的极度不均衡，即每个划分刚好把 n 个元素划分为一边 0 个元素，一边 n-1 个元素，此时

$T(n) = T(n-1) + \Theta(n)$

解该递归式，可得时间复杂度为$\Theta(n^2)$。

对一待排序序列分别进行直接插入排序和简单选择排序，若待排序序列中有两个元素的值相同，则（63）保证这两个元素在排序前后的相对位置不变。

（63）A. 直接插入排序和简单选择排序都可以 B. 直接插入排序和简单选择排序都不能

C. 只有直接插入排序可以 D. 只有简单选择排序可以

【答案】C

【解析】本题考查简单排序算法特点。

直接插入排序的思想是：是将 n 个待排序的元素由一个有序表和一个无序表组成，开始时有序表中只包含一个元素。排序过程中，每次从无序表中取出第一个元素，将其插入到有序表中的适当位置，使有序表的长度不断加长，完成排序过程。

例如，对序列 21，48，21*，9 进行直接插入排序，21 和 21*的相对位置在排序前后可保持，如下所示：

第一趟得到有序子序列：21，48

第二趟得到有序子序列：21，21*，48

第三趟得到有序序列：9,21，21*，48

简单选择排序的过程是：第一趟在 n 个记录中选取最小记录作为有序序列的第一个记录；第二趟在 n-1 个记录中选取最小记录作为有序序列的第二个记录：第 i 趟在 n-i+1 个记录中选取最小的记录作为有序序列中的第 i 个记录，直到将序列排列有序。

对序列 21，48，21*，9 进行简单选择排序，过程如下：

第一趟选出最小元素，将其交换至 1 号位置，序列为：9,48,21*，21

第二趟选出次小元素，将其交换至 2 号位置，序列为：9,21*，48,21

第三趟选出第三小元素，将其交换至 3 号位置，序列为：9,21*，21，48

从该例可知，简单选择排序过程不能保证排序码相同的两个元素在排序前后的相对位置不变，直接插入排序则可以。

已知一个文件中出现的各字符及其对应的频率如下表所示。若采用定长编码，则该文件中字符的码长应为（64）。若采用 Huffman 编码，则字符序列“face”的编码应为（65）。

字符	a	b	c	d	e	f
频率(%)	45	13	12	16	9	5

(64) A. 2　　B. 3　　C. 4　　D. 5

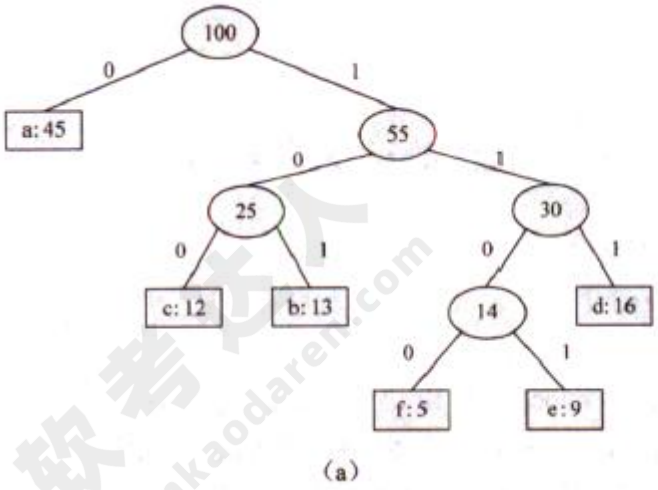
(65) A. 110001001101　　B. 001110110011　　C. 101000010100　　D. 010111101011

【答案】B　A

【解析】本题考查 Huffman 编码的相关知识。

字符在计算机中是用二进制表示的，每个字符用不同的二进制编码来表示。码的长度影响存储空间和传输效率。若是定长编码方法，用 2 位码长，只能表示 4 个字符，即 00、01、10 和 11;若用 3 位码长，则可以表示 8 个字符，即 000、001、010、011、100、 101、110、111。对于题中给出的例子，一共有 6 个字符，因此采用 3 位码长的编码可以表示这些字符。

Huffman 编码是一种最优的不定长编码方法，可以有效的压缩数据。要使用 Huffman 编码，除了知道文件中出现的字符之外，还需要知道每个字符出现的频率。下图（a)是题下中给出对应的编码树，可以看到，每个字符及其对应编码为图（b)，因此字符序列“face”的编码应为 1100 0 1001101,即 65 选择 A。



(a)

字符	编码
a	0
b	101
c	100
d	111
e	1101
f	1100

(b)

PPP 中的安全认证协议是（66），它使用三次握手的会话过程传送密文。

（66）A. MD5　　B. PAP　　C. CHAP　　D. HASH

【答案】C

【解析】

PPP 认证是可选的。PPP 扩展认证协议（Extensible Authentication Protocol，EAP）可支持多种认证机制，并且允许使用后端服务器来实现复杂的认证过程。例如通过 Radius 服务器进行 Web 认证时，远程访问服务器（RAS）只是作为认证服务器的代理传递请求和应答报文，并且当识别出认证成功/失败标志后结束认证过程。通常 PPP 支持的两个认证协议是：①口令验证协议（Password Authentication Protocol，PAP）：提供了一种简单的两次握手认证方法，由终端发送用户标识和口令字，等待服务器的应答，如果认证不成功，则终止连接。这种方法不安全，因为采用文本方式发送密码，可能会被第三方窃取。

②质询握手认证协议（Challenge Handshake Authentication Protocol，CHAP）：采用三次握手方式周期地验证对方的身份。首先是逻辑链路建立后认证服务器就要发送一个挑战报文（随机数），终端计算该报文的 Hash 值并把结果返回服务器，然后认证服务器把收到的 Hash 值与自己计算的 Hash 值进行比较，如果匹配，则认证通过，连接得以建立，否则连接被终止。计算 Hash 值的过程有一个双方共享的密钥参与，而密钥是不通过网络传送的，所以 CHAP 是更安全的认证机制。在后续的通信过程中，每经过一个随机的间隔，这个认证过程都可能被重复，以缩短入侵者进行持续攻击的时间。值得注意的是，这种方法可以进行双向身份认证，终端也可以向服务器进行挑战，使得双方都能确认对方身份的合法性。

