

【软考达人】

软考资料免费获取

- 1、最新软考题库
- 2、软考备考资料
- 3、考前压轴题



微信扫一扫，立马获取



6W+ 免费题库



免费备考资料

PC版题库: ruankaodaren.com

软考设计师模拟试题 2（上午题答案）

●需要一个 $16\text{MB} \times 8$ 位的存储器，现有存储芯片为 $1\text{MB} \times 8$ 位。则需要 (1) 个存储器芯片。存储器芯片的地址长度需要 (2) 位，主存储器的地址长度需要 (3) 位。

(1) A. 16

B. 8

C. 32

D. 24

(2) A. 10

B. 20

C. 27

D. 12

(3) A. 4

B. 27

C. 20

D. 12

【解析】1.需要存储器芯片：；2.存储器芯片的地址长度为 20 位。主存储器的地址长度为 24 位。

●假设一个有 3 个盘片的硬盘，共有 4 个记录面，转速为 7200 转 / 分，盘面有效记录区域的外直径为 30cm，内直径为 10cm，记录位密度为 250 位 / 毫米，磁道密度为 8 道 / 毫米，每磁道分 16 个扇区，每扇区 512 字节，则该硬盘的非格式化容量和格式化容量约为 (4)，数据传输率最接近于 (5)，若一个文件超出一个磁道容量，剩下的部分 (6)。

(4) A. 120MB 和 100MB

B. 30MB 和 25MB

C. 60MB 和 50MB

D. 22.5MB 和 25MB

(5) A. 2356kB / s

B. 3534kB / s

C. 7069kB / s

D. 1178kB / s

(6) A. 存于同一盘面的其他编号的磁道上

B. 存于其他盘面的同一编号的磁道上

C. 存于其他盘面的其他编号的磁道上

D. 存放位置随机

【解析】磁盘容量有两个指标，即非格式化容量和格式化容量。它们可以分别计算如下：

非格式化容量=最大位密度×最内圈周长×总磁道数

在这里位密度为 250 位 / 毫米；内圈周长为 $3.1416 \times 100 = 314.16$ 毫米；

每记录面总磁道数= $(150-50) \times 8 = 800$ 道，则每记录面的非格式化容量为： $314.16 \times 250 \times 800 = 60\text{MB}$

该磁盘有 4 个记录面，则其非格式化容量为： $60\text{Mbit} \times 4 = 240\text{Mbit} = 30\text{MB}$

格式化容量计算公式如下：格式化容量=每道扇区数×扇区容量×总磁道数

$= 16 \times 512 \times 800 \times 4 \div 1024 \div 1024 = 25\text{MB}$

硬盘的平均数据传输率可由下式计算：

平均数据传输率=每道扇区数×扇区容量×盘片转速

根据题中所给出的已知条件和上述计算式，算出平均数据传输率= $16 \times 512 \times 7200 \div 60\text{B} / \text{s} = 960\text{kB} / \text{s}$ 。

该传输率最接近可选的 1178kbit / s。

在题目所给出的条件中可以看到，该硬盘是由多个盘片构成的盘组。该盘组有 4 个记录面构成。同号磁道构成一系列的柱面。因此，在向磁盘记录一个文件时，应将文件尽可能记录在同一柱面上，当一个面记录不下时，再记

录到相邻柱面上。因此，当一个文件超出一个磁道容量时，剩下的部分应存于其他面的同一编号的磁道上，即一柱面的其他磁道上。

●软件能力成熟度模型 CMM(Capability Maturity Model)描述和分析了软件过程能力的发展与改进的程度，确立了一个软件过程成熟程度的分级标准。在初始级，软件过程定义几乎处于无章法可循的状态，软件产品的成功往往依赖于个人的努力和机遇。

在 (7) ，已建立了基本的项目管理过程，可对成本、进度和功能特性进行跟踪。在 (8) ，用于软件管理与工程两方面的软件过程均已文档化、标准化，并形成了整个软件组织的标准软件过程。在已管理级，对软件过程和产品质量有详细的度量标准。

