

在程序执行过程中，Cache 与主存的地址映射是由(1)完成的。

- (1) A. 操作系统      B. 程序员调度      C. 硬件自动      D. 用户软件

**【答案】C**

**【解析】**

由于 Cache 比主存小的多，因此必须使用一种机制将主存地址定位到 Cache 中，即地址映射。这个映射过程全部由硬件实现，对程序员透明。

某四级指令流水线分别完成取指、取数、运算、保存结果四步操作。若完成上述操作的时间依次为 8ns、9ns、4ns、8ns，则该流水线的操作周期应至少为(2) ns。

- (2) A. 4      B. 8      C. 9      D. 33

**【答案】C**

**【解析】**

流水线稳定之后，9ns 就会完成一个周期

内存按字节编址。若用存储容量为 32Kx8bit 的存储器芯片构成地址从 A0000H 到 DFFFFH 的内存，则至少需要(3)片芯片。

- (3) A. 4      B. 8      C. 16      D. 32

**【答案】B**

**【解析】**

$DFFFFH - A0000H + 1 = 40000H = 4 \times 16^4$ ；因为是按字节编址，所以所需芯片数为： $4 \times 16^4 / (32 \times 1024) = 8$

计算机系统的主存主要是由(4)构成的。

- (4) A. DRAM      B. SRAM      C. Cache      D. EEPROM

**【答案】A**

**【解析】**

DRAM (Dynamic Random Access Memory)，即动态 随机存取存储器，最为常见的系统内存。

SRAM (Static Random Access Memory)，即静态随机存取存储器。它是一种具有静止存取功能的内存，不需要刷新电路即能保存它内部存储的数据。

Cache，是一种后关系型数据库。能并发访问同一数据的数据库技术。

EEPROM(Electrically Erasable Programmable read only memory)，电可擦可编程只读存储器——一种掉电后数据不丢失的存储芯片。EEPROM 可以在电脑上或专用设备上擦除已有信息，重新编程。一般用在即插即用。

以下关于海明码的叙述中，正确的是(5)。

- (5)A. 海明码利用奇偶性进行检错和纠错
- B. 海明码的码距为 1
- C. 海明码可以检错但不能纠错
- D. 海明码中数据位的长度与校验位的长度必须相同

**【答案】A**

**【解析】**

海明码是奇偶校验的一种扩充。和奇偶校验的不同之处在于海明码采用多位校验码的方式，在这些多个校验位中的每一位都对不同的信息数据位进行奇偶校验，通过合理地安排每个校验位对原始数据进行的校验的位组合，可以达到发现错误、纠正错误的目的（当出现两位错误时，海明码能够查错，但无法纠错）。

还需要记住以下几个关键的关系。

- 可查出多少位错误：可以发现“ $\leq$ 码距-1”位的错误
- 可以纠正多少位错误：可以纠正“ $<$ 码距/2”位的错误，因此如果要能够纠正  $n$  位错误，所需最小的码距应该是“ $2n+1$ ”

计算机运行过程中，CPU 需要与外设进行数据交换。采用(6)控制技术时，CPU 与外设可并行工作。

- (6)A. 程序查询方式和中断方式
- B. 中断方式和 DMA 方式
- C. 程序查询方式和 DMA 方式
- D. 程序查询方式、中断方式和 DMA 方式

**【答案】B**

**【解析】**

程序查询方式：由于无中断机构，处理机对 I/O 设备的控制采取程序 I/O 方式，或称为

忙-等待方式,即在处理机向控制器发送一条 I/O 指令启动输入设备输入数据时,要同时把状态寄存器中的忙/闲标志置为 1。然后便不断测试标志。当为 1 时,表示输入机尚未输完一个字,处理机应继续对该标志测试,直到它为 0,表明数据已输入到控制器的数据寄存器中,于是处理机将数据取出送入内存单元,便完成了一个字的 I/O 可见,CPU 和外设是串行执行

中断方式:当某进程要启动某个 I/O 设备时,便由 CPU 向相应的设备控制器发出一条 I/O 命令,然后立即返回继续执行原来的任务。设备控制器于是按照命令的要求去控制指定 I/O 设备。这时 CPU 与 I/O 设备并行操作。

DMA 方式:数据传输的基本单位是数据块;所传送的数据是从设备直接送入内存的,或者相反;仅在传送一个或多个数据块的开始和结束时,才需 CPU 干预,整块数据的传送是在控制器的控制下完成的。

与 HTTP 相比,HTTPS 协议对传输的内容进行加密,更加安全。HTTPS 基于 (7) 安全协议,其默认端口是 (8)。

- (7) A. RSA                      B. DES                      C. SSL                      D. SSH
- (8) A. 1023                      B. 443                      C. 80                      D. 8080

【答案】C    B

【解析】

HTTPS (全称: Hyper Text Transfer Protocol over Secure Socket Layer),是以安全为目标的 HTTP 通道,简单讲是 HTTP 的安全版。即 HTTP 下加入 SSL 层,HTTPS 的安全基础是 SSL,因此加密的详细内容就需要 SSL。它是一个 URI scheme (抽象标识符体系),句法类同 http:体系。HTTPS 使用端口 443,而不是像 HTTP 那样使用端口 80 来和 TCP/IP 进行通信。

下列攻击行为中,属于典型被动攻击的是 (9)。

- (9) A. 拒绝服务攻击                      B. 会话拦截
- C. 系统干涉                      D. 修改数据命令

【答案】C

【解析】

系统干涉来自邻近的攻击者访问并实施窃听、降级等干涉系统的攻击。

(10) 不属于入侵检测技术。

- (10)A. 专家系统      B. 模型检测      C. 简单匹配      D. 漏洞扫描

**【答案】D**

**【解析】**

漏洞扫描是指基于漏洞数据库,通过扫描等手段对指定的远程或者本地计算机系统的安全脆弱性进行检测,发现可利用漏洞的一种安全检测(渗透攻击)行为。而不是对入侵行为的检测技术。

本文档由微信号:ruankaopass,一手整理,通过他人购买的,拒绝售后。本人专业提供软考历年真题

以下关于防火墙功能特性的叙述中,不正确的是(11)。

- (11)A. 控制进出网络的数据包和数据流向  
B. 提供流量信息的日志和审计  
C. 隐藏内部 IP 以及网络结构细节  
D. 提供漏洞扫描功能

**【答案】D**

**【解析】**

防火墙是被动防御,无法提供系统漏洞扫描

某软件公司项目组的程序员在程序编写完成后均按公司规定撰写文档,并上交公司存档。此情形下,该软件文档著作权应由(12)享有。

- (12)A. 程序员      B. 公司与项目组共同  
C. 公司      D. 项目组全体人员

**【答案】C**

**【解析】**

本题考查知识产权知识,即软件知识产权归属。公民为完成法人或者其他组织工作任务所创作的作品是职务作品。职务作品可以是作品分类中的任何一种形式,如文字作品、电影作品、计算机软件都可能由于为执行工作任务而创作,属于职务作品。其著作权归属分两种情形:(1)一般职务作品:一般职务作品的著作权由作者享有。单位或其他组织享有在其业务范围内优先使用的权利,期限为2年。单位的优先使用权是专有的,未经单位同意,作者不得许可第三人以与单位使用的相同方式使用该作品。在作品完成两年内,如单位

在其业务范围内不使用，作者可以要求单位同意由第三人以与单位使用的相同方式使用，所获报酬，由作者与单位按约定的比例分配。（2）特殊的职务作品：所谓特殊职务作品一是指是利用法人或者其他组织的物质技术条件创作，并由法人或者其他组织承担责任的工程设计、产品设计图纸、地图、计算机软件等职务作品；二是指法律、行政法规规定或者合同约定著作权由法人或者其他组织享有的职务作品。对于特殊职务作品，作者享有署名权，其他权利由法人或非法人单位享有。

本题涉及软件知识产权，王某为完成公司指派的工作任务所开发的软件是职务软件，职务软件包括软件文档和源程序。该软件文档属于特殊职务作品，依据著作权法，对于特殊职务作品，除署名权以外，著作权的其他权利由公司享有。

我国商标法规定了申请注册的商标不得使用的文字和图形，其中包括县级以上行政区的地名(文字)。以下商标注册申请，经审查，能获准注册的商标是(13)。

- (13)A. 青岛(市)      B. 黄山(市)      C. 海口(市)      D. 长沙(市)

**【答案】B**

**【解析】**

李某购买了一张有注册商标的应用软件光盘，则李某享有(14)。

- (14)A. 注册商标专用权      B. 该光盘的所有权  
C. 该软件的著作权      D. 该软件的所有权

**【答案】B**

**【解析】**

软件开发具有软件的著作权和所有权、收益权。光盘的购买者具有光盘的所有权和软件的使用权。

某医院预约系统的部分需求为：患者可以查看医院发布的专家特长介绍及其就诊时间；系统记录患者信息，患者预约特定时间就诊。用 DFD 对其进行功能建模时，患者是(15)；用 ERD 对其进行数据建模时，患者是(16)。

- (15)A. 外部实体      B. 加工      C. 数据流      D. 数据存储  
(16)A. 实体      B. 属性      C. 联系      D. 弱实体

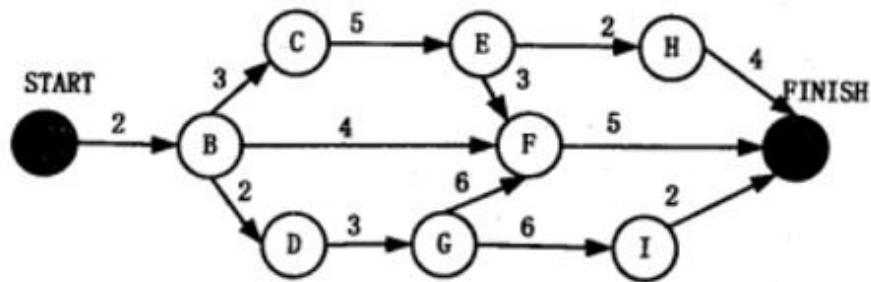
**【答案】A A**

**【解析】**

DFD(数据流图)的实体,可以是人、物或其他软件系统。

ERD 弱实体是指存在和标识需要依赖于其他实体的实体。比如成绩需要依赖于某个课程。

某软件项目的活动图如下图所示,其中顶点表示项目里程碑,链接顶点的边表示包含的活动,变色数字表示活动的持续时间(天)。完成该项目的最少时间为(17)天。由于某种原因,现在需要同一个开发人员完成BC和BD,则完成该项目的最少时间为(18)天。



(17) A. 11                      B. 18                      C. 20                      D. 21

(18) A. 11                      B. 18                      C. 20                      D. 21

**【答案】B C**

**【解析】**

此类关键路径计算必须掌握,由于计算过程比较复杂,看免费视频学习

<http://edu.51cto.com/center/course/lesson/index?id=73796> 此链接可复制,从 56 分处开始看。

某企业财务系统的需求中,属于功能需求的是(19)。

- (19) A. 每个月特定的时间发放员工工资  
B. 系统的响应时间不超过 3 秒  
C. 系统的计算精度符合财务规则的要求  
D. 系统可以允许 100 个用户同事查询自己的工资

**【答案】A**

**【解析】**

功能需求:系统要做什么,在何时做,在何时以及如何修改升级。

更适合用来开发操作系统的编程语言是\_(20)。

- (20) A. C/C++                      B. Java                      C. Python                      D. JavaScript