ICMP 协议属于因特网中的（67）协议，ICMP 协议数据单元封装在（68）中传送。

（67）A. 数据链路层　　B. 网络层　　C. 传输层　　D. 会话层

（68）A. 以太帧　　B. TCP 段　　C. UDP 数据报　　D. IP 数据报

【答案】B　D

【解析】

ICMP（Internet control Message Protocol）与 IP 协议同属于网络层，用于传送有关通信问题的消息，例如数据报不能到达目标站，路由器没有足够的缓存空间，或者路由器向发送主机提供最短通路信息等。ICMP 报文封装在 IP 数据报中传送，因而不保证可靠的提交。

DHCP 客户端可从 DHCP 服务器获得（69）。

（69）A. DHCP 服务器的地址和 Web 服务器的地址

B. DNS 服务器的地址和 DHCP 服务器的地址

C. 客户端地址和邮件服务器地址

D.默认网关的地址和邮件服务器地址

【答案】B

【解析】本试题考查 DHCP 协议的工作原理。

DHCP 客户端可从 DHCP 服务器获得本机 IP 地址、DNS 服务器的地址、DHCP 服务器的地址、默认网关的地址等，但没有 Web 服务器、邮件服务器地址。

分配给某公司网络的地址块是 210.115.192.0/20，该网络可以被划分为（70）个 C 类子网。

（70）A.4　　B.8　　C.16　　D.32

【答案】C

【解析】

由于分配给公司网络的地址块是 210.115.192.0/20,留给子网掩码的比特数只有 4 位，所以只能划分为 16 个 C 类子网，这 16 个 C 类子网的子网号为 11000000～11001111，即 192～207，所以 210.115. 210. 0 不属于该公司的网络地址。

Teams are required for most engineering projects. Although some small hardware or software products can be developed by individuals, the scale and complexity of modem systems is such, and the demand for short schedules so great, that it is no longer （71） for one person to do most engineering jobs. Systems development is a team （72） ,and the effectiveness of the team largely determines the （73） of the engineering.

Development teams often behave much like baseball or basketball teams. Even though they may have multiple specialties, allthe members work toward （74） .However,on systems maintenance and enhancement teams, the engineers often work relatively independently, much like wrestling and track teams.

A team is （75） just a group of people who happen to work together. Teamwork takes practice and it involves special skills. Teams require common processes; they need agreed-upon goals; and they need effective guidance and leadership. The methods for guiding and leading such teams are well known, but they are not obvious.

（71）A.convenient	B.existing	C.practical	D.real
（72） A.activity	B.job	C.process	D.application
（73）A.size	B.quality	C.scale	D.complexity

(74)A.multiple objectives　　B.different objectives

C.a single objective　　D.independent objectives

(75) A.relatively	B./	C.only	D.more than

【答案】C　A　B　C　D

【解析】

大多数工程项目需要团队完成。虽然有些小规模的硬件或软件产品可以由个人完成，但是现代系统的规模大、复杂性高以及开发周期短的极高需求，使得一个人完成大多工程工作已经不再现实。系统开发是一个团队活动，团队的效率很大程度上决定工程的质量。

开发团队经常表现的像是棒球队或篮球队，即使棒球队或篮球队可能有多种不同专长，但是所有的队员都朝着一个目标努力。然而在系统维护和增强团队，工程师们的工作就像摔跤和田径队一样经常相对独立。

团队不仅仅是一群人碰巧在一起工作，团队工作需要实践，涉及到多种特殊的技能。 团队需要共同的过程，需要达成一致的目标，需要有效地指导和领导。尽管指导和领导这样的团队的方法是众所周知的，但是它们并不明显。

试题一（共 15 分）

阅读下列说明和图，回答问题 1 至问题 3,将解答填入答题纸的对应栏内。

【说明】

某大型披萨加工和销售商为了有效管理生产和销售情况，欲开发一披萨信息系统，其主要功能如下：

（1）销售。处理客户的订单信息，生成销售订单，并将其记录在销售订单表中。销售订单记录了订购者、所订购的披萨、期望的交付日期等信息。

（2）生产控制。根据销售订单以及库存的披萨数量，制定披萨生产计划（包括生产哪些披萨、生产顺序和生产量等），并将其保存在生产计划表中。

（3）生产。根据生产计划和配方表中的披萨配方，向库存发出原材料申领单，将制作好的披萨的信息存入库存表中，以便及时进行交付。

（4）采购。根据所需原材料及库存量，确定采购数量，向供应商发送采购订单，并将其记录在采购订单表中；得到供应商的供应量，将原材料数量记录在库存表中，在采购订单表中标记已完成采购的订单。