在 (9) ，通过对来自过程、新概念和新技术等方面的各种有用信息的定量分析，能够不断地、持续地对过程进行改进。

(7) A. 可重复级

B. 管理级

C. 功能级

D. 成本级

(8) A. 标准级

B. 已定义级

C. 可重复级

D. 优化级

(9) A. 分析级

B. 过程级

C. 优化级

D. 管理级

【解析】事实表明，在无规则和混乱的管理条件下，先进的技术和工具并不能发挥应有的作用。人们认识到，改进软件过程的管理是解决上述难题的突破口，不能忽视软件过程的影响。但是各个软件机构的过程成熟度有着较大的差别，为了做出客观、公正的比较，需要建立一种衡量的标准。使用这个标准一方面可以评价软件承包机构的质量保证能力，在软件项目评标活动中，选择中标机构；另一方面该标准也必然成为软件机构改进软件质量，加强质量管理，以及提高软件产品质量的依据。1987 年美国卡内基-梅隆大学软件工程研究所受国防部资助，提出了软件机构的能力成熟度模型。该模型将软件由低到高分 5 个级别：初始级、可重复级、已定义级、已管理级和优化级。

1)初始级的软件过程是未知定义的随意过程，项目的执行随意甚至混乱，规范未能覆盖基本的关键过程要求。2)可重复级仅定义了管理的基本过程，而没有定义执行的步骤标准。3)已定义级要求制定企业范围的工业标准，管理和工程开发都有一套文档化的标准。4)已管理级中所有过程均建立了相应的度量方式，所有产品的质量(包括工作产品和提交给用户的产品)需有明确的质量指标。5)优化级是达到一个持续改进的境界，企业过程能够根据反馈信息进行不断的优化。

●我国著作权法中对公民作品的发表权的保护期限是 (10) 。

(10) A. 作者有生之年加死后 50 年

B. 作品完成后 50 年

C. 没有限制

D. 作者有生之年

【解析】《中华人民共和国著作权法》第二十一条规定："公民的作品，其发表权、本法第十条第一款第(五)项至第(十七)项规定的权利的保护期为作者终生及其死亡后五十年，截止于作者死亡后第五十年的 12 月 31 日；如果是合作作品，截止于最后死亡的作者死亡后第五十年的 12 月 31 日。法人或者其他组织的作品、著作权(署名权除外)由法人或者其他组织享有的职务作品，其发表权、本法第十条第一款第(五)项至第(十七)项规定的权利的保护期为五十年，截止于作品首次发表后第五十年的 12 月 31 日，但作品自创作完成后五十年内未发表的，本法不再保护。"

● (11) 不是 ERP 的发展阶段。

- (11) A. MRP 阶段
B. MRPII 阶段
C. 闭环 MEP 阶段
D. OLAP

【解析】此题是考查信息系统的 ERP 的发展史。OLAP 是联机事务分析系统，它不属于 ERP 的发展阶段。MRP、MRP II 和 ERP，是企业管理信息系统发展的不同阶段。MPR 主要对制造环节中的物流进行管理，使企业达到“既要保证生产又要控制库存”的目的；而 MRP II 则集成了物流和资金流，将人、财、物，时间等各种资源进行周密计划，合理利用，以提高企业的竞争力；ERP 的概念则由 Garter Group 率先提出，它将供应链、企业业务流程和信息流程都囊括其中。

● 下列不属于人工智能技术研究领域的是 (12) 。

- (12) A. 专家系统
B. 自然语言理解
C. 人工神经网络
D. 计算机辅助设计

【解析】此题是人工智能问题。因为计算机辅助设计是一种提高计算机设计效率的一种工具。并不是人工智能知识。“人工智能”(Artificial Intelligence)简称 AI。它是研究、开发用于模拟、延伸和扩展人的智能的理论、方法、技术及应用系统的一门新的技术科学。