**【答案】A**

**【解析】**

Linux 就是用 C 开发的

以下关于程序设计语言的叙述中，不正确的是\_(21)。

- (21) A. 脚本语言中不使用变量和函数  
B. 标记语言常用于描述格式化和链接  
C. 脚本语言采用解释方式实现  
D. 编译型语言的执行效率更高

**【答案】A**

**【解析】**

脚本语言中可以使用变量

将高级语言源程序通过编译或解释方式进行翻译时，可以先生成与源程序等价的某种中间代码。以下关于中间代码的叙述中，正确的是\_(22)。

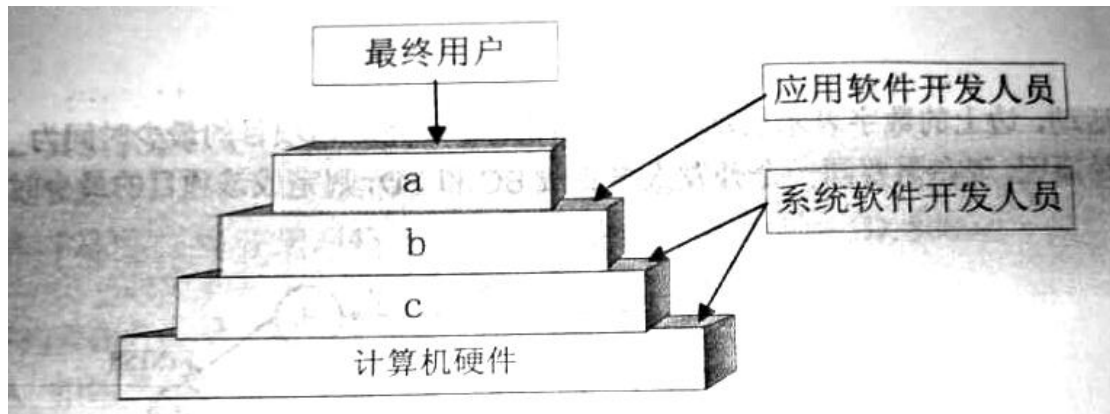
- (22) A. 中间代码常采用符号表来表示  
B. 后缀式和三地址码是常用的中间代码  
C. 对中间代码进行优化要依据运行程序的机器特性  
D. 中间代码不能跨平台

**【答案】B**

**【解析】**

中间代码的作用是可使程序的结构在逻辑上更为简单明确，特别是可使目标代码的优化比较容易实现。中间代码有多种形式，常见的有逆波兰记号（后缀式）、四元式和三元式，它们的共同特点是与具体的机器无关，不依赖于具体的计算机。

计算机系统的层次结构如下圈所示，基于硬件之上的软件可分为 a、b 和 c 三个层次。图中 a、b 和 c 分别表示\_(23)。



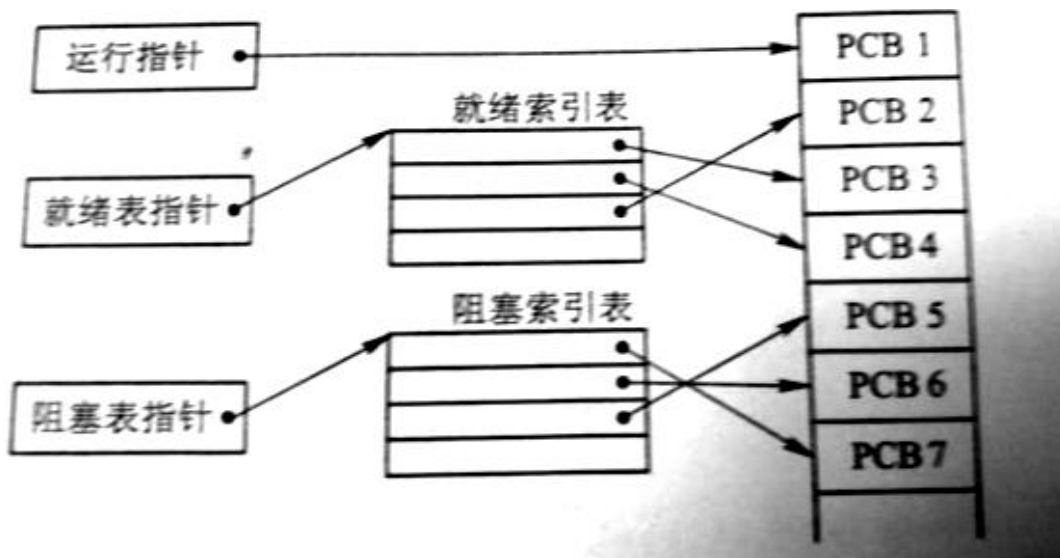
- (23) A. 操作系统、系统软件和应用软件      B. 操作系统、应用软件和系统软件  
C. 应用软件、系统软件和操作系统      D. 应用软件、操作系统和系统软件

【答案】C

【解析】

系统软件开发人员开发操作系统和系统软件。应用软件开发人员开发应用软件。

下图所示的 PCB（进程控制块）的组织方式是 (24)，图中 (25)。



- (24) A. 链接方式      B. 索引方式      C. 顺序方式      D. Hash

- (25) A. 有 1 个运行进程、2 个就绪进程、4 个阻塞进程  
B. 有 1 个运行进程、2 个就绪进程、4 个阻塞进程  
C. 有 1 个运行进程、3 个就绪进程、3 个阻塞进程  
D. 有 1 个运行进程、4 个就绪进程、2 个阻塞进程

【答案】B C

【解析】



都给出索引表了，这就不用说了；运行指针就指向了一个进程，就绪有三个，阻塞有三个。

某文件系统采用多级索引结构。若磁盘块的大小为 1K 字节，每个块号占 3 字节，那么采用二级索引时的文件最大长度为 (26) 字节。

(26) A. 1024

B. 2048

C. 116281

D. 232562

【答案】C

【解析】

磁盘块的大小为 1K 字节，每个块号占 3 字节，一共有 341 个磁盘块存一级索引文件，每个磁盘块还可以指向 341 个磁盘块，因此一共有  $341 \times 341 = 116281$  个磁盘块存二级索引文件，每个磁盘块 1K 字节，则该二级索引文件\*\*\*长度为 11628K 字节。

某操作系统采用分页存储管理方式，下图给出了进程 A 和进程 B 的页表结构。如果物理页的大小为 1K 字节，那么进程 A 中逻辑地址为 1024 (十进制) 用变量存放在 (27) 号物理内存页中。假设进程 A 的逻辑页 4 与进程 B 的逻辑页 5 要共享物理页 4，那么应该在进程 A 页表的逻辑页 4 和进程 B 页表的逻辑页 5 对应的物理页处分别填 (28)。

进程 A 页表		进程 B 页表		物理页
逻辑页	物理页	逻辑页	物理页	0
0	8	0	1	1
1	3	1	6	2
2	5	2	9	3
3	2	3	7	4
4		4	0	5
5		5		6
				7
				8
				9

(27) A. 8

B. 3

C. 5

D. 2

(28) A. 4、4

B. 4、5

C. 5、4

D. 5、5

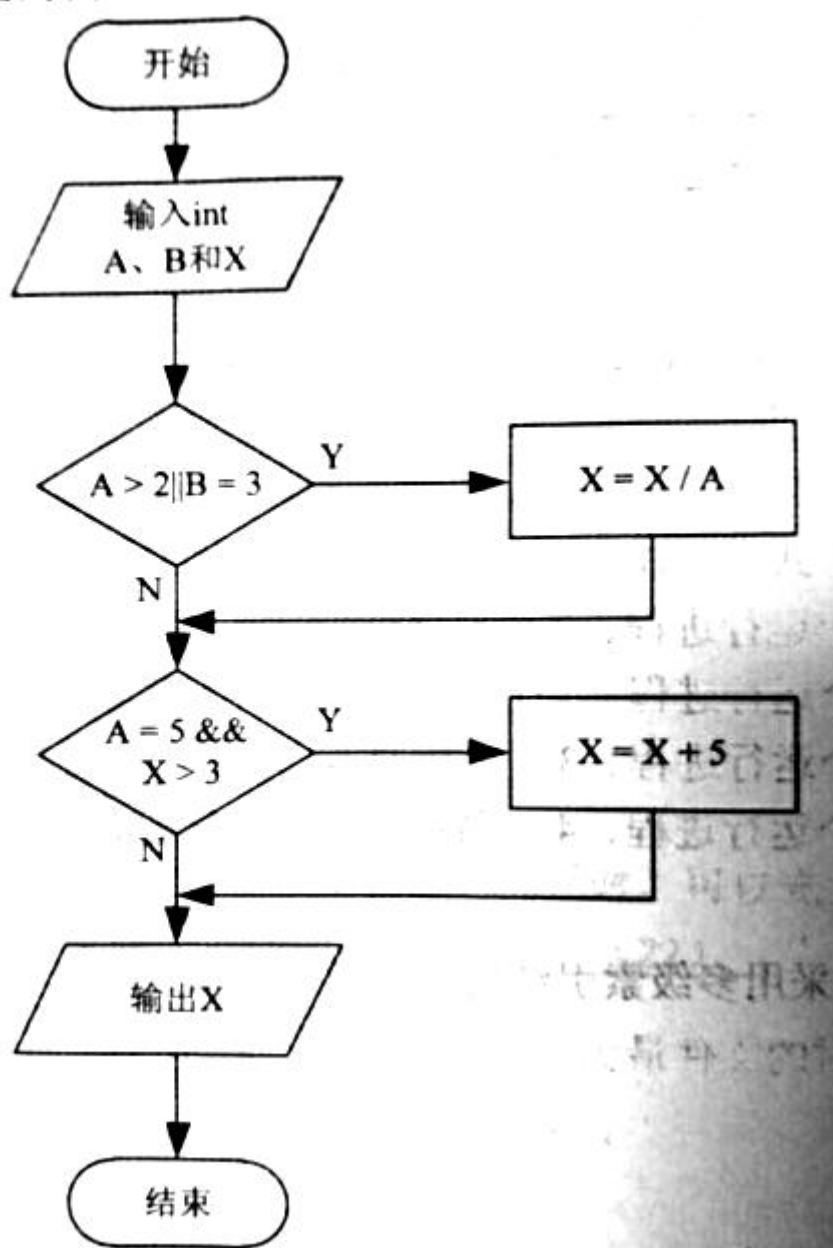
【答案】B A

【解析】

物理页大小=逻辑页大小等于 1K，逻辑地址为 1024 的进程存放在逻辑页的第 1 页，对

应物理页号是 3。进程 A 的逻辑页 4 与进程 B 的逻辑页 5 要共享物理页 4，则它们对应的物理页号都是 4。

用自盒测试方法对如下图所示的流程图进行测试。若要满足分支覆盖，则至少要 (29) 个测试用例，正确的测试用例对是 (30) (测试用例的格式为 (A, B, X; X))。



(29) A. 1

B. 2

C. 3

D. 4

(30) A. (1, 3, 3;3) 和 (5, 2, 15;3)

B. (1, 1, 5;5) 和 (5, 2, 20;9)

C. (2, 3, 10;5) 和 (5, 2, 18;3)

D. (5, 2, 16;16) 和 (5, 2, 21;9)

【答案】B BD(30 两个答案)

**【解析】**

先用代入法求出第 30 题，然后判断第 29 题需要两个测试用例即可。

配置管理贯穿软件开发的整个过程。以下内容中，不属于配置管理的是(31)。

- (31) A. 版本控制                      B. 风险管理                      C. 变更管理                      D. 配置状态报告

**【答案】B**

**【解析】**

配置管理是通过技术或行政手段对软件产品及其开发过程和生命周期进行控制、规范的一系列措施。配置管理的目标是记录软件产品的演化过程，确保软件开发者在软件生命周期中的各个阶段都能得到精确的产品配置。