（5）运送。根据销售订单将披萨交付给客户，并记录在交付记录表中。

（6）财务管理。在披萨交付后，为客户开具费用清单，收款并出具收据；依据完成的采购订单给供应商支付原材料费用并出具支付细节；将收款和支付记录存入收支记录表中。

（7）存储。检查库存的原材料、拔萨和未完成订单，确定所需原材料。

现采用结构化方法对披萨信息系统进行分析与设计，获得如图 1-1 所示的上下文数据流图和图 1-2 所示的 0 层数据流图。

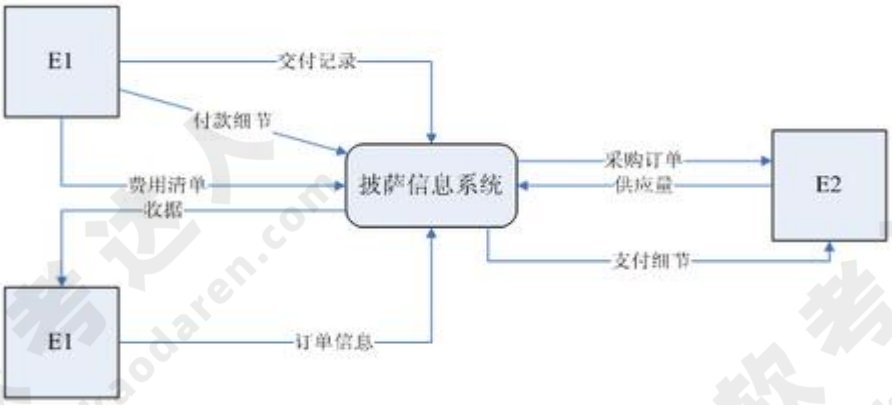


图1-1 上下文数据流图

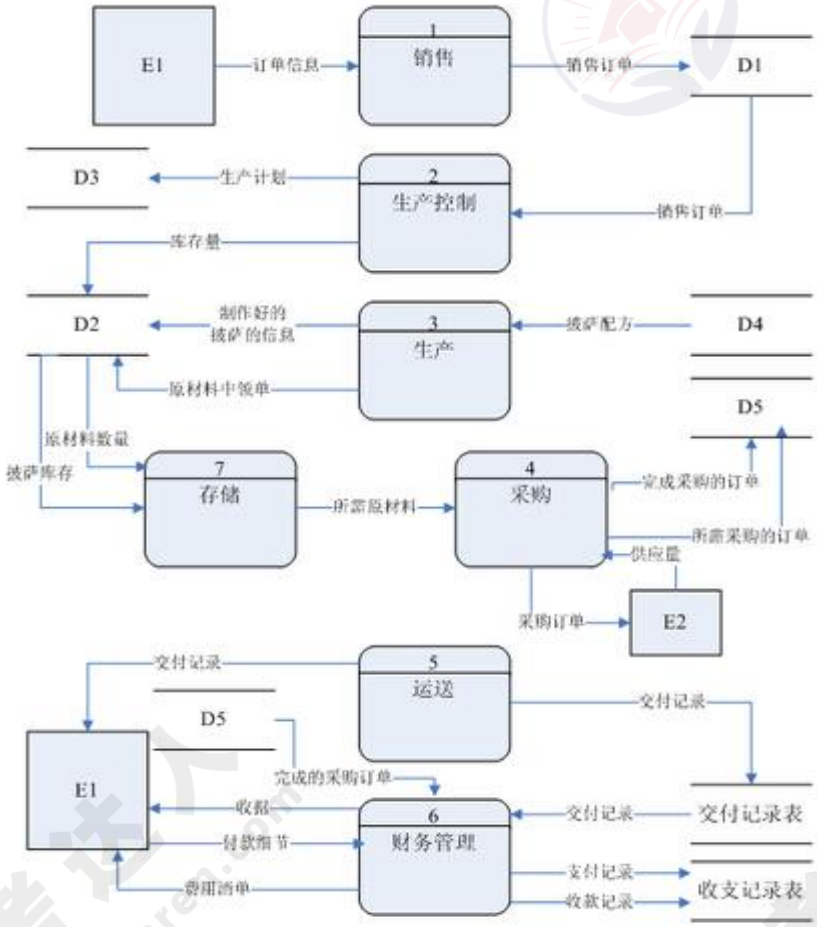


图1-2 0层数数据流图

【问题 1】

根据说明中的词语，给出图 1-1 中的实体 E1～E2 的名称。

E1：客户；E2：供应商

【试题分析】

本题考查采用结构化方法进行系统分析与设计，主要考查数据流图（DFD)的应用，是比较传统的题目，考点与往年类似，要求考生细心分析题目中所描述的内容。

DFD 是一种便于用户理解、分析系统数据流程的图形化建模工具，是系统逻辑模型的重要组成部分。上下文 DFD (顶层 DFD)通常用来确定系统边界，将待开发系统看作一个大的加工（处理)，然后根据系统从哪些外部实体接收数据流，以及系统将数据流发送到哪些外部实体，建模出的上下文图中只有唯一的一个加工和一些外部实体，以及这两者之间的输入输出数据流。0 层 DFD 在上下文确定的系统外部实体以及与外部实体的输入输出数据流的基础上，将上下文 DFD 中的加工分解成多个加工，识别这些加工的输入输出数据流，使得所有上下文 DFD 中的输入数据流，经过这些加工之后变换成上下文 DFD 的输出数据流。根据 0 层 DFD 中加工的复杂程度进一步建模加工的内容。

在建分层 DFD 时，根据需求情况可以将数据存储建模在不同层次的 DFD 中，注意要在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其子图的输入输出数据流在数量和名字上相同，或者父图中的一个输入（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。

本问题考查上下文 DFD，要求确定外部实体。考察系统的主要功能，不难发现，系统中涉及到客户和供应商，没有提到其他与系统交互的外部实体。根据描述（1)中“处理客户的订单信息”等信息、从供应商出进行采购时“向供应商发送采购订单”，从而可确定 E1 为“客户”实体，E2 为“供应商”实体。

【问题 2】

根据说明中的词语，给出图 1-2 中的数据存储 D1～D5 的名称。

【参考答案】

D1：销售订单表；D2：库存表；D3：生产计划表；D4：配方表；D5：采购订单表

【试题分析】

本问题要求确定 0 层数据流图中的数据存储。分析说明中和数据存储有关的描述，根据说明中（1)客户的订单信息，生成销售订单，“并将其记录在销售订单表”，可知 D1 为销售订单表；说明中（2)根据销售订单以及库存的匹萨数量制定匹萨生产计划，并将其保存在生产计划表中；说明（3)中“根据生产计划和配方表中的匹萨配方”、“将制作好的匹萨的信息存入

库存表中”，可知 D2 为库存表、D3 为生产计划表、D4 为匹萨配方表；说明（4)中向供应商发送采购订单，“并将其记录在采购订单表中”，可能 D5 为采购订单表。

【问题 3】

根据说明和图中词语，补充图 1-2 中缺失的数据流及其起点和终点。

数 据 流	起 点	终 点
生产计划	D3 或 生产计划表	3 或 生产
未完成订单	D1 或 销售订单表	7 或 存储
销售订单	D1 或 销售订单表	5 或 运送
原材料数量	4 或 采购	D2 或 库存表
库存量	D2 或 库存表	4 或 采购
支付细节	6 或 财务管理	E2 或 供应商

【试题分析】

本问题要求补充缺失的数据流及其起点和终点。对照图 1-1 和图 1-2 的输入、输出数据流，数量不同，考查图 1-1 中输出至 E2 的数据流，有“采购订单”和“支付细节”，而图 1-2 中缺少了 “支付细节”数据流，所以需要确定此数据流或者其分解的数据流的起点。

考查说明中的功能，功能（3)根据生产计划和配方表中的匹萨配方，向库存发出原材料申领单，对照图 1-2,不难发现，处理 3 缺少了输入流生产计划，而生产计划存储在“生产计划表”中。功能（4)根据所需原材料及库存量，确定采购数量，可以看出处理（4）缺少了从库存表中获取的库存量；得到供应商的供应量，将原材料数量记录在库存表中，可以发现，缺少了从处理（4）到库存表的数据流“原材料数量”。图 1-2 中处理（5）没有输入流，考察功能（5)根据销售订单将匹萨交付给客户，可知，处理 5 的输入数据流为“销售订单”，其来源为数据存储订单记录表。功能（6)中依据完成的采购订单给供应商支付原材料费用并出具支付细节，可以看出缺少了支付细节输出流，结合对比图 1-1 和图 1-2,可知支付细节并未分解。功能（7)检查库存的原材料、匹萨和未完成订单，不难发现，处理 7 缺少输入流未完成订单，客户订单在数据存储“销售订单表”中。