● 虚拟存储管理系统的基础是程序的 (13) 理论。这个理论的基本含义是当程序执行时往往会 (14) 访问内存。程序的 (13) 表现在 (15) 和 (16) 上。(15) 是指最近被访问的存储单元可能马上又要被访问。(16) 是指马上被访问的单元，而其附近的单元也可能马上被访问。根据这个理论，Denning 提出了工作集理论。工作集是进程运行时被频繁地访问的页面集合。在进程运行时，如果它的工作页面都在 (17) 器内，能够使进程有效地运行，否则会出现频繁的页面调入 / 调出现象。

- (13) A. 局部性
B. 全局性
C. 动态性
D. 虚拟性
- (14) A. 频繁地
B. 均匀地
C. 不均匀地
D. 全面地
- (15) ,(16) A. 数据局部性
B. 空间局部性
C. 时间局部性
D. 数据全局性
- (17) A. 外部存储
B. 主存储
C. 辅助存储
D. 虚拟存储

【解析】本题考查虚拟存储管理系统知识，虚拟存储管理系统的基础是程序的局部性原理。所谓程序局部性原理是指程序在执行时所呈现的局部性规律，即在一段较短时间内，程序的执行仅限于某个部分。相应地，它所访问的存储器空间也局限在某个空间。

局部性原理又表现为两个方面。

(1)时间局限性。如果某条指令被执行，则不久以后该指令很可能再次被执行；如果某条数据结构被访问，则不久以后该数据结构很可能再次被访问。产生时间局限性的主要原因是程序中有大量的循环操作。

(2)空间局限性。一旦程序访问了某个内存单元，不久以后，其附近的内存单元也要被访问，即程序在一段时间内所访问的存储器空间可能集中在一定的范围之内，其最常见情况就是程序的顺序执行。

工作集是指在某段时间间隔内，进程所要访问的页面集合。虽然程序只需少量的几页内存就可以运行，但为了使程序更有效地运行，必须使程序的工作集全部在内存(主存储器)当中，否则会使进程在运行中频繁出现缺页中断，从而出现频繁的页面调入 / 调出现象。

● (18) 是将系统化的、规范的、可量化的方法应用于软件的开发、运行和维护的过程，它包括方法、工具和过程 3 个要素。ISO9000 是由 ISO/TC176 制定的关于 (19) 和质量保证的国际标准。CMM 提供了一个框架，将软件过程改进的进化步骤组织成 5 个成熟度等级。除第 1 级外，每个等级都包含了实现该成熟度等级目标的若干 (20) 。在软件开发的各资源中，(21) 是最重要的资源。软件的复杂性是 (22) ，它引起人员通信困难、开发费用超支、开发时间超时等问题。

(18) A. 软件过程

B. 软件测试

C. 软件工程

D. 信息工程

(19) A. 质量控制

B. 质量分析

C. 质量管理

D. 质量改进

(20) A. 关键过程域

B. 关键实践

C. 软件过程

D. 软件过程能力

(21) A. 人员

B. 方法

C. 硬件环境

D. CASE 工具

(22) A. 固有的

B. 人为的

C. 可消除的

D. 不可降低的

【解析】软件工程是研究和应用如何以系统化的、规范的、可度量的方法去开发、运行和维护软件，即把工程化应用到软件上。软件工程是一种层次化的技术，其中过程、方法和工具是软件工程的 3 个要素。ISO9000 不是特指一个标准，而是一个标准序列，是由 ISO / TC176 制定的关于质量管理和质量保证的国际标准。ISO9000 的目标是促进全面质量，通过对生产和经营的每一个环节进行文件化和系统化控制，从而达到防止不合格品的目的。在 CMM 中，除第 1 级外，每个软件能力成熟度等级包含若干个对该成熟度等级至关重要的过程域，它们的实施对达到该成熟度等级的目标起到保证作用。这些过程域就称为该成熟度等级的关键过程域，反之，非关键过程域是指对达到相应软件成熟度等级的目标不起关键作用。归纳为：互相关联的若干软件实践活动和有关基础设施的一个集合。

在软件开发的各资源中，人员是最重要的资源。软件的复杂性是固有的，它引起人员通信困难、开发费用超支、开发时间超时等问题。

●在设计算法时，通常应考虑以下原则：首先说设计的算法必须是 (23) ，其次应有很好的 (24) ，还必须具有 (25) ，最后应考虑所设计的算法具有 (26) 。