配置管理的活动包括：制定配置管理计划、配置库管理、配置控制、配置状态报告、配置审计、发布管理和交付。

风险管理是与配置管理并列的项目管理过程

极限编程（XP）的十二个最佳实践不包括(32)。

- (32) A. 小的发布                      B. 结对编程                      C. 持续集成                      D. 精心设计

**【答案】D**

**【解析】**

极限编程（XP）

4 大价值观：沟通，简单性，简单，反馈与勇气；

5 个原则：快速反馈，简单性假设，逐步修改，提倡更改和优质工作

12 个最佳实践：计划游戏（快速制定计划、随着细节的不断变化而完善）、小型发布（系统的设计要能够尽可能早地交付）、隐喻（找到合适的比喻传达信息）、简单设计（只处理当前的需求，使设计保持简单）、测试现行（先写测试代码，然后再编写程序）、重构（重新审视需求和设计，重新明确地描述它们以符合新的和现有的需求）、结对编程、集体代码所有制、持续集成（可以按日甚至按小时为客户提供可运行的版本）、每周工作 40 个小时、现场客户和编码标准。

以下关于管道过滤器体系结构的有点的叙述中，不正确的是(33)。

- (33) A. 软件构件具有良好的高内聚、低耦合的特点

- B. 支持重用
- C. 支持并行执行
- D. 提高性能

【答案】D

【解析】

管道/过滤器体系结构的优点：

- (1) 使得软件构件具有良好的隐蔽性和高内聚、低耦合的特点；
- (2) 允许设计者将整个系统的输入/输出行为看成是多个过滤器的行为的简单合成；
- (3) 支持软件重用。提供适合在两个过滤器之间传送的数据，任何两个过滤器都可被连接起来；
- (4) 系统维护和增强系统性能简单。新的过滤器可以添加到现有系统中来；旧的可以被改进的过滤器替换掉；
- (5) 允许对一些属性如吞吐量、死锁等的分析；
- (6) 支持并行执行。每个过滤器是作为一个单独的任务完成，因此可与其他任务并行执行。

模块 A 将学生信息，即学生姓名、学号、手机号等放到一个结构体中，传递给模块 B。  
模块 A 和 B 之间的耦合类型为 (34) 耦合。

- (34) A. 数据                      B. 标记                      C. 控制                      D. 内容

【答案】B

【解析】

数据耦合。模块间通过参数传递基本类型的数据，称为数据耦合。

标记耦合。模块间通过参数传递复杂的内部数据结构，称为标记耦合。如高级语言的数组名、记录名、文件名等这些名字即为标记，其实传递的是这个数据结构的地址。

某模块内涉及多个功能，这些功能必须以特定的次序执行，则该模块的内聚类型为 (35) 内聚。

- (35) A. 实践                      B. 过程                      C. 信息                      D. 功能

【答案】B

【解析】

内聚是一个模块内部各成分之间相关联程度的度量。

内聚按强度从低到高有以下几种类型：

(1) 偶然内聚

如果一个模块的各成分之间毫无关系，则称为偶然内聚，也就是说模块完成一组任务，这些任务之间的关系松散，实际上没有什么联系。

(2) 逻辑内聚

几个逻辑上相关的功能被放在同一模块中，则称为逻辑内聚。如一个模块读取各种不同类型外设的输入。尽管逻辑内聚比偶然内聚合理一些，但逻辑内聚的模块各成分在功能上并无关系，即使局部功能的修改有时也会影响全局，因此这类模块的修改也比较困难。

(3) 时间内聚

如果一个模块完成的功能必须在同一时间内执行（如系统初始化），但这些功能只是因为时间因素关联在一起，则称为时间内聚。

(4) 通信内聚

如果一个模块的所有成分都操作同一数据集或生成同一数据集，则称为通信内聚。

(5) 顺序内聚

如果一个模块的各个成分和同一个功能密切相关，而且一个成分的输出作为另一个成分的输入，则称为顺序内聚。

(6) 过程内聚

过程内聚是指构件或者操作的组合方式是，允许在调用前面的构件或操作之后，马上调用后面的构件或操作，即使两者之间没有数据进行传递。

(7) 信息内聚

模块完成多个功能，各个功能都在同一数据结构上操作，每一项功能有一个唯一的入口点。这个模块将根据不同的要求，确定该模块执行哪一个功能。由于这个模块的所有功能都是基于同一个数据结构（符号表），因此，它是一个信息内聚的模块。

(8) 功能内聚

模块的所有成分对于完成单一的功能都是必须的，则称为功能内聚。

系统交付用户使用后，为了改进系统的图形输出而对系统进行修改的维护行为属于(36)维护。

(36)A. 改正性

B. 适应性

C. 改善性

D. 预防性

【答案】C

**【解析】**

软件的维护内容主要有以下几个方面：

- (1) 改正性维护是指改正在系统开发阶段已发生而系统测试阶段尚未发现的错误。
- (2) 适应性维护是指使应用软件适应信息技术变化和管理需求变化而进行的修改。
- (3) 完善性维护是为扩充功能和改善性能而进行的修改，主要是指对已有的软件系统增加一些在系统分析和设计阶段中没有规定的功能与性能特征。
- (4) 预防性维护是为了改进应用软件的可靠性和可维护性，为了适应未来的软/硬件环境的变化，应主动增加预防性的新的功能，以使应用系统适应各类变化而不被淘汰。

在面向对象方法中，将逻辑上相关的数据以及行为绑定在一起，使信息对使用者隐蔽称为 (37)。当类中的属性或方法被设计为 private 时，(38) 可以对其进行访问。

(37) A. 抽象                      B. 继承                      C. 封装                      D. 多态

(38) A. 应用程序中所有方法  
B. 只有此类中定义的方法  
C. 只有此类中定义的 public 方法  
D. 同一个包中的类中定义的方法

**【答案】C B**

**【解析】**

逻辑上相关的数据以及行为绑定在一起，使信息对使用者隐蔽称为封装。对于私有成员来说只能是该类中定义的方法才能对其进行访问。

采用继承机制创建子类时，子类中 (39)。

(39) A. 只能有父类中的属性                      B. 只能有父类中的行为  
C. 只能新增行为                      D. 可以有新的属性和行为

**【答案】D**

**【解析】**

在令子类继承父类的同时，可以重新定义某些属性，并重写某些方法，即覆盖父类的原有属性和方法，使其获得与父类不同的功能。另外，为子类追加新的属性和方法也是常见的做法。

面向对象分析过程中，从给定需求描述中选择 (40) 来识别对象。

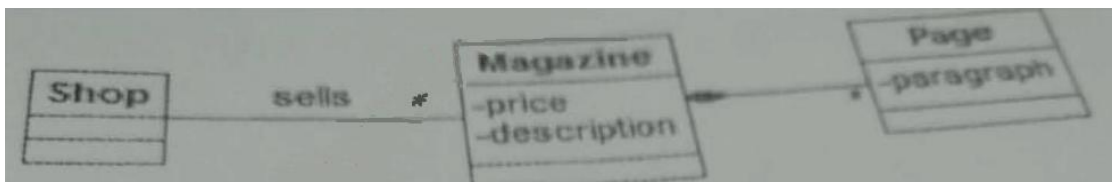
- (40) A. 动词短语      B. 名词短语      C. 形容词      D. 副词

【答案】B

【解析】

在面向对象分析过程中，选择名词短语来识别对象，动词短语识别对象的操作。

如图所示的 UML 类图中，Shop 和 Magazine 之间为 (41) 关系，Magazine 和 Page 之间为 (42) 关系。UML 类图统称不用于对 (43) 进行建模。



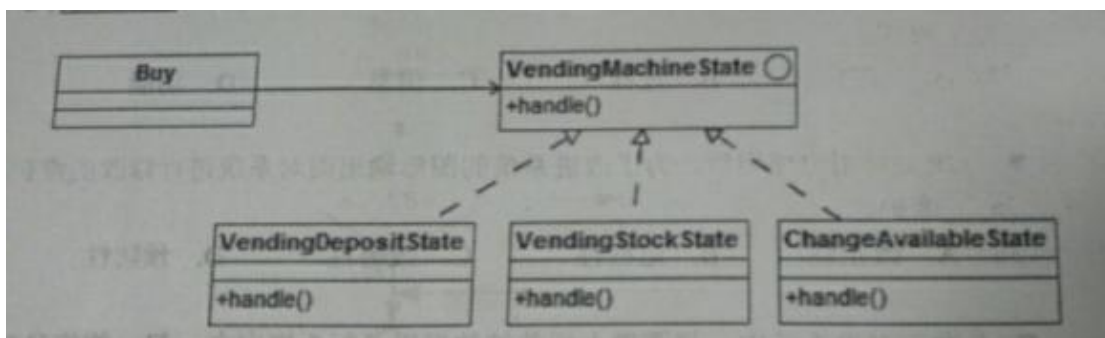
- (41) A. 关联      B. 依赖      C. 组合      D. 继承  
(42) A. 关联      B. 依赖      C. 组合      D. 继承  
(43) A. 系统的词汇      B. 简单的协作  
C. 逻辑数据库模式      D. 对象快照

【答案】A C D

【解析】

第一个空和第二个空，可以通过他们之间的连线得出；第三空：对系统的词汇建模；对简单的协作建模；对逻辑数据库模式建模都可以用到类图

自动售货机根据库存、存放货币量、找零能力、所选项目等不同，在货币存入并进行选择时具有如下行为：交付产品不找零：交付产品找零：存入货币不足而不提供任何产品；库存不足而不提供任何产品。这一业务需求适合采用 (44) 模式设计实现，其类图如下图所示，其中 (45) 是客户程序使用的主要接口，可用状态来对其进行配置。此模式为 (46)，体现的最主要的意图是 (47)。



- (44) A. 观察者 (Observer)                      B. 状态 (State)  
       C. 策略 (Strategy)                      D. 访问者 (Visitor)
- (45) A. VendingMachineState                  B. Buy  
       C. VendingDepositState                D. VendingStockState
- (46) A. 创建型对象模式                      B. 结构型对象模式  
       C. 行为型类模式                      D. 行为型对象模式
- (47) A. 当一个对象状态改变时所有依赖它的对象得到通知并自动更新  
       B. 在不破坏封装性的前提下，捕获对象的内部状态并在对象之外保存  
       C. 一个对象在其内部状态改变时改变其行为  
       D. 将请求封装为对象从而可以使用不同的请求对客户进行参数化

【答案】B A D C

【解析】

有 State 关键词 状态模式：允许一个对象在其内部状态改变时改变它的行为。对象看起来似乎修改了它的类。在很多情况下，一个对象的行为取决于一个或多个动态变化的属性，这样的属性叫做状态，这样的对象叫做有状态的 (stateful) 对象，这样的对象状态是从事先定义好的一系列值中取出的。当一个这样的对象与外部事件产生互动时，其内部状态就会改变，从而使得系统的行为也随之发生变化。

是整个状态模式的基类也是对外的接口

编译过程中进行的语法分析主要是分析 (48)。

- (48) A. 源程序中的标识符是否合法              B. 程序语句的含义是否合法  
       C. 程序语句的结构是否合法              D. 表达式的类型是否合法