试题二（共 15 分）

阅读下列说明，回答问题 1 至问题 3,将解答填入答题纸的对应栏内。

【说明】

某集团公司在全国不同城市拥有多个大型超市，为了有效管理各个超市的业务工作，需要构建一个超市信息管理系统。

【需求分析结果】

（1）超市信息包括：超市名称、地址、经理和电话，其中超市名称唯一确定超市关系的每一个元组。每个超市只有一名经理。

（2）超市设有计划部、财务部、销售部等多个部门，每个部门只有一名部门经理，有多名员工，每个员工只属于一个部门。部门信息包括：超市名称、部门名称、部门经理和联系电话。超市名称、部门名称唯一确定部门关系的每一个元组。

（3）员工信息包括：员工号、姓名、超市名称、部门名称、职位、联系方式和工资。其中，职位信息包括：经理、部门经理、业务员等。员工号唯一确定员工关系的每一个元组。

（4）商品信息包括：商品号、商品名称、型号、单价和数量。商品号唯一确定商品关系的每一个元组。一名业务员可以负责超市内多种商品的配给，一种商品可以由多名业务员配给。

【概念模型设计】

根据需求分析阶段收集的信息，设计的实体联系图和关系模式（不完整）如下：

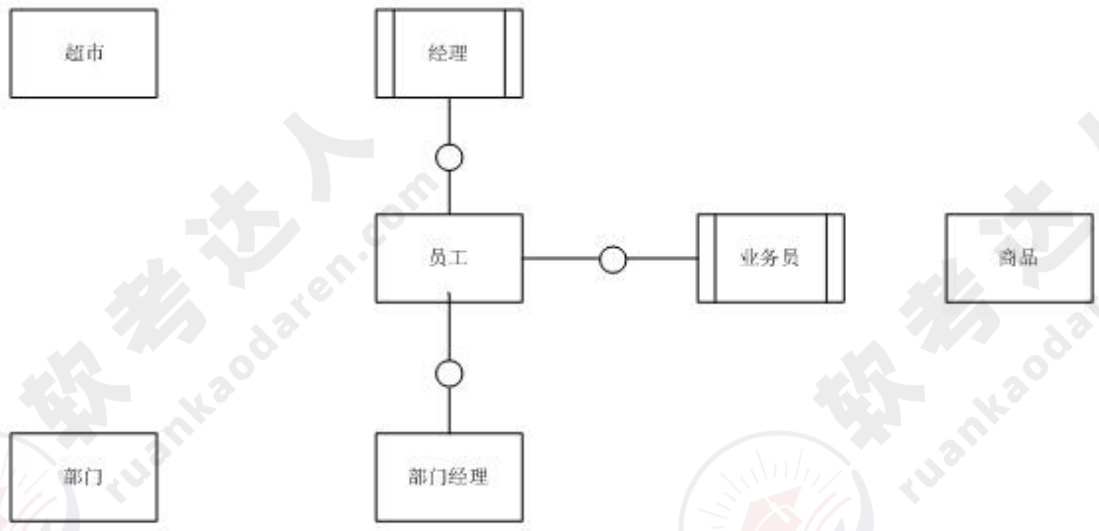


图1-1 实体联系图

【关系模式设计】

超市（超市名称，经理，地址，电话）

部门（ （a） ，部门经理，联系电话）

员工（ （b） ，姓名，联系方式，职位，工资）

商品（商品号，商品名称，型号，单价，数量）

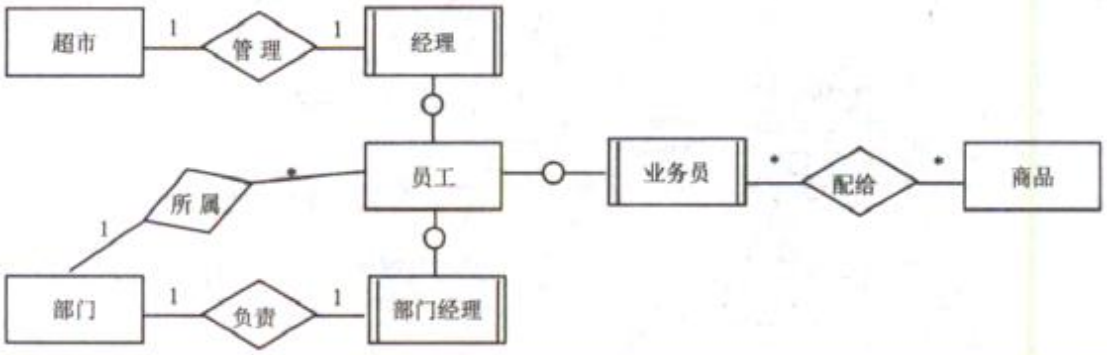
配给（ （c） ，配给时间，配给数量，业务员）

【问题 1】

根据问题描述，补充四个联系，完善图 1-1 的实体联系图。联系名可用联系 1、联系 2、联系 3 和联系 4 代替，联系的类型分为 1:1、1:n 和 m:n（或 1:1、1:*和*:*）。

【参考答案】

联系名称可不做要求，但不能出现重名。



【试题分析】

本题考查数据库概念结构设计及其向逻辑结构转换的过程。

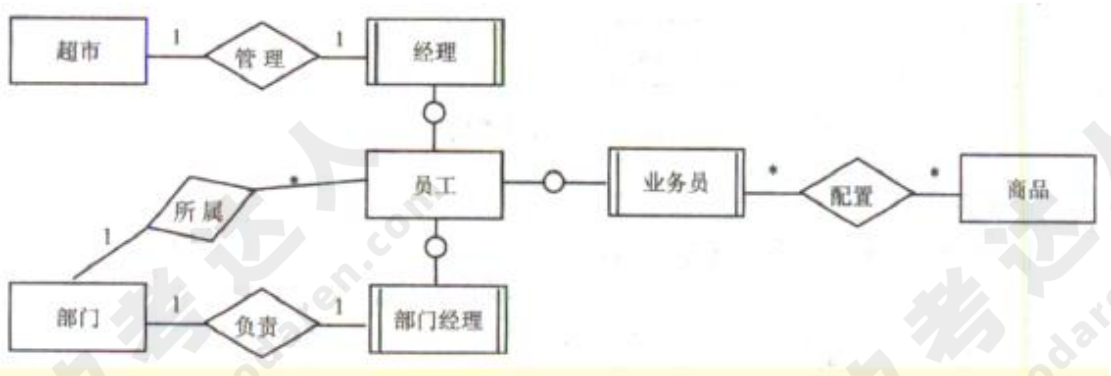
此类题目要求考生认真阅读题目对现实问题的描述，经过分类、聚集、概括等方法，从中确定实体及其联系。题目已经给出了 4 个实体，需要根据需求描述，给出实体间的联系。