(23) A. 正确的

B. 有穷性

C. 有输入

D. 用户需求

(24) A. 有穷性

B. 可读性

C. 确定性

D. 高效率与低存储量

(25) A. 有输入

B. 确定性

C. 健壮性

D. 可读性

(26) A. 可读性

B. 有输入

C. 健壮性

D. 高效率与低存储量

【解析】算法是对问题求解过程的一种描述，是为解决一个或一类问题给出的一个确定的、有限长的操作序列。在设计算法时，通常应考虑以下原则：首先说设计的算法必须是“正确的”，其次应有很好的“可读性”，还必须具有“健壮性”，最后应考虑所设计的算法具有“高效率与低存储量”。

所谓算法是正确的，除了应该满足算法说明中写明的“功能”之外，应对各组典型的带有苛刻条件的输入数据得出正确的结果。在算法是正确的前提下，算法的可读性是摆在第一位的。算法的效率是指算法的执行时间，算法的存储量是指算法执行过程中所需最大存储空间。

●在面向对象的系统中，系统责任的良好分配原则是 (27) 。软件配置项是软件配置管理的对象，即软件工程过程中产生的 (28) 。软件测试计划中的白盒测试是属于 (29) 个阶段的测试。

(27) A. 集中分配在少数控制类中

B. 在类之间均匀分配

C. 根据交互图的消息进行分配

D. 根据随意分配，并不重要

(28) A. 文档

B. 软件环境

C. 信息项

D. CASE 工具

(29) A. 系统设计

B. 编码阶段

C. 维护阶段

D. 验收阶段

【解析】在面向对象的系统中，系统责任的良好分配原则是集中分配在少数控制类中。把众多的事物归纳并划分成一些类似人类在认识客观世界时经常采用的思维方法，分类的原则是抽象，从那些与当前目标有关的本质特征中找出事物的共性，并将具有共同性质的事物划分成一类，得出一个抽象的概念。白盒测试一般属于程序员自己或者测试人员对程序内部执行路径的测试，属于编码阶段完成。

软件配置项是软件配置管理的对象，即软件工程过程中产生的信息项。

●xDSL 技术包括了 (30) 。

(30) A. SDH, VDSL, HDSL

B. ADSL, VDSL, HDSL

C. ISDN, HDSL, ADSL

D. HFC, VDSL, ADSL

【解析】xDSL 是对 HDSL(高速数字用户线)、ADSL(非对称数字用户线)、VDSL(甚高比特数字用户线)等的通称。

●整个报文(message)的端到端(end-end)传递是 (31) 层的事情。

- (31) A. 数据链路层
B. 网络层
C. 运输层
D. 表示层

【解析】运输层是利用网络层的服务和运输实体的功能，向会话层提供服务。运输层是整个协议层次结构的核心。其任务是为从源端机到目的机提供可靠的、价格合理的数据传输，而与当前网络或使用的网络无关。如果没有传输层，整个分层协议的概念也没有意义了。

运输层弥补了各个通信子网提供的服务的差异和不足，而在各通信子网提供的服务的基础上，利用本身的运输协议，增加了服务功能，使得对两端的网络用户来说，各通信子网都变成透明的，而对各子网的用户，面向通信的运输接口就成为通用的。换言之，运输层向高层用户屏蔽了下面通信子网的细节，使高层用户看不见实现通信功能的物理链路是什么，看不见数据链路采用的是什么规程，也看不见下面有几个子网以及这些子网是怎样互连起来的。运输层使高层用户看见的就是好象在两个运输层实体之间有一条端到端的可靠的通信通路。

●当采用数据报服务时，负责端到端的流量控制的是 (32) 。

- (32) A. 主机
B. 通信子网
C. 主机和通信子网
D. 交换机

【解析】当采用数据报服务时，负责端到端的流量控制的是主机。通信子网并不负责端到端的流控，它负责子网内子链路间的流控。交换机负责信息的交换，工作在链路层以下。

●在下列程序中：

```
Program test(input, output);  
var i,j:integer;  
procedure calc(p1,p2: integer);  
begin p2:=p2*p2; p1:=p1-p2; p2:=p2-p1; end{calc}  
begin{main}i:=2; j:=3;  
calc(i,j); write(j);  
end{main}
```