【答案】C

【解析】



编译程序的工作过程一般划分为五个阶段：词法分析、语法分析、语义分析、优化、目标代码生成。

(1) 词法分析：也就是从左到右一个一个地读入源程序，识别一个单词或符号，并进行归类。

(2) 语法分析：在词法分析的基础上，将单词序列分解成各类语法短语，如“程序”，“语句”，“表达式”等。

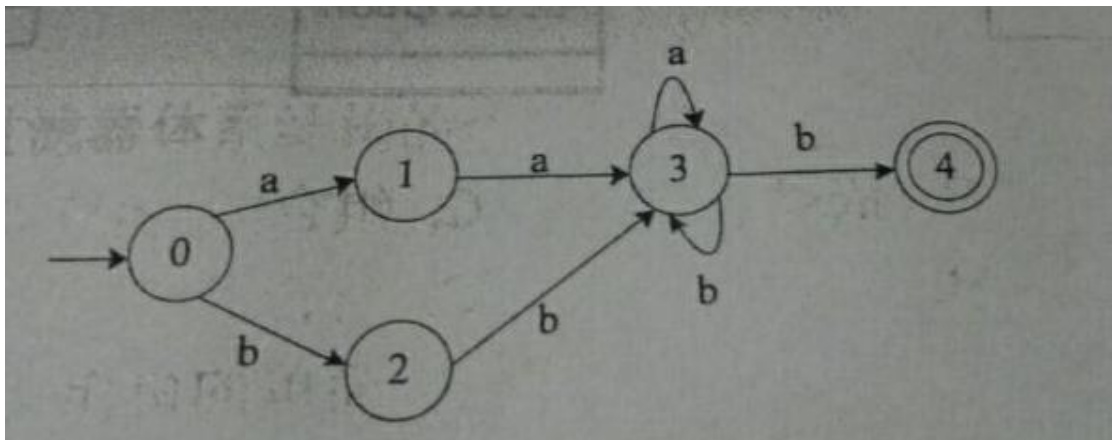
(3) 语义分析：审查源程序是否有语义的错误，当不符合语言规范的时候，程序就会报错。

(4) 代码优化：这个阶段是对前阶段的中间代码进行变换或改造，目的是使生成的目标代码更为高效，即节省时间和空间。

(5) 目标代码生成：也就是把优化后的中间代码变换成指令代码或汇编代码。

词法分析和语法分析本质上都是对源程序的结构进行分析。

某确定的有限自动机 (DFA) 的状态转换图如下图所示 (0 是初态, 4 是终态), 则该 DFA 能识别 (49)。



(49) A. aaab

B. abab

C. bbba

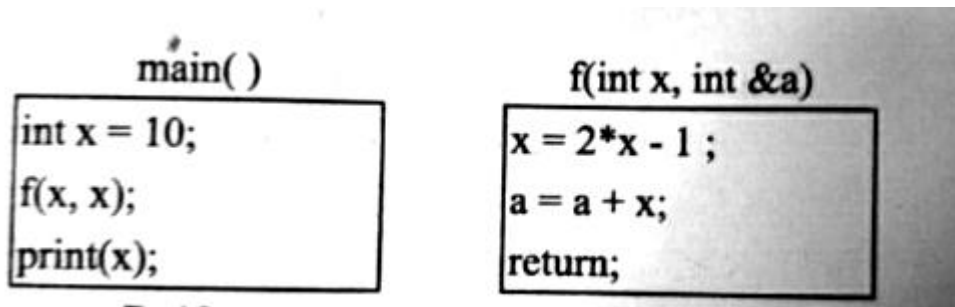
D. abba

**【答案】A**

**【解析】**

将选项分别带入判断。

函数 `main()`、`f()` 的定义如下所示。调用函数 `f()` 时, 第一个参数采用传值 (call by value) 方式, 第二个参数采用传引用 (call by referen) 方式, 则函数 `main()` 执行后输出的值为 (50)。



(50) A. 10                      B. 19                      C. 20                      D. 29

【答案】D

【解析】

x=19, a=29。因为 a 属于传引用，\*\*\*结果是 29。

传值与传引用的区别是：值传递时，实参被拷贝了一份，只能在函数体内使用。因此题干中的 x=19，只能作用于 f 函数中。

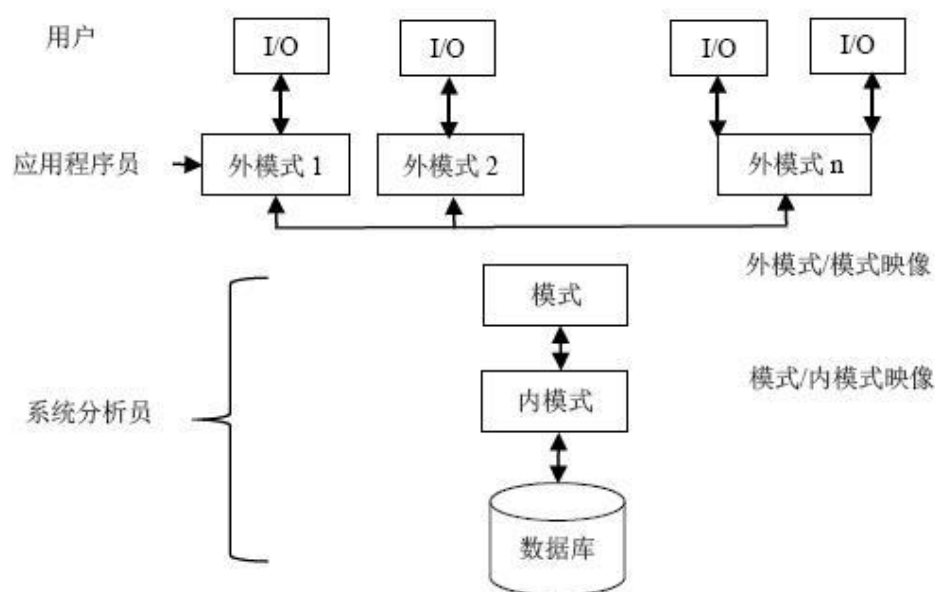
采用三级结构/两级映像的数据库体系结构，如果对数据库的一张表创建聚簇索引，改变的是数据库的(51)。

(51) A. 用户模式                      B. 外模式                      C. 模式                      D. 内模式

【答案】D

【解析】

数据库采用“三级模式两级映像”的结构：



(1) 概念模式也称模式，是数据库中全部数据的整体逻辑结构的描述。

(2) 外模式也称用户模式或子模式，是用户与数据库系统的接口，是用户用到的那部分数据的描述。

(3) 内模式也称存储模式，是数据库在物理存储方面的描述，定义所有内部记录类型、索引和文件的组织方式，以及数据控制方面的细节。

(4) 两级映像：

① 模式/内模式映像存在于概念级和内部级之间，用于定义概念模式和内模式之间的对应性；

② 外模式/模式映像存在于外部级和概念级之间，用于定义外模式和概念模式之间的对应性。

聚簇索引也叫簇类索引，它对磁盘上实际数据重新组织以按指定的一个或多个列的值排序，改变的是数据库的内模式。

某企业的培训关系模式 R(培训科目，培训师，学生，成绩，时间，教室)，R 的函数依赖集  $F = \{\text{培训科目} \rightarrow \text{培训师}, (\text{学生}, \text{培训科目}) \rightarrow \text{成绩}, (\text{时间}, \text{教室}) \rightarrow \text{培训科目}, (\text{时间}, \text{培训师}) \rightarrow \text{教室}, (\text{时间}, \text{学生}) \rightarrow \text{教室}\}$ 。关系模式 R 的主键为 (52)，其规范化程度最高达到 (53)。

(52) A. (学生，培训科目)

B. (时间，教室)

C. (时间，培训师)

D. (时间，学生)

(53) A. 1NF

B. 2NF

C. 3NF

D. BCNF

**【答案】D B**

**【解析】**

(52) A. (学生，培训科目) B. (时间，教室) 该属性组就可以成为一个主键。选项 D (时间，学生) 的组合中，(时间，学生)  $\rightarrow$  教室，(时间，教室)  $\rightarrow$  培训科目，培训科目  $\rightarrow$  培训师，(学生，培训科目)  $\rightarrow$  成绩，可见，包括了关系模式 R 的全部属性，因此 (时间，学生) 是该关系模式的主键。

数据库的关系范式有以下几种。

(1) \*\*\*范式 (1NF)：属性不可拆分或无重复的列。

(2) 第二范式 (2NF)：完全函数依赖。在一个关系中，若某个非主属性数据项依赖于全部关键字称之为完全函数依赖。例：成绩表 (学号，课程号，成绩) 关系中，(学号，课程号)  $\rightarrow$  成绩，只有学号或者只有课程号都无法推导出成绩，因此该关系式属于完全函数依赖。

(3) 第三范式 (3NF)：消除传递依赖。不依赖于其他非主属性 (消除传递依赖)。满足第三

范式的数据库必须先满足第二范式。也就是说，数据库中的属性依赖仅能依赖于主属性，不存在于其他非主属性的关联。例如，图书，图书室的关系。图书包括编号、出版商、页码等信息，图书室包括图书室编号、所存图书（外键）。其中，图书室的表中不应该存储任何图书的具体信息（例如，出版商。），而只能通过主键图书编号来获得对应图书的信息。

（4）BC 范式（BCNF）：所有非主属性对每一个码都是完全函数依赖；所有的主属性对于每一个不包含它的码，也是完全函数依赖；没有任何属性完全函数依赖于非码的任意一个组合。如果说关系模式 R 属于 3NF，不一定属于 BCNF，如果 R 属于 BCNF，则一定属于 3NF。

（5）第四范式（4NF）：对于每一个  $X \rightarrow Y$ ，X 都能找到一个候选码（若关系中的某一属性组的值能唯一地表示一个元组，而其真子集不行，则称该属性组为候选码）。

题干中“(时间，教室)→培训科目”且“培训科目→培训师”，这之间存在传递依赖，不满足 3NF，属于 2NF。

设关系模式 R (U, F)，其中：U= {A, B, C, D, E}，F={A→B, DE→B, CB→E, E→A, B→D}。(54) 为关系模式 R 的候选关键字。分解 (55) 是无损连接，并保持函数依赖的。

(54) A. AB                      B. DE                      C. DB                      D. CE

(55) A.  $\rho = \{ R_1 (AC), R_2 (ED), R_3 (B) \}$   
B.  $\rho = \{ R_1 (AC), R_2 (E), R_3 (DB) \}$   
C.  $\rho = \{ R_1 (AC), R_2 (ED), R_3 (AB) \}$   
D.  $\rho = \{ R_1 (ABC), R_2 (ED), R_3 (ACE) \}$

【答案】D    D

【解析】

第 54 题选项 D 中，E→A，A→B，B→D，CB→E 包括了全部的属性。

第 55 题判断无损连接有多种方法，下面介绍一种构造二维表进行判断的方法（以正确选项 D 为例）：

① 构造一个初始的二维表，若“属性”属于“模式”中的属性，则填 a<sub>j</sub>，否则填 b<sub>ij</sub>。

模式 \ 属性	A	B	C	D	E
ABC	a1	a2	a3	b14	b15
ED	b21	b22	b23	a4	a5
ACE	a1	b32	a3	b34	a5

② 逐一考察关系式中的函数依赖，将表中的  $b_{ij}$  修改成  $a_j$ 。如  $CB \rightarrow E$ ，则上表中\*\*\*行第六列的  $b_{15}$ ，修改成  $a_5$ 。修改结果如下表所示。

模式 \ 属性	A	B	C	D	E
ABC	a1	a2	a3	a4	a5
ED	a1	a2	b23	a4	a5
ACE	a1	a2	a3	b34	a5

③ 修改后的表格若任一行存在  $a_1, a_2, a_3, a_4, a_5$  的情况，该分解就属于无损连接，如上表格中\*\*\*行存在  $a_1, a_2, a_3, a_4, a_5$ ，因此属于无损连接。