根据题意，“每个超市有一位经理”并且部门经理也是超市的员工，可以得出超市与经理之间的管理联系类型为 1:1。又由于“每个部门有一名部门经理”并且部门经理也是超市的员工，可以得出部门与部门经理之间的负责联系类型为 1:1。

由“每个部门有多名员工，每个员工属于一个部门”可以得出部门与员工间的所属联系类型为 1:*;并且员工是经理的超类型，经理是员工的子类型。

由“一名业务员可以负责超市内多种商品配置，一种商品可以由多名业务员配置”，可以得出，业务员与商品之间的配置联系类型为*:*。

完整的 ER 图如下：



【问题 2】

（1）根据实体联系图，将关系模式中的空（a）～（c）补充完整；

（2）给出部门和配给关系模式的主键和外键。

【参考答案】

（1）（a）超市名称，部门名称

（b）员工号，超市名称，部门名称

（c）商品号

（2）部门关系模式的主键：超市名称，部门名称

外键：部门经理，超市名称

配给关系模式的主键：商品号，配给时间，业务员

外键：商品号，业务员

【试题分析】

(1) 完整的关系模式如下：

超市（超市名称，经理，地址，电话）

部门（超市名称，部门名称，部门经理，联系电话）

员工（员工号，超市名称，部门名称，姓名，联系方式，职位，工资）

商品（商品号，商品名称，型号，单价，数量）

配置（商品号，配置时间，配置数量，业务员）

(2) 部门和配置关系模式的主键和外键的分析如下：

在部门关系模式中，由于每个超市都设有计划部、财务部、销售部等多个部门，因此为了区分部门关系的每一个元组，需要“超市名称、部门名称”作为部门的主键。又因为部门经理也是员工，因此部门经理为员工关系的外键。

在配置关系模式中，“商品号，配置时间，业务员”唯一标识配置关系的每一个元组，故为配置关系的主键，外键为商品号，业务员。

【问题 3】

（1）超市关系的地址可以进一步分为邮编、省、市、街道，那么该属性是属于简单属性还是复合属性？请用 100 字以内文字说明。

（2）假设超市需要增设一个经理的职位，那么超市与经理之间的联系类型应修改为（d），超市关系应修改为（e）。

【参考答案】

(1) 该属性属于复合属性，因为简单属性是原子的、不可再分的。

(2) (d) 1:n

(e) 超市名称，经理，电话

【试题分析】

(1)超市的地址属性不属于简单属性。因为根据题意，超市关系的地址可以进一步分为邮编、省、市、街道，而简单属性是原子的、不可再分的，复合属性可以细分为更小的部分（即划分为别的属性)。本小题的超市的地址可以进一步分为邮编、省、市、街 道，故属于复合属性。

(2)因为根据题意，超市需要增设一位经理的职位，那么超市与经理之间的联系类型应修改为 1:n，超市的主键应修改为超市名称，经理，电话。

试题三（共 15 分）

阅读下列说明和图，回答问题 1 至问题 3,将解答填入答题纸的对应栏内。

【说明】

某公司欲开发一个管理选民信息的软件系统。系统的基本需求描述如下：

（1）每个人(Person)可以是一个合法选民(Eligible)或者无效的选民(Ineligible)。

（2）每个合法选民必须通过该系统对其投票所在区域（即选区，Riding）进行注册(Registration)。每个合法选民仅能注册一个选区。

（3）选民所属选区由其居住地址(Address)决定。假设每个人只有一个地址，地址可以是镇(Town)或者城市(City)。

（4）某些选区可能包含多个镇；而某些较大的城市也可能包含多个选区。

现采用面向对象方法对该系统进行分析与设计，得到如图 1-1 所示的初始类图。

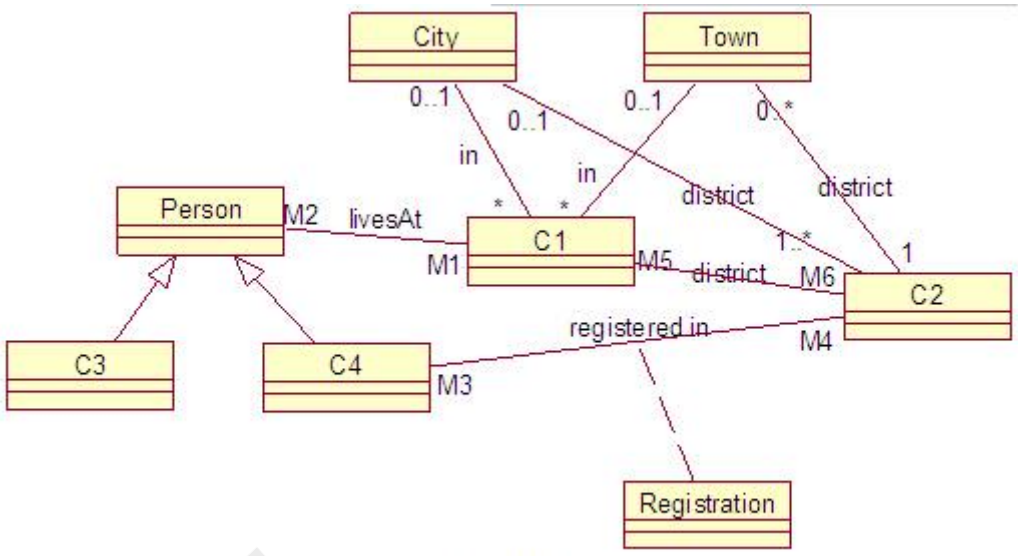


图1-1类图

【问题 1】

根据说明中的描述，给出图 1-1 中 C1～C4 所对应的类名（类名使用说明中给出的英文词汇）。

【参考答案】

C1：Address C2：Riding C3：Ineligible C4：Eligible

【试题分析】

本题属于经典的考题，主要考查面向对象分析方法与设计的基本概念与应用。在建模方面，本题中只涉及到了 UML 类图。类图上的考点也是比较常规的对类的识别以及多重度的确定，