当参数传递采用引用方式(Call by reference)时，所得结果 j= (33) ；

当参数传递采用换名方式(Call by name)时，所得结果 j= (34) ；

当参数传递采用赋值方式(Call by value)时，所得结果 j= (35) ；

递归是程序设计中很重要的一种控制结构，通常实现递归时，采用的数据结构是 (36) 。

对那些既可以用递归方式，也可以用循环方式求解的问题，就执行效率而言 (37) 。

- (33) ,(34) A. 10
B. 16
C. 20
D. 28
(35) A. 0
B. 3
C. 5
D. 6
(36) A. 数组
B. 栈
C. 队列
D. 循环链表
(37) A. 数组

- B. 两者相同
- C. 循环优于递归
- D. 递归优于循环

【解析】一个过程的过程体若包含对其自身的调用，则称此过程是直接递归的。若一个过程的过程体调用某过程，而该过程又调用原过程或经一系列调用后又回到对原过程的调用，则称此原过程是间接递归的。通常实现递归时采用的数据结构是栈，这是因为栈有先进后出的特性，可以保存调用时的“现场”，并在调用结束时恢复“现场”，栈是实现递归的简单途径。对于既可用递归方式求解，也可用循环方式求解的问题，就执行效率和资源而言，显然是循环优于递归，因为递归的开销大。

当用户在调用点调用一个过程时，会通过参数传送信息，一个过程的形式参数用来向过程传送信息的标识符，实在参数用来在调用点向被调用过程传送信息。形式参数和实在参数之间的关系通常按位置来标定，不同程序语言所规定的参数信息传送方式不同。

当采用引用方式(Call by reference)或换名方式(Call by name)时，在过程中对形式参数的调用本质上是对实在参数单元的引用。先是给形式参数赋初值，而后，在过程中对该形式参数的赋值最终引起调用程序中实在参数值的改变。在本题中形式参数为 $p1$ 和 $p2$ 。实在参数初值为 $i=2$ 和 $j=3$ ，通过引用方式调用这两个参数，将执行以下计算过程：

$p1=2, p2=3$ $p2:=p2*p2=9$ $p1:=p-p2=2-3=-7$ $p2:=p2-p1=9-(-7)=16$

所得结果为 $j=16$ 。

参数传送采用赋值方式时，从调用点向被调用过程传送的是实在参数的值。这一值成为过程中相应位置上形式参数的初值，此后该形式参数在过程中实际是局部变量，其结果无需返回给实在参数。本题中实在参数 $j=3$ ，在过程中仅起向形式参数 $p2$ 赋初值的作用。过程中关于 $p2$ 的运算对 j 不再起作用，因而过程调用结束后 j 的值仍为 3。

●表示实体类型及实体间联系的模型称为数据模型，其中关系模型主要特征是用 (38) 表示实体集，(39) 表示实体间联系。关系运算以关系代数为理论基础，关系代数的最基本操作是并、差、笛卡尔积、(40)。完整性规则提供了一种手段来保证当授权用户对数据库作修改时不会破坏数据的一致性。关系模型的完整性规则是对关系的某种约束条件，其中参照完整性要求 (41)。假设关系 R 的属性 F 是关系 S 的外码，在对关系 R 和 S 的操作中，不可能破坏参照完整性的是 (42)。