在基于 Web 的电子商务应用中，访问存储于数据库中的业务对象的常用方式之一是(56)。

(56) A. JDBC                      B. XML                      C. CGI                      D. COM

【答案】A

【解析】

JDBC(Java DataBase Connectivity, java 数据库连接)是一种用于执行 SQL 语句的 JavaAPI，可以为多种关系数据库提供统一访问，它由一组用 Java 语言编写的类和接口组成。JDBC 提供了一种基准，据此可以构建更高级的工具和接口，使数据库开发人员能够编写数据库应用程序，简单地说，JDBC 可做三件事：与数据库建立连接、发送 操作数据库的语句并处理结果。

XML 可扩展标记语言是一种元标记语言，它可以对文档和数据进行结构化处理，从而能够在部门、客户和供应商之间进行交换，实现动态内容生成，企业集成和应用开发。XML 简化了网络中数据交换和表示，使得代码、数和表示分离，并作为数据交换的标准格式，因此它常被称为智能数据文档。

Common Gateway Interface，公共网关接口，简称 CGI。是 WWW 技术中最重要的技术之一，有着不可替代的重要地位。CGI 是外部应用程序（CGI 程序）与 WEB 服务器之间的接口标准，是在 CGI 程序和 Web 服务器之间传递信息的过程。

COM 是 Component Object Model（组件对象模型）的缩写。COM 是微软公司为了计算机工

业的软件生产更加符合人类的行为方式开发的一种新的软件开发技术。在 COM 构架下，人们可以开发出各种各样的功能专一的组件，然后将它们按照需要组合起来，构成复杂的应用系统。

设  $S$  是一个长度为  $n$  的非空字符串，其中的字符各不相同，则其互异的非平凡子串（非空且不同于  $S$  本身）个数为 (57)。

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

**【答案】D**

**【解析】**

任取一个字符串  $abcd$ ，它的非空子集有： $a$ 、 $b$ 、 $c$ 、 $d$ 、 $ab$ 、 $bc$ 、 $cd$ 、 $abc$ 、 $bcd$  共 9 个。

注意： $abd$ 、 $acd$ 、 $bd$ 、 $ad$  等不属于它的子串。

假设某消息中只包含 7 个字符恰  $\{a, b, c, d, e, f, g\}$ ，折 7 个字符在消息中出现的次数为  $\{5, 24, 8, 17, 34, 4, 13\}$ ，利用哈夫曼树（最优二叉树）为该消息中的字符构造符合前缀编码要求的不等长编码。各字符的编码长度分别为 (58)。

- (58) A.  $a:4, b:2, c:3, d:3, e:2, f:4, g:3$                       B.  $a:6, b:2, c:5, d:3, e:1, f:6, g:4$   
C.  $a:3, b:3, c:3, d:3, e:3, f:2, g:3$                       D.  $a:2, b:6, c:3, d:5, e:6, f:1, g:4$

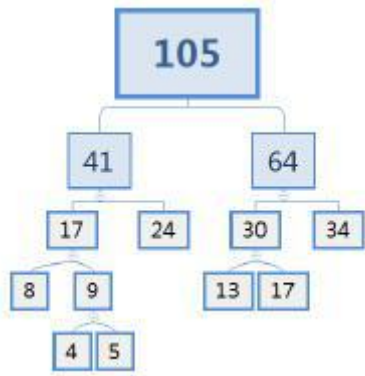
**【答案】A**

**【解析】**

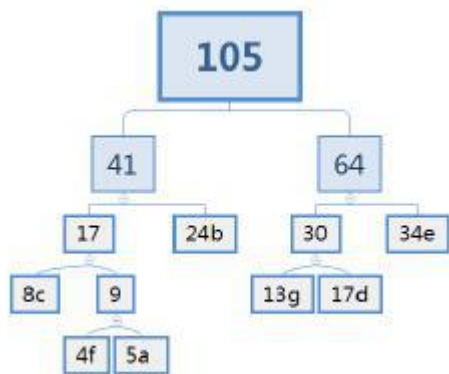
哈夫曼的构造过程如下：

本文档由微信号:ruankaopass，一手整理，通过他人购买的，拒绝售后。本人专业提供软考历年真题

- ① 将给定的序列从小到大进行排列得到  $\{4, 5, 8, 13, 17, 24, 34\}$ ，选序列中最小两个数 4 和 5 画出一个树，得到数字 9。
- ② 将 4 与 5 的和 9，放入序列中，继续重复步骤①，直至画出整个哈夫曼树，如下图所示。



各个字符对应的节点为：



设某二叉树采用二叉链表表示（即结点的两个指针分别指示左、右孩子）。当该二叉树包含  $k$  个节点时，其二叉链表节点中必有 (59) 个空的汉子指针。

(59) A.  $k-1$

B.  $k$

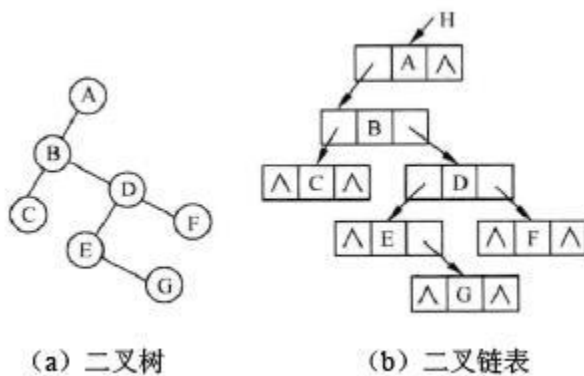
C.  $k+1$

D.  $2k$

【答案】C

【解析】

所谓二叉树的链式存储结构是指，用链表来表示一棵二叉树，即用链来指示着元素的逻辑关系，如下图所示。



二叉链表中每个节点有 2 个指针，一共有  $2k$  个指针。在二叉树中除了根节点之外，其他的

节点都有一条边进入该节点，即一个指针指向该节点，所以二叉树中边的总个数为  $k-1$ ，也就说明非空指针的个数为  $k-1$  个。那么空指针的个数为：总的节点数-非空指针的个数  $=2k-(k-1)=k+1$ 。

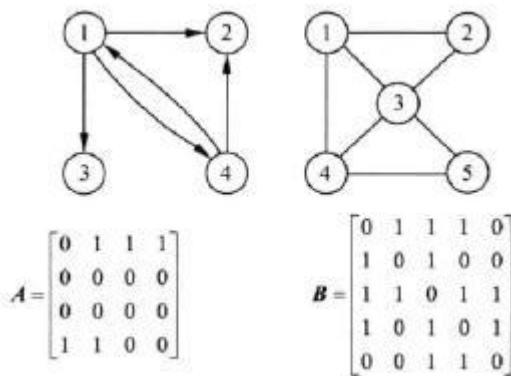
以下关于无向连通图  $G$  的叙述中，不正确的是 (60)。

- (60) A.  $G$  中任意两个顶点之间均有边存在  
 B.  $G$  中任意两个顶点之间存在路径  
 C. 从  $G$  中任意顶点出发可遍历图中所有顶点  
 D.  $G$  的邻接矩阵是对称矩阵

【答案】A

【解析】

有向连通图与无相连通图如下所示：



两个递增序列 A 和 B 的长度分别为  $m$  和  $n$  ( $m < n$  且  $m$  与  $n$  接近)，将二者归并为一个长度为  $m+n$  的递增序列。当元素关系为 (61)，归并过程中元素的比较次数最少。

- (61) A.  $a_1 < a_2 < \dots < a_{m-1} < a_m < b_1 < b_2 < \dots < b_{n-1} < b_n$   
 B.  $b_1 < b_2 < \dots < b_{n-1} < b_n < a_1 < a_2 < \dots < a_{m-1} < a_m$   
 C.  $a_1 < b_1 < a_2 < b_2 < \dots < a_{m-1} < b_{m-1} < a_m < b_m < b_{m+1} < \dots < b_{n-1} < b_n$   
 D.  $b_1 < b_2 < \dots < b_{m-1} < b_m < a_1 < a_2 < \dots < a_{m-1} < a_m < b_{m+1} < \dots < b_{n-1} < b_n$

【答案】A

【解析】

两个递增序列 A、B 进行归并时，从序列的\*\*\*个元素开始，分别从这两个序列中取一



个元素并进行比较，将较小者输出，然后从较小者所在序列取下一个元素再进行比较，循环往复，直到某个序列的全部元素已经输出，再将另一个序列的剩余元素依次输出即可。

若  $a_m < b_1$ ，则需要依次比较  $a_1$  与  $b_1$ ， $a_2$  与  $b_1$ ， $a_3$  与  $b_1$ ， $a_{m-1}$  与  $b_1$ ， $a_m$  与  $b_1$  共需要  $m$  次比较，这是归并时比较次数最少的情况。

求解两个长度为  $n$  的序列  $X$  和  $Y$  的一个最长公共序列（如序列 ABCBDAB 和 BDCABA 的一个最长公共子序列为 BCBA）可以采用多种计算方法。如可以采用蛮力法，对  $X$  的每一个子序列，判断其是否也是  $Y$  的子序列，最后求出最长的即可，该方法的时间复杂度为 (62)。经分析发现该问题具有最优子序列，可以定义序列成都分别为  $i$  和  $j$  的两个序列  $X$  和  $Y$  的最长公共子序列的成都为  $C[i, j]$ ，如下式所示。该方法的时间复杂度为 (63)。

$$C[i, j] = \begin{cases} 0 & \text{若 } i=0 \text{ 或 } j=0 \\ C[i-1, j-1]+1 & \text{若 } i, j > 0 \text{ 且 } X_i = Y_j \\ \max(C[i-1, j], C[i, j-1]) & \text{其它} \end{cases}$$

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

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

【答案】D A

【解析】

第 62 题一个给定的序列的子序列，就是将给定序列中零个或多个元素去掉之后得到的结果。序列 ABCBDAB 去掉划线的字符 ABCBDAB 得到的 BCBA 就是它的一个子序列。一个

序列的子序列有  $2^n$  个，将  $X$ 、 $Y$  的所有子序列都检查过后即可求出  $X$ 、 $Y$  的最长公共子序列。这种方式的时间复杂度为  $O(n2^n)$ 。

第 63 题引进一个二维数组  $c[i][j]$  表示  $X$  的  $i$  位和  $Y$  的  $j$  位之前的最长公共子序列的长度，按照公式将数值填入下表。

$\begin{matrix} & j \\ i \end{matrix}$		0	1	2	3	4	5	6
		Y1	B	D	C	A	B	A
0	x1	0	0	0	0	0	0	0
1	A	0	0	0	0	1	1	1
2	B	0	1	1	1	1	2	2
3	C	0	1	1	2	2	2	2
4	B	0	1	1	2	2	3	3
5	D	0	1	2	2	2	3	3
6	A	0	1	2	2	3	3	4
7	B	0	1	2	2	3	4	4

此时， $c[i][j]$ 中\*\*\*的数就是 X 和 Y 的最长公共子序列的长度，等于 4。该算法的时间复杂度为  $O(n^2)$ 。

现需要对一个基本有序的数组进行排序。此时最适宜采用的算法为 (64) 排算法，时间复杂度为 (65)。

- (64) A. 插入                      B. 快速                      C. 归并                      D. 堆
- (65) A.  $O(n)$                       B.  $O(n \lg n)$                       C.  $O(n^2)$                       D.  $O(n^2 \lg n)$