题目难度不大。

根据【说明】中的“(1)每个人（Person)可以是一个合法选民（Eligible)或者无效的选民（Ineligible）”，可以推断出这里有一个“一般/特殊”关系，应采用继承结构。 再对照类图，C3、C4 处显而易见应该是 Ineligible 和 Eligible。由于 C4 和 C2 之间的关联关系，这里 C3 和 C4 的答案是不能互换的。

根据【说明】中的“(3)选民所属选区由其居住地址（Address)决定。假设每个人只有一个地址，地址可以是镇（Town)或者城市（City）”，可以推断出 C1、City、Town 这 3 个类描述的是与地址相关的内容，因此 C1 处应该是 Address。对应地，C2 处应该是 Riding，这个由 C2 与 City、C2 与 Town 之间的联系名称“district”也能推断出来。

【问题 2】

根据说明中的描述，给出图 1-1 中 M1～M6 处的多重度。

【参考答案】

M1：1，M2：*或 0..*，M3：*，M4：1，M5：*，M6：1。

【试题分析】

对于联系的多重度的判定，应注意题目中关于不同概念之间关联数量的描述。

M1、M2 这一对多重度，刻画的是“Person”和“Address”之间的关系。由【说明】中的“假设每个人只有一个地址”，可以得出 M1 和 M2 处分别为 1 和*。

M3 和 M4 描述的是合法选民与选区之间的关系。由【说明】中的“每个合法选民仅能注册一个选区”，可知 M3 和 M4 分别为*和 1。

M5 和 M6 描述的是选区和地址之间的关系。在【说明】中假设，每个合法选民在选区中只注册一个地址，因为 M5 和 M6 处分别为*和 1。

【问题 3】

对该系统提出了以下新需求：

（1）某些人拥有在多个选区投票的权利，因此需要注册多个选区；

（2）对于满足（1）的选民，需要划定其“主要居住地”，以确定他们应该在哪个选区进行投票。

为了满足上述需求，需要对阁3-1所示的类图进行哪些修改？请用100字以内文字说明。

【参考答案】

(1) M1 处改为 1..*，在 Registration 类中增加 address 属性，指明注册时使用的是哪个地址。

(2) 增加一个类“主要居住地”，作为类 Address 的子类；类 Person 与类“主要居住地”之间具有关系联系，且每个人只有一个主要居住地。

【试题分析】

本问题考查当需求发生变化时，对设计模型的修改。这里给出了两个需求变更，分别对初始类图进行修改。

需求 1:某些人拥有在多个选区投票的权利，因此需要注册多个选区。由于选区由住址确定，能够在多个选区注册，意味着其居住地址不止一个。所以“Person”和“Address” 之间的多重度会发生变化。在选区注册时所使用的地址也不唯一了，因此需要增加属性来记录在注册选区时所使用的地址，从而对 C2 和 C4 之间的关联类进行修改，增加其属性。

需求 2:对于满足需求 1 的选民，需要划定其“主要居住地”，以确定他们应该在哪个选区进行投票。这个需求对选民的地址信息提出了更为详细的需求，按照面向对象方法将“不变部分和可变部分分离”的思想，在类图中增加一个新的类，并采用继承机制继承原有 Address 类中的共性元素。

试题四

阅读下列说明和C代码，回答问题1至问题3，将解答写在答题纸的对应栏内。

【说明】

计算一个整数数组a的最长递增子序列长度的方法描述如下：

假设数组a的长度为n，用数组b的元素b[i]记录以a[i](0≤i<n)为结尾元素的最长递增予序列的长度，则数组a的最长递增子序列的长度为$\max_{0\le i<n}\{b[i]\}$；其中b[i]满足最优子结构，可递归定义为：

$$\begin{cases} b[0]=1 \\ b[i]=\max\limits_{\substack{0\le k\le i \\ a[k]\le a[i]}}\{b[k]\}+1 \end{cases}$$

【C代码】

下面是算法的C语言实现。

（1）常量和变量说明

a：长度为n的整数数组，待求其最长递增子序列

b：长度为n的数组，b[i]记录以a[i](0≤i<n)为结尾元素的最长递增子序列的长度，其中0≤i<n

len：最长递增子序列的长度

i,j：循环变量

temp：临时变量

（2）C程序

```
#include <stdio.h>
int maxL(int*b, int n) {
  int i, temp=0;
  for(i=0; i<n; i++) {
    if(b[i]>temp)
      temp=b[i];
  }
  return temp;
}
int main() {
  int n, a[100], b[100], i, j, len;
  scanf("%d", &n);
  for(i=0; i<n; i++) {
    scanf("%d", &a[i]);
  }
    (1)   ;
  for(i=1; i<n; i++) {
    for(j=0, len=0;   (2)   ; j++) {
      if(   (3)   && len<b[j])
        len=b[j];
    }
      (4)   ;
  }
  Printf("len:%d\n", maxL(b,n));
  printf("\n");
}
```

【问题 1】

根据说明和 C 代码，填充 C 代码中的空（1）～（4）。

【参考答案】

（1）b[0]=1

（2）j<i

（3）a[j]<=a[i]

（4）b[i]=len+1

【试题分析】

本题考查算法设计与分析以及用 C 程序设计语言来实现算法的能力。

此类题目要求考生认真阅读题目对问题的描述，理解算法思想，并会用 C 程序设计语言来实现。

根据题干描述，用一个数组 b 来记录数组 a 每个子数组的最长递增子序列的长度，即 b[i]记录 a[0..i]的最长递增子序列的长度。首先，只有一个元素的数组的最长递增子序列的长度为 1，即给 b[0]直接赋值 1。因此，空（1)处填写“b[0] = 1”。两重 for 循环中，第一重是从 a 数组的第二个元素开始，考虑每个子数组 a[0..i]的最长递增子序列的长度，第二重是具体的计算过程。考虑子数组 a[0..i]，其最长递增子序列的长度应该等于子数组 a[0..i-1]中的比元素 a[i]小的元素的最长递增子序列的长度加 1，当然，可能存在多个元素比元素 a[i]小，那么存在多个最长递增子序列的长度，此时，取最大者。因此，空处填写“j<i”，即考虑子数组 a[0..i-1]。空（3)处填写“a[j]<=a[i]”，要求元素值小于等于 a[i]而且目前的长度应该小于当前考虑的子数组的最长子序列长度。空（4) 处填写“b[i] = len+1”。简单的说，程序是根据题干给出的公式

$$\begin{cases} b[0]=1 \\ b[i]=\max\limits_{\substack{0\le k<i \\ a[k]<a[i]}}\{b[k]\}+1 \end{cases}$$