- (38) A. 二维表
B. 三维表
C. 树节点
D. 对象

- (39) A. 主键
B. 外键
C. 关系
D. 链接

- (40) A. 自然连接和外接
B. 投影和选择
C. 交和外接
D. 自然连接和外接

- (41) A. 不允许引用不存在的元组
B. 不允许引用不存在的行
C. 不允许引用不存在的属性
D. 不允许引用不存在的列

- (42) A. 对关系 R 执行 INSERT 命令
B. 对关系 R 执行 Delete 命令
C. 对关系 S 执行 INSERT 命令

D. 对关系 S 执行 Delete 命令

【解析】关系模型的主要特征是用表格(也就是二维表)结构表达实体集，用外键表示实体间联系。关系代数有五种最基本的操作：并、差、笛卡尔积、投影和选择，从这 5 种操作中可以推出交、连接、自然连接、除法等操作。参照完整性规定外码值或者取空值，或者等于被参照关系中某个元组的主码值，也就是不允许引用不存在的元组。假设关系 R 的属性 F 是关系 S 的外码，在对关系 R 和 S 的操作中，可能破坏参照完整性的操作有：对关系 R 执行 INSERT 和 UPDATE 命令，对关系 S 执行 DELETE 和 UPDATE 命令。

●与线性表的链接存储不相符的特性是 (43) 。

(43) A. 插入和删除操作灵活

B. 需连续的存储空间

C. 存储空间动态分配

D. 需另外开辟空间来保存元素间的关系

【解析】此题是考查数据结构的链接表存储问题。需要连续的存储空间为顺序存储的特性。其他三项都是正确。

●开发软件时对提高软件开发人员工作效率至关重要 (44) 。软件工程中描述生存周期的瀑布模型一般包括计划、(45)、设计、编码、测试、维护等几个阶段，其中设计阶段在管理上又可以依次分成 (46) 和 (47) 两步。

(44) A. 程序开发环境

B. 操作系统的资源管理功能

C. 程序人员数量

D. 计算机的并行处理能力

(45) A. 需求分析

B. 需求调查

C. 可行性分析

D. 问题定义

(46) A. 方案设计

B. 代码设计

C. 概要设计

D. 数据设计

(47) A. 运行设计

B. 详细设计

C. 故障处理设计

D. 软件体系结构设计

【解析】有合适的程序开发环境可以提供有用的工具，大大提高开发人员的工作效率。软件工程中描述软件生存周期的瀑布模型一般包括计划、需求分析、设计、编码、测试和运行维护等六个阶段。需求分析完成对软件产品在功能、性能、用户接口、运行环境、可靠性、安全性、开发资源、开发进度、开发成本等方面的需求定义。问题定义、可行性分析、需求调查都可能是需求分析中要做的一部分工作。软件设计在管理上划分为概要设计与详细设计两个步骤。概要设计的目标是建立软件的体系结构，完成全局数据结构设计，同时进行处理方式设计、运行配置设计、出错处理设计、故障恢复设计等。详细设计是对每一个模块的操作的控制流程和局部数据结构进行设计。

●具有 n 个顶点 e 条边的无向图，若用邻接矩阵作为存储结构，则深度优先或广度优先搜索遍历的时间复杂度为 (48) ；若用邻接表作为存储结构，则深度优先或广度优先搜索遍历时的时间复杂度为 (49) ；深度优先或广度优先搜索遍历的空间复杂度为 (50) 。

(48) ,(50) A. $O(n^2)$

B. $O(n)$

C. $O(n-1)$

D. $O(n+1)$

- (49) A. $O(e)$
 B. $O(e-1)$
 C. $O(e^2)$
 D. $O(e+10)$

【解析】不论是深度优先还是广度优先搜索遍历，图中 n 个顶点都必须被访问一次。从某个顶点出发，要搜索到其他顶点，必须沿着图中的边去找。用邻接矩阵做图的存储结构时，这些边是分布在一个 n 阶方阵中，要检测出这些边，必须对矩阵中 n^2 个元素进行检测，因此，其时间复杂度为 $O(n^2)$ 。若用邻接表作为存储结构，只需对代表 e 条无向边的 $2e$ 个边表结点进行检测，其时间复杂度为 $O(e)$ 。深度优先搜索遍历需要用一个栈来保存本身已被访问但可能还有邻接顶点未被访问的那些顶点的序号，每个顶点都要进栈一次，故 n 个顶点需要开辟 n 个元素的栈(若用递归算法则由系统开辟)。广度优先搜索遍历需要用一个队列来保存顶点的序号，每个顶点都要进队一次，故队列长度为 n ，所以深度优先或广度优先搜索遍历的空间复杂度为 $O(n)$ 。