【答案】A A

【解析】

插入排序对基本有序的数组排序速度快；插入排序的时间复杂度都是  $O()$

相比于 TCP，UDP 的优势为 (66)。

- (66) A. 可靠传输                      B. 开销较小                      C. 拥塞控制                      D. 流量控制

【答案】B

【解析】

UDP 与 TCP 的主要区别在于 UDP 不一定提供可靠的数据传输。事实上，该协议不能保证数据准确无误地到达目的地。UDP 在许多方面非常有效，在数据传输过程中延迟小、数据传输效率高。当某个程序的目标是尽快地传输尽可能多的信息时，可使用 UDP。

若一台服务器只开放了 25 和 110 两个端口，那么这台服务器可以提供 (67) 服务。

- (67) A. E-Mail                      B. WEB                      C. DNS                      D. FTP

【答案】A

【解析】

25 端口是为 SMTP (Simple Mail Transfer Protocol, 简单邮件传输协议) 服务器开放的，

主要用于发送邮件。

110 端口是为 POP3（邮件协议 3）服务开放的，POP2、POP3 都是主要用于接收邮件的，目前 POP3 使用的比较多，许多服务器都同时支持 POP2 和 POP3。客户端可以使用 POP3 协议来访问服务端的邮件服务

本文档由微信号:ruankaopass，一手整理，通过他人购买的，拒绝售后。本人专业提供软考历年真题

SNMP 是一种异步请求/响应协议，采用\_(68)\_协议进行封装。

(68)A. IP                      B. ICMP                      C. TCP                      D. UDP

**【答案】D**

**【解析】**

简单网络管理协议(SNMP)，由一组网络管理的标准组成，包含一个应用层协议(application layer protocol)、数据库模型(database schema)和一组资源对象。该协议能够支持网络管理系统，用以监测连接到网络上的设备是否有任何引起管理上关注的情况。SNMP 使用 UDP（用户数据报协议）作为第四层协议（传输协议），进行无连接操作。

在一台安装好 TCP/IP 协议的计算机上，当网络连接不可用时，为了测试编写好的网络程序，通常使用的目的主机 IP 地址为\_(69)。

(69)A. 0. 0. 0. 0              B. 127. 0. 0. 1              C. 10. 0. 0. 1              D. 210. 225. 21. 255/24

**【答案】B**

**【解析】**

127. 0. 0. 1 是回送地址，指本地机，一般用来测试使用。回送地址（127. x. x. x）是本机回送地址（Loopback Address），即主机 IP 堆栈内部的 IP 地址，主要用于网络软件测试以及本地机进程间通信，无论什么程序，一旦使用回送地址发送数据，协议软件立即返回，不进行任何网络传输。

测试网络连通性通常采用的命令是\_(70)。

(70)A. Nstar                      B. Ping                      C. Mscinfug                      D. Cmd

**【答案】B**

**【解析】**

Netstat 是控制台命令，是一个监控 TCP/IP 网络的非常有用的工具，它可以显示路由表、实际的网络连接以及每一个网络接口设备的状态信息。Netstat 用于显示与 IP、TCP、UDP 和 ICMP 协议相关的统计数据，一般用于检验本机各端口的网络连接情况。

Ping 是一个通信协议，是 TCP/IP 协议的一部分。利用“ping”命令可以检查网络是否连通，可以很好地帮助分析和判定网络故障。

msconfig 即系统配置实用程序，是 Microsoft System Configuration 的缩写。是在开始菜单里运行中输入然后确认就可以找到程序开启或者禁用，可以帮助电脑禁止不需要运行的程序，加快电脑运行。

cmd 命令提示符是在操作系统中，提示进行命令输入的一种工作提示符。在不同的操作系统环境下，命令提示符各不相同。

The development of the Semantic Web proceeds in steps, each step building a layer on top of another. The pragmatic justification for this approach is that it is easier to achieve (71) on small steps, whereas it is much harder to get everyone on board if too much is attempted. Usually there are several research groups moving in different directions; this (72) of ideas is a major driving force for scientific progress. However, from an engineering perspective there is a need to standardize. So, if most researchers agree on certain issues and disagree on others, it makes sense to fix the point of agreement. This way, even if the more ambitious research efforts should fail, there will be at least (73) positive outcomes.

Once a (74) has been established, many more groups and companies will adopt it, instead of waiting to see which of the alternative research lines will be successful in the end. The nature of the Semantic Web is such that companies and single users must build tools, add content, and use that content. We cannot wait until the full Semantic Web vision materializes—it may take another ten years for it to be realized to its full (75) (as envisioned today, of course).

- |                    |              |                |                  |
|--------------------|--------------|----------------|------------------|
| (71)A. conflicts   | B. consensus | C. success     | D. disagreement  |
| (72)A. competition | B. agreement | C. cooperation | D. collaboration |
| (73)A. total       | B. complete  | C. partial     | D. entire        |
| (74)A. technology  | B. standard  | C. pattern     | D. model         |

(75) A. area

B. goal

C. object

D. extent

【答案】C A C A D

【解析】

语义网目前正在稳步发展中，每次在已经完成的基础之上又增加一些内容。客观来说这种方式会成功是因为实现一个小的(71)比较容易，而迈出一大步实现所有功能则比较困难。研究开发者们通常会按照不同方向进行开发工作，这种(72)的思想是促进科学进步的重要驱动力。然而就软件工程的观点来说，应该把开发过程标准化，若是大家就某些观点能达成共识，而其他的问题有不同的意见，那么就很有必要落实这些已经达成共识的部分。这么做的话，就算是那些野心勃勃的开发者在开发的道路上遭遇了滑铁卢，整个行业还是会产生(73)积极成果。这个(74)一旦被建立，就会有更多的开发者采用它，而不是观望哪种研究路线会取得最终的胜利。语义网的本质是企业 and 单个用户必须构建工具，添加内容并使用。我们已经迫不及待语义网真正实现的那一天——也许(75)实现它的内容需要再过十年时间(当然这是按照今天所设想的那样)。

(71) A. 冲突 B. 一致同意 C. 成功 D. 异议

(72) A. 竞争 B. 同意 C. 合作 D. 协作

(73) A. 全部 B. 完整 C. 部分 D. 整体

(74) A. 技术 B. 标准 C. 方法 D. 模型

(75) A. 局部 B. 目的 C. 目标 D. 程度

## 试题一

阅读下列说明和图，回答问题 1 至问题 4，将解答填入答题纸的对应栏内。

### 【说明】

某公司拟开发一个共享单车系统，采用北斗定位系统进行单车定位，提供针对用户的 APP 以及微信小程序、基于 Web 的管理与监控系统。该共享单车系统的主要功能如下。

1) 用户注册登录。用户在 APP 端输入手机号并获取验证码后进行注册，将用户信息进行存储。用户登录后显示用户所在位置周围的单车。

2) 使用单车。

①扫码/手动开锁。通过扫描二维码或手动输入编码获取开锁密码，系统发送开锁指令进行开锁，系统修改单车状态，新建单车行程。

②骑行单车。单车定时上传位置，更新行程。

③锁车结账。用户停止使用或手动锁车并结束行程后，系统根据已设置好的计费规则及使用时间自动结算，更新本次骑行的费用并显示给用户，用户确认支付后，记录行程的支付状态。系统还将重置单车的开锁密码和单车状态。

3) 辅助管理。

①查询。用户可以查看行程列表和行程详细信息。

②报修。用户上报所在位置或单车位置以及单车故障信息并进行记录。

4) 管理与监控。

①单车管理及计费规则设置。商家对单车基础信息、状态等进行管理，对计费规则进行设置并存储。

②单车监控。对单车、故障、行程等进行查询统计。

③用户管理。管理用户信用与状态信息，对用户进行查询统计。现采用结构化方法对共享单车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。

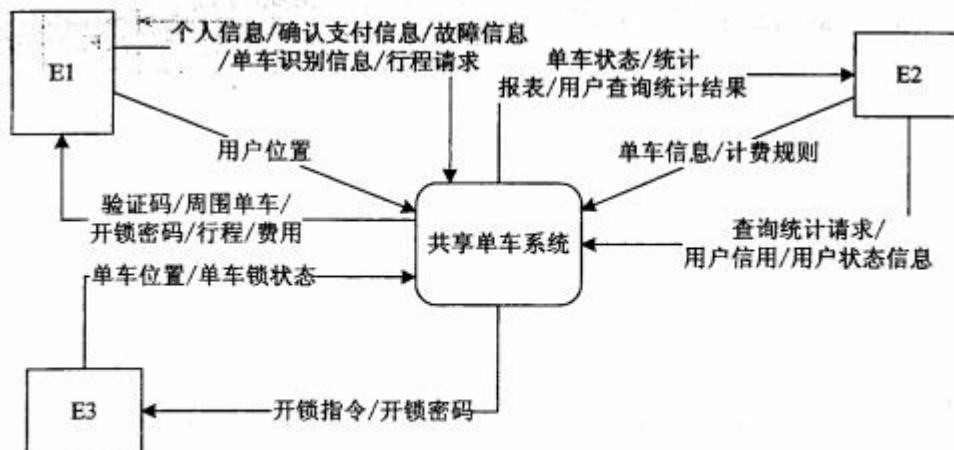
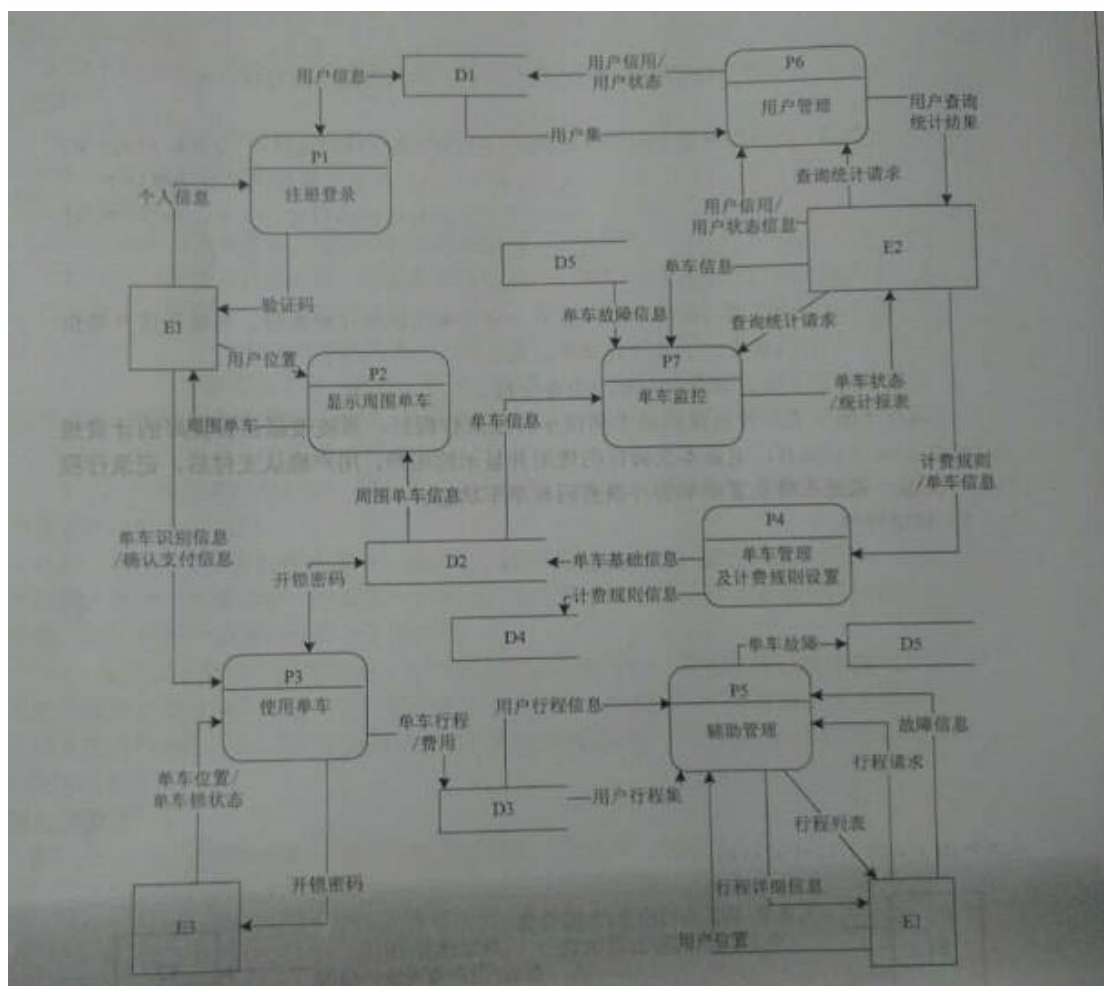