来实现的。另外，计算的过程不是采用递归的方式，而是以一种自底向上的方式进行的。

【问题 2】

根据说明和 C 代码，算法采用了 （5） 设计策略，时间复杂度为 （6） （用 O 符号表示)。

【参考答案】

（5）动态规划法

（6）O(n2)

【试题分析】

从题干说明和 C 程序来看，这是一个最优化问题，而且问题具有最优子结构，一个序列的最

长递增子序列由其子序列的最长递增子序列构成。在计算过程中采用了自底向上的方式来进行，这具有典型的动态规划特征。因此采用的是动态规划设计策略。

C 程序中，有两重 for 循环，因此时间复杂度为 O(n2)。

【问题 3】

已知数组 a={3,10,5,15,6,8}，根据说明和 C 代码，给出数组 b 的元素值。

【参考答案】

b={1,2,2,3,3,4}

【试题分析】

输入数组为数组 a = {3,10,5, 15,6, 8}，很容易得到，子数组 a[0..0]，a[0..1]，…，a[0..5]的最长递增子序列的长度分别为 1,2,2,3,3,4,因此答案为 b={1,2,2,3,3,4}。该题可以根据题干说明、C 代码来计算。由于输入很简单，答案也可以从输入直接计算出来。

试题五

阅读下列说明和C++代码，将应填入（n）处的字句写在答题纸的对应栏内。

【说明】

某灯具厂商欲生产一个灯具遥控器，该遥控器具有7个可编程的插槽，每个插槽都有开关按钮，对应着一个不同的灯。利用该遥控器能够统一控制房间中该厂商所有品牌灯具的开关，现采用Command（命令）模式实现该遥控器的软件部分。Command模式的类图如图1-1所示。

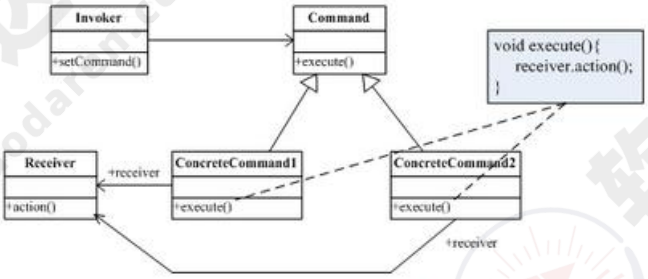


图1-1 Command模式类图

【C++代码】

```
class Light {
public:
  Light(string name) { /* 代码省略 */ }
  void on() { /* 代码省略 */ }   // 开灯
  void off() { /* 代码省略 */ } // 关灯
};
class Command {
public:
   （1）   ;
};
class LightOnCommand:public Command { // 开灯命令
private:
  Light* light;
public:
  LightOnCommand(Light* light) { this->light=light; }
  void execute() {   （2）   ; }
};
class LightOffCommand:public Command { // 关灯命令
private:
  Light *light;
public:
  LightOffCommand(Light* light) { this->light=light; }
  void execute() {   （3）   ; }
};
class RemoteControl{ // 遥控器
private:
  Command* onCommands[7];
  Command* offCommands[7];
public:
  RemoteControl() { /* 代码省略 */ }
  void setCommand(int slot, Command* onCommand, Command* offCommand) {
    （4）  =onCommand;
    （5）  =offCommand;
  }
  void onButtonWasPushed(int slot) {   （6）   ; }
  void offButtonWasPushed(int slot) {   （7）   ; }
};
int main() {
  RemoteControl* remoteControl=new RemoteControl();
  Light* livingRoomLight=new Light("Living Room");
  Light* kitchenLight=new Light("kitchen");
  LightOnCommand* livingRoomLightOn=new LightOnCommand(livingRoomLight);
  LightOffCommand* livingRoomLightOff=newLightOffCommand(livingRoomLight);
  LightOnCommand* kitchenLightOn=new LightOnCommand(kitchenLight);
  LightOffCommand* kitchenLightOff=new LightOffCommand(kitchenLight);
  remoteControl->setCommand(0, livingRoomLightOn, livingRoomLightOff);
  remoteControl->setCommand(1, kitchenLightOn, kitchenLightOff);
  remoteControl->onButtonWasPushed(0);
  remoteControl->offButtonWasPushed(0);
  remoteControl->onButtonWasPushed(1);
  remoteControl->offButtonWasPushed(1);
  /* 其余代码省略 */
  return 0;
}
```

【参考答案】

（1）virtual void execute()=0

（2）light->on()

（3）light->off()
（4）onCommands[slot]
（5）offCommands[slot]
（6）onCommands[slot]->execute()
（7）offCommands[slot]->execute()

【试题分析】
本题考查命令（Command)模式的基本概念和应用。
命令模式把一个请求或者操作封装到一个对象中。命令模式允许系统使用不同的请求把客户端参数化，对请求排队或者记录请求日志，可以提供命令的撤销和恢复功能。
在软件系统中，行为请求者与行为实现者之间通常呈现一种紧耦合的关系。但在某些场合，比如要对行为进行记录撤销重做事务等处理，这种无法抵御变化的紧耦合是不合适的。这种情况下，使用 Command 模式将行为请求者与行为实现者进行解耦。
题目中给出了 Command 模式的类图，其中：
Command 类为所有命令声明了一个接口。调用命令对象的 execute()方法，就可以让接收者进行相关的动作。
ConcreteCommand 类定义了动作和接收者之间的绑定关系。调用者只要调用 execute() 就可以发出请求，然后由 ConcreteCommand 调用接收者的一个或多个动作。
Invoker 持有一个命令对象，并在某个时间点调用命令对象的 execute()方法，将请求付诸实行。
Receiver 知道如何进行必要的工作，实现这个请求。任何类都可以当接收者。
了解了 Command 模式的内涵之后，下面来看程序。
由于 Command 类的主要作用是为所有的 ConcreteCommand 定义统一的接口，在 C++ 中通常采用抽象类来实现。C++的抽象类是至少具有一个纯虚拟函数的类。本题中的接口就是 execute()方法，所以(1)处要求填写的是纯虚拟函数 execute 的定义方式，即 virtual void execute()= 0。
类 LightOnCommand、LightOffCommand 对应的就是模式中的 ConcreteCommand。
ConcreteCommand 中 execute()方法的代码在类图中已经给出，现在需要确定 receiver 是谁。类 Light 充当的是 Receiver，其中定义了两种 action: on 和 off。所以（2)、(3)对应代码分别为 light->on()、light->off()。
类 RemoteControl 对应的是模式中的 Invoker，在该类中设置需要控制的命令对象。(4)处对应的代码为 onCommandstslot]，设置“开灯”命令对象：(5)处对应的代码为 offCommands[slot],设置‘关灯”命令对象。类 RemoteControl 中的方法 onButtonWasPushed 和 offButtonWasPushed，分别完成对开灯、关灯命令对象的 execute 方法的调用，所以(6)、(7) 处分别对应代码 onCommands[slot]->execute()、offCommands[slot]->execute()。