●设二叉树的根为第一层，则第 i 层上的结点数最多有 (51) 。

- (51) A. $2i$
 B. 2^{i+1}
 C. 2^i
 D. 2^{i-1}

【解析】此题是考查数据结构的二叉树问题。当二叉树第 i 层的点数最多时，则此二叉树在第二层往以上层都是满节点，所以第 i 层的节点数为 $2^{(i-1)}$ 。

●为下列文法选择最准确的答案：

文法 $G[S]$ 属于 (52) ：

$S \rightarrow CD \quad Ab \rightarrow bA$
 $C \rightarrow aCABa \rightarrow aB$
 $C \rightarrow bCBBb \rightarrow bB$
 $AD \rightarrow aDC \rightarrow \varepsilon$
 $BD \rightarrow bDD \rightarrow \varepsilon$
 $Aa \rightarrow bD$

$L(G) = \{ww \mid w \in \{a, b\}^*\}$

文法 $G[P]$ 属于 (53) ：

$P \rightarrow 0A \mid 1B \mid 0$
 $A \rightarrow 0A \mid 1B \mid 0P$
 $B \rightarrow 1B \mid 1 \mid 0$

文法 $G[I]$ 属于 (54) ：

$I \rightarrow IT$
 $I \rightarrow 1$
 $T \rightarrow IT$
 $T \rightarrow dT$
 $T \rightarrow 1$
 $T \rightarrow d$

其中， 1 表示 $a \sim z$ 中的任意一个英文字母， d 表示 $0 \sim 9$ 中的任意一个数字。

- (52) ~ (54) A. 1 型(上下文有关)文法
 B. 2 型(上下文无关)文法
 C. 定义标识符的 3 型(正规)文法
 D. 0 型文法

【解析】本题考查 4 种文法的定义。需要注意的是，4 个文法类的定义是逐渐增加限制的，因此每一种正规文法都是上下文无关的，每一种上下文无关文法都是上下文有关的，而每一种上下文有关文法都是 0 型文法。称 0

型文法产生的语言为 0 型语言。上下文有关文法、上下文无关文法和正规文法产生的语言分别称为上下文有关语言、上下文无关语言和正规语言。

●数字音频采样和量化过程所用的主要硬件是 (55) 。

- (55) A. 数字转换器
B. 数字编解码器
C. 数字到模拟的转换器(D / A 转换器)
D. 模拟到数字的转换器(A / D 转换器)

【解析】声音数字化的两个步骤是采样和量化。采样就是每间隔一段时间就读一次声音信号的幅度；量化就是把采样得到的声音信号幅度转换为数字值。时间上的离散叫采样，幅度上的离散称为量化。因此，数字音频采样和量化过程所用的主要硬件是模拟到数字的转换器(A / D 转换器)。

●计算机识别、存储和加工处理的对象统称为 (56) 。

- (56) A. 数据
B. 数据元素
C. 数据结构
D. 数据类型

【解析】本题考察了数据的概念。

●ISO 所制定的 OSI / RM 是 (57) 。

- (57) A. 一个具体的计算机网络
B. 一个计算机网络体系结构标准
C. 一个开放的全球标准
D. 一个标准的计算机网络

【解析】OSI / RM 是一个开放的网络体系结构标准。体系结构的英文名词 Architecture 的原意是建筑学或建筑的设计风格，它和具体的建筑物的概念是不同的。例如，可以走进一个建筑物中，但却走不进一个建筑风格中。同理，也不能把一个具体的计算机网络说成是一个抽象的网络体系结构。总之，体系结构是抽象的，而实现则是具体的，是真正在运行的计算机硬件和软件。

●MPEG 是一种 (58) ，它能够 (59) ，MPEG-4 相对于 MPEG 的以前版本的最大特点是 (60) 。

- (58) A. 静止图像的存储标准
B. 音频、视频的压缩标准
C. 动态图像的传输标准
D. 图形国家传输标准

- (59) A. 快速读写
B. 有高达 200 : 1