图 1-1 上下文数据流图



【问题 1】(3 分)

使用说明中的词语，给出图 1-1 中的实体 E1~E3 的名称。

E1: 用户 E2: 商家 E3: 单车

**【问题 2】**（5 分）

使用说明中的词语，给出图 1-2 中的数据存储 D1~D5 的名称。

D1: 存储用户信息/用户信息表

D2: 记录单车位置及状态信息/单车信息表

D3: 记录单车行程信息/行程信息表

D4: 存储计费规则/计费规则信息表

D5: 记录单车故障信息/单车故障信息表

**【问题 3】**（5 分）

根据说明和图中术语及符号，补充图 1-2 中缺失的数据流及其起点和终点。

起点: p3 终点: E1 数据流名称: 开锁密码

起点: p3 终点: E1 数据流名称: 锁车结账

起点: p3 终点: D2 数据流名称: 单车状态

起点: D4 终点: P3 数据流名称: 计费规则

起点: D3 终点: p7 数据流名称: 行程查询统计

**【问题 4】**（2 分）

根据说明中术语，说明“使用单车”可以分解为哪些子加工？

扫码/手动开锁，骑行单车，锁车结帐



## 试题二

阅读下列说明，回答问题 1 至 4 问题，将解答填入答题纸的对应栏内。

### 【说明】

M 公司为了便于开展和管理各项业务活动，提高公司的知名度和影响力，拟构建一个基于网络的会议策划系统。

### 【需求分析结果】

该系统的部分功能及初步需求分析的结果如下：

(1) M 公司旗下有业务部、策划部和其他部门。部门信息包括部门号、部门名、主管、联系电话和邮箱号；每个部门只有一名主管，只负责管理本部门的工作，且主管参照员工关系的员工号；一个部门有多名员工，每名员工属于且仅属于一个部门。

(2) 员工信息包括员工号、姓名、职位、联系方式和薪资。职位包括主管、业务员、策划员等。业务员负责受理用户申请，设置受理标志。一名业务员可以受理多个用户申请，但一个用户申请只能由一名业务员受理。

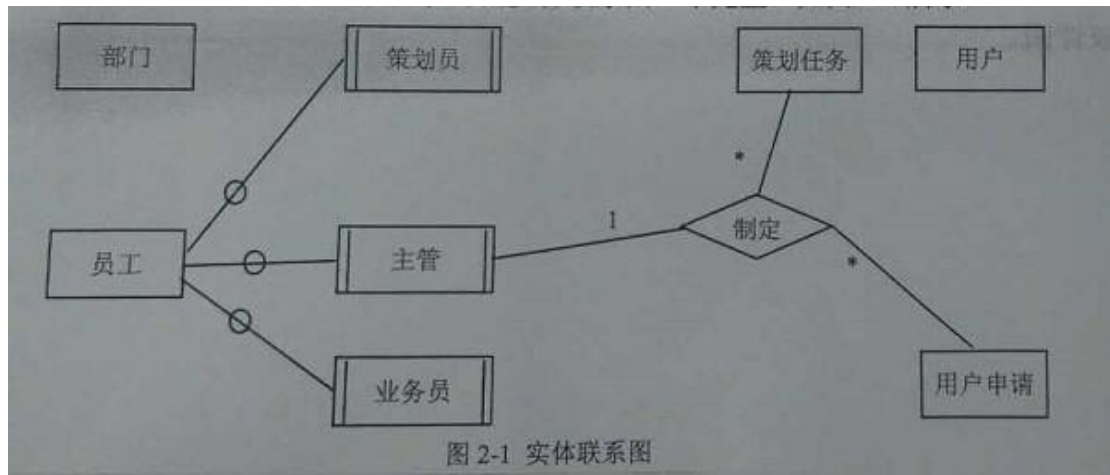
(3) 用户信息包括用户号、用户名、银行账号、电话、联系地址。用户号唯一标识用户信息中的每一个元组。

(4) 用户申请信息包括申请号、用户号、会议日期、天数、参会人数、地点、预算和受理标志。申请号唯一标识用户申请信息中的每一个元组，且一个用户可以提交多个申请，但一个用户申请只对应一个用户号。

(5) 策划部主管为已受理的用户申请制定会议策划任务。策划任务包括申请号、任务明细和要求完成时间。申请号唯一标识策划任务的每一个元组。一个策划任务只对应一个已受理的用户申请，但一个策划任务可由多名策划员参与执行，且一名策划员可以参与执行，且在项策划任务。

### 【概念模型设计】

根据需求阶段收集的信息，设计的实体联系图（不完整）如图 2-1 所示。



### 【关系模型设计】

部门（部门号，部门名，部门主管，联系电话，邮箱号）

员工（员工号，姓名，（ ），联系方式，薪资）

用户（用户名，（ ），电话，联系地址）

用户申请（申请号，用户号，会议日期，天数，参会人数，地点，受理标志，（ ））

执行（申请号，任务明细，（ ））

### 【问题 1】（5 分）

根据问题描述，补充五个联系，完善图 2-1 的实体联系图。联系名可用联系 1、联系 2、联系 3、联系 4 和联系 5，联系的类型为 1:1、1: n 和 m: n（或 1:1、1: \*和\*: \*）。

联系 1：部门和员工，1:n

联系 2：业务员和用户申请，1:n

联系 3：用户和用户申请，1:n

联系 4：策划员和策划任务，m:n

联系 5：策划任务和用户申请，1:1

### 【问题 2】（4 分）

根据题意，将关系模型中的空（a）~（d）补充完整，并填入答题纸对应的位置上。

1. 职位，部门号
2. 用户名，银行帐号
3. 预算费用，业务员/员工号
4. 要求完成时间

**【问题 3】(4 分)**

给出“用户申请”和“策划任务”关系模式的主键和外键。

用户申请：主键：申请号 外键：用户号，业务员

策划任务：主键：申请号 外键：申请号

**【问题 4】(2 分)**

请问“执行”关系模式的主键为全码的说法正确吗？为什么？

不正确。

All-key 关系模型的所有属性组成该关系模式的候选码，称为全码。即所有属性当作一个码。

若关系中只有一个候选码，且这个候选码中包含全部属性，则该候选码为全码。

策划任务中的任务明细和要求完成时间不包括全部属性，因此不是全码。

### 试题三

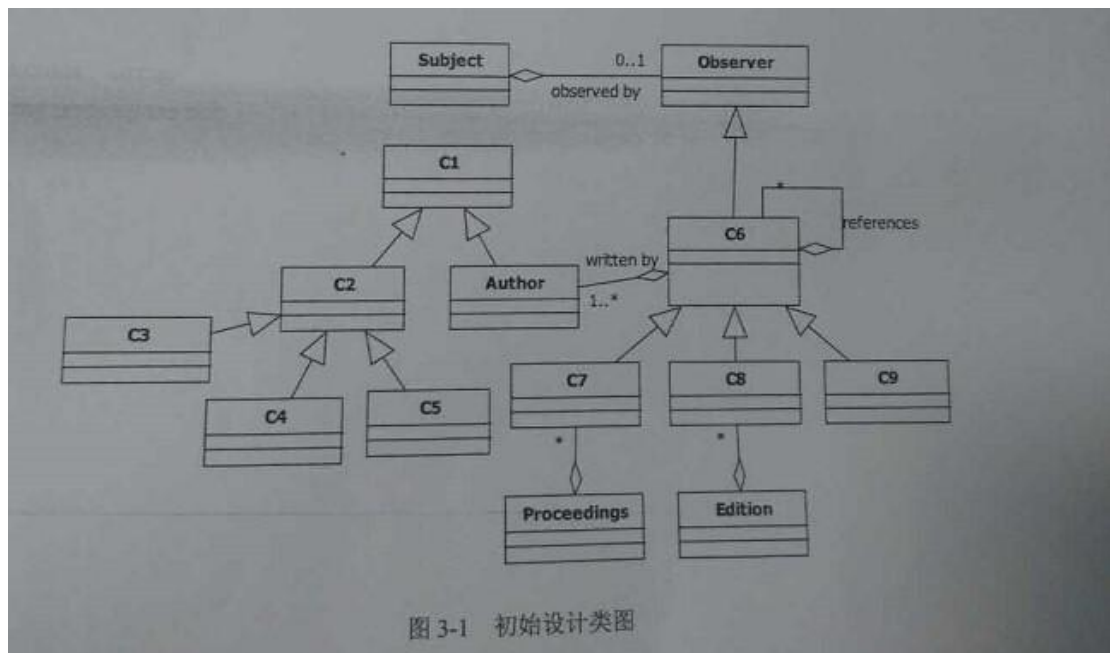
阅读下列说明，回答问题 1 至问题 3，将解答填入答题纸的对应栏内。

#### 【说明】

某大学拟开发一个用于管理学术出版物(Publication) 的数字图书馆系统，用户可以从该系统查询或下载已发表的学术出版物。系统的主要功能如下：

1. 登录系统。系统的用户 (User) 仅限于该大学的学生 (Student) 、教师 (Faculty) 和其他工作人员 (Staff) 。在访问系统之前，用户必须使用其校园账户和密码登录系统。
2. 查询某位作者 (Author) 的所有出版物。系统中保存了会议文章 (ConfPaper)、期刊文章 (JurnalArticle) 和校内技术报告 (TechReport ) 等学术出版物的信息，如题目、作者以及出版年份等。除此之外，系统还存储了不同类型出版物的一些特有信息：
  - (1) 对于会议文章，系统还记录了会议名称、召开时间以及召开地点；
  - (2) 对于期刊文章，系统还记录了期刊名称、出版月份、期号以及主办单位；
  - (3) 对于校内技术报告，系统记录了由学校分配的唯一 ID。
3. 查询指定会议集 (Proceedings) 或某个期刊特定期 (Edition) 的所有文章。会议集包含了发表在该会议 (在某个特定时间段、特定地点召开) 上的所有文章。期刊的每一期在特定时间发行，其中包含若干篇文章。
4. 下载出版物。系统记录每个出版物被下载的次数。
5. 查询引用了某篇出版物的所有出版物。在学术出版物中引用他人或早期的文献作为相关工作或背景资料是很常见的现象。用户也可以在系统中为某篇出版物注册引用通知，若有新的出版物引用了该出版物，系统将发送电子邮件通知该用户。