试题六

阅读下列说明和Java代码，将应填入 (n) 处的字句写在答题纸的对应栏内。

【说明】

某灯具厂商欲生产一个灯具遥控器，该遥控器具有7个可编程的插槽，每个插槽都有开关灯具的开关，现采用Command（命令）模式实现该遥控器的软件部分。Command模式的类图如图1-1所示。

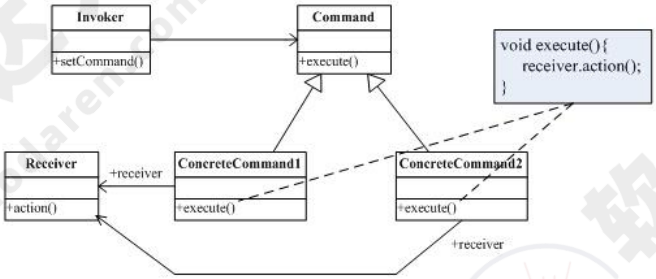


图1-1 Command模式类图

【Java代码】

```
class Light {
  public Light() {}
  public Light(String name) { /* 代码省略 */ }
  public void on()  { /* 代码省略 */ }    // 开灯
  public void off()  { /* 代码省略 */ }    // 关灯
  // 其余代码省略
}
  （1）  {
  public void execute();
}
class LightOnCommand implements Command { // 开灯命令
  Light light;
  public LightOnCommand(Light light) { this.light=light; }
  public void execute() {  （2）  ; }
}
class LightOffCommand implements Command { // 关灯命令
  Light light;
  public LightOffCommand(Light light) { this.light=light; }
  public void execute(){  （3）  ; }
}
class RemoteControl { // 遥控器
  Command[] onCommands=new Command[7];
  Command[] offCommands=new Command[7];
  public RemoteControl() { /* 代码省略 */ }
  public void setCommand(int slot, Command onCommand, Command offCommand) {
    （4）  =onCommand;
    （5）  =offCommand;
  }
  public void onButtonWasPushed(int slot) {
     （6）  ;
  }
  public void offlButtonWasPushed(int slot){
     （7）  ;
  }
}
class RemoteLoader {
  public static void main(String[] args) {
    RemoteControl remoteControl=new RemoteControl();
    Light livingRoomLight=new Light("Living Room");
    Light kitchenLight=new Light("kitchen");
    LightOnCommand livingRoomLightOn=new LightOnCommand(livingRoomLight);
    LightOffCommand livingRoomLightOff=new LightOffCommand(livingRoomLight);
    LightOnCommand kitchenLightOn=new LightOnCommand(kitchenLight);
    LightOffCommand kitchenLightOff=new LightOffCommand(kitchenLight);
    remoteControl.setCommand(0, livingRoomLightOn, livingRoomLightOff);
    remoteControl.setCommand(1, kitchenLightOn, kitchenLightOff);
    remoteControl.onButtonWasPushed(0);
    remoteControl.offButtonWasPushed(0);
    remoteControl.onButtonWasPushed(1);
    remoteControl.offButtonWasPushed(1);
  }
}
```

【参考答案】

（1）interface Command

（2）light.on()

（3）light.off()

（4）onCommands[slot]

（5）offCommands[slot]

（6）onCommands[slot].execute()

（7）offCommands[slot].execute()

【试题分析】

本题考查命令（Command)模式的基本概念和应用。

命令模式把一个请求或者操作封装到一个对象中。命令模式允许系统使用不同的请求把客户端参数化，对请求排队或者记录请求日志，可以提供命令的撤销和恢复功能。

在软件系统中，行为请求者与行为实现者之间通常呈现一种紧耦合的关系。但在某些场合，比如要对行为进行记录撤销重做事务等处理，这种无法抵御变化的紧耦合是不合适的。这种情况下，使用 command 模式将行为请求者与行为实现者进行解耦。

题目中给出了 Command 模式的类图，其中：

Command 类为所有命令声明了一个接口。调用命令对象的 execute()方法，就可以让接收者进行相关的动作。

ConcreteCommand 类定义了动作和接收者之间的绑定关系。调用者只要调用 execute()就可以发出请求，然后由 ConcreteCommand 调用接收者的一个或多个动作。

Invoker 持有一个命令对象，并在某个时间点调用命令对象的 execute()方法，将请求付诸实行。

Receiver 知道如何进行必要的工作，实现这个请求。任何类都可以当接收者。

了解了 Command 模式的内涵之后，下面来看程序。

由于 Command 类的主要作用是为所有的 ConcreteCommand 定义统一的接口，在 Java 中通常采用接口（Interface)来实现，所以（1)处对应的代码为 interface Command。

类 LightOnCommand 、 LightOffCommand 对应的就是模式中的 ConcreteCommand 。ConcreteCommand 中 execute()方法的代码在类图中已经给出，现在需要确定 receiver 是谁。类 Light 充当的是 Receiver，其中定义了两种 action: on 和 off。所以（2)、(3)对应代码分别为 light.on()和 light.off()。

类 RemoteControl 对应的是模式中的 Invoker，在该类中设置需要控制的命令对象。(4)处对应的代码为 onCommands[slot]，设置“开灯”命令对象；(5)处对应的代码为 offPCommands[slot]，设置“关灯”命令对象。类 RemoteControl 中的方法 onButtonWasPushed 和 offButtonWasPushed，分别完成对开灯、关灯命令对象的 execute 方法的调用。所以(6)、(7)处分别对应代码 onCommands[slot].execute()、offCommands[slot].execute()。