现在采用面向对象方法对该系统进行开发，得到系统的初始设计类图如图 3-1 所示。



【问题 1】（9 分）

根据说明中的描述，给出图 3-1 中 C1~C9 所对应的类名。

C1:Object

C2:User

C3:Student

C4:Faculty

C5:Staff

C6:Publication

C7:ConfPaper

C8:JournalArticle

C9:TechReport

【问题 1】（4 分）

根据说明中的描述，给出图 3-1 中类 C6~C9 的属性。

C6:题目，作者，出版年份

C7:会议名称，召开时间，召开地点

C8:期刊名称，出版月份，期号，主办单位

C9:ID

**【问题3】**（2分）

图 3-1 中包含了哪种设计模式？实现的是该系统的哪个功能？

观察者模式，实现：引用他人学术出版物发送电子邮件通知该用户。

#### 试题四

阅读下列说明和 C 代码，回答问题 1 至问题 2，将解答写在答题纸的对应栏内。

##### 【说明】

一个无向连通图  $G$  点上的哈密尔顿 (Hamilton) 回路是指从图  $G$  上的某个顶点出发，经过图上所有其他顶点一次且仅一次，最后回到该顶点的路径。一种求解无向图上哈密尔顿回路算法的基础如下：

假设图  $G$  存在一个从顶点  $V_0$  出发的哈密尔顿回路  $V_1 \text{---} V_2 \text{---} V_3 \text{---} \dots \text{---} V_{n-1} \text{---} V_0$ 。算法从顶点  $V_0$  出发，访问该顶点的一个未被访问的邻接顶点  $V_1$ ，接着从顶点  $V_1$  出发，访问  $V_1$  一个未被访问的邻接顶点  $V_2, \dots$ ；对顶点  $V_i$ ，重复进行以下操作：访问  $V_i$  的一个未被访问的邻接点  $V_{i+1}$ ；若  $V_i$  的所有邻接顶点均已被访问，则返回到顶点  $V_{i-1}$ ，考虑  $V_{i-1}$  的下一个未被访问的邻接顶点，仍记为  $V_i$ ；知道找到一条哈密尔顿回路或者找不到哈密尔顿回路，算法结束。

##### 【C 代码】

下面是算法的 C 语言实现。

(1) 常量和变量说明

$n$  : 图  $G$  中的顶点数

$c[i][j]$  : 图  $G$  的邻接矩阵

$K$ : 统计变量，当期已经访问的定点数为  $k+1$

$x[k]$  : 第  $k$  个访问的顶点编号，从 0 开始

$Visited[x[k]]$ : 第  $k$  个顶点的访问标志，0 表示未访问，1 表示已访问

(2) C 程序

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

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

```
#define MAX 100
```

```
Vido Hamilton (int n,int x [MAX],int c[MAX][MAX]) {
```

```
    int i;
```

```
    int visited[MAX];
```

```
    int k;
```

```
    /*初始化 x 数组和 visited 数组*/
```

```

for (i = 0; i < n; i++) {
    x[i] = 0;
    visited [i]=0;
}
/*访问起始顶点*/
k = 0;
(1);
x[0]=0
k = k + 1;
/*访问其他顶点*/
while (k >= 0) {
    x[k]=x[k] + 1;
    while (x[k] < n) {
        if (( 2 ) && c[x[k-1]][x[k] ]==1) { /*邻接顶点 x[k]未被访问过*/
            break;
        } else {
            x[k] = x[k] + 1
        }
    }
    if (x[k] < n && k == n-1 && ( 3 ) ) { /*找到一条哈密尔顿回路*/
        for (k = 0; k < n;k++) {
            printf ("%d--", x[k]) ; /*输出哈密尔顿回路*/
        }
        printf ("%d--", x[0]) ;
        return;
    } else if x[k] < n && k < n-1) { /*设置当期顶点的访问标志，继续下一个顶点
*/
        ( 4 );
        k = k + 1;
    } else { /*没有未被访问过的邻接顶点，回退到上一个顶点*/

```



```

        x[k] = 0;
        visited [x[k]] = 0;
        ( 5 );
    }
}
}

```

**【问题 1】(10 分)**

根据题干说明。填充 C 代码中的空 (1) ~ (5)。

- 1、visited[0] = 1
- 2、visited[x[k]] == 0
- 3、c[x[0]][x[k]]
- 4、visited[x[k]] = 1
- 5、k = k - 1

**【问题 2】(5 分)**

根据题干说明和 C 代码，算法采用的设计策略为 (6)，该方法在遍历图的顶点时，采用的是 (7) 方法（深度优先或广度优先）。

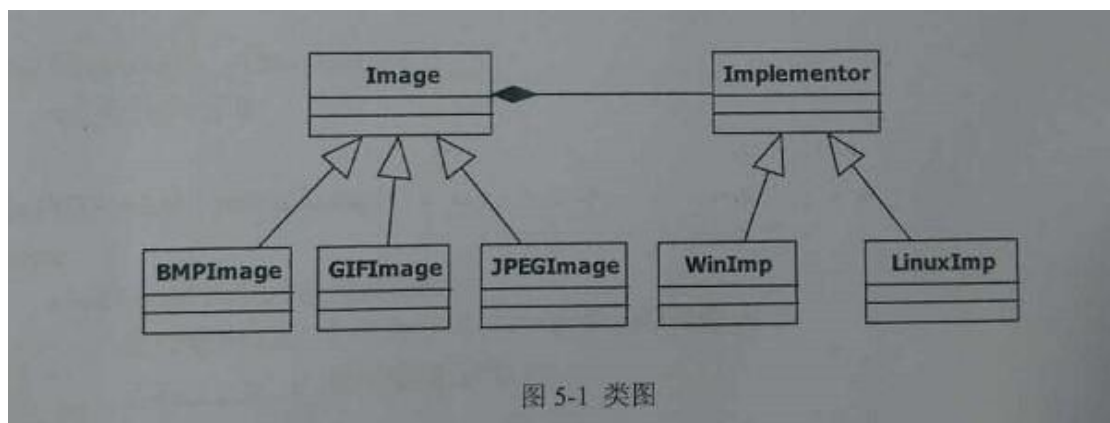
- 6、回溯法
- 7、深度优先

## 试题五

阅读下列说明和 C++代码，将应填入（ ）处的字句写在答题纸的对应栏内。

### 【说明】

某图像预览程序要求能够查看 BMP 、JPEG 和 GIF 三种格式的文件，且能够 Windows 和 Linux 两种操作系统上运行程序需具有较好的扩展性以支持新的文件格式和操作系统。为满足上述需求并减少所需生成的子类数目，现采用桥接（Bridge）模式进行设计，得到如图 5-1 所示的类图。



### 【问题 1】

#### 【C++代码】

```
#include
#include<string> ;
Using namespace std;
class Matrix { //各种格式的文件最终都被转化为像素矩阵
    //此处代码省略
};

class Implementor {
public:
    (1) ; 显示像素矩阵 m
};
```

```

class WinImp:public Implementor {
    public:
        void doPaint (Matrix m) {
            /*调用 Windows 系统的绘制函数绘制像素矩阵*/
        } ;

class LinuxImp:public Implementor { /*调用 Linux 系统的绘制函数绘制像素矩阵*/ }
};

class Image {
    public:
        void setImp (Implement*imp) ? {this->imp=imp;}
        Virtual void parseFile (string fileName) =0
    protected:
        Implementor*imp;
};

class BMPImage:public Image {
    //此处省略代码
};

class GIFImage:public Image {
    Public:
        void parseFile (string fileName)  {
            //此处解析 GIF 文件并获得一个像素矩阵对象 m
            (2); 显示像素矩阵 m
        }
};

class JPEGImage:public Image {
    //此处代码省略

```

```
} ;
```

```
int main () {  
    //在 linux 操作系统上查看 demo.gif 图像文件  
    Image*image= (3) ;  
    Implementor*imageImp= (4) ;  
    (5)  
    Image->parseFile ( "demo.gif " ) ;  
    return 0;  
}
```

1. virtual void doPaint(Matrix m) = 0
2. imp->doPaint(m)
3. new GIFImage()
4. new LinuxImp()
5. image->setImp(imageImp)

## 试题六

阅读下列说明和 Java 代码，将应填入一旦上一处的字句写在答题纸的对应栏内。

### 【说明】

某图像预览程序要求能够查看 BMP 、 JPEG 和 GIF 三种格式的文件，且能够在 Windows 和 Linux 两种操作系统上运行。程序需具有较好的扩展性以支持新的文件格式和操作系统。为满足上述需求并减少所需生成的子类数目，现采用桥接模式进行设计，得到如图 6-1 所示的类图。

本文档由微信号:ruankaopass，一手整理，通过他人购买的，拒绝售后。本人专业提供软考历年真题

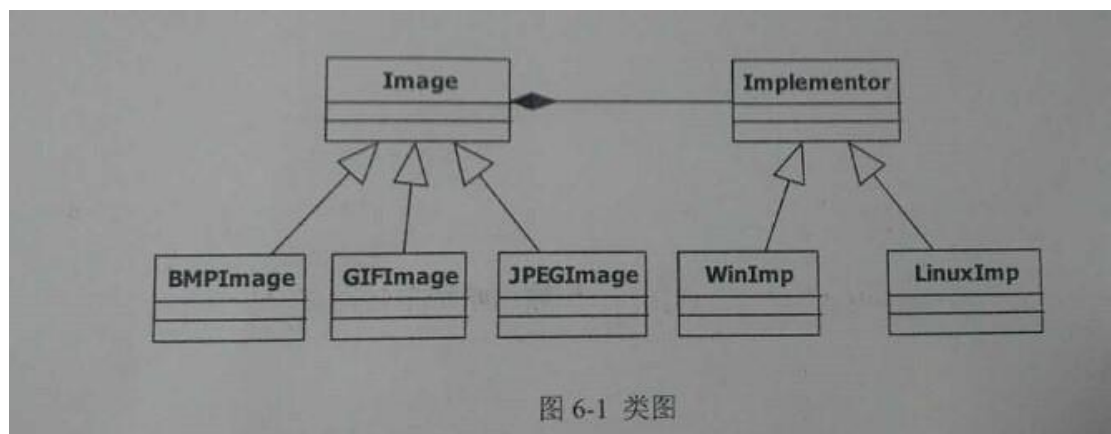


图 6-1 类图

### 【问题 1】

#### 【Java 代码】

```
Import java.util.*;

class Matrix{ //各种格式的文件最终都被转化为像素矩阵
    //此处代码省略
};

abstract class Implementor {
    Public (1) ;//显示像素矩阵 m
};

class WinImp extends Implementor {
    public void doPaint (Matrix m) { //调用 Windows 系统的绘制函数绘制像素矩阵
    }
};
```

```

class LinuxImp extends Implementor{

    public void doPaint (Matrix m) { //调用 Linux 系统的绘制函数绘制像素矩阵

    }

};

abstract class Image {

    public void setImp (Implementor imp) { this.imp= imp; }

    public abstract void parseFile (String fileName) ;

    protected Implementor imp;

} ;

class BMPImage extends Image {

    //此处代码省略

};

class GIFImage extends Image {

    public void parseFile (String fileName)  {

        //此处解析 BMP 文件并获得一个像素矩阵对象 m

        (2); //显示像素矩阵 m

    }

};

Class Main {

    Public static void main (String[]args) {

        //在 Linux 操作系统上查看 demo.gif 图像文件

        Image image= (3)

        Implementor imageImp= (4)

        (5)

        Image.parseFile ( "demo.gif ");

    }

}

```

```
}
```

```
1. abstract void doPaint(Matrix m)
```

```
2. imp. doPaint(m)
```

```
3. new GIFImage()
```

```
4. new LinuxImp()
```

```
5. image.setImp(imageImp)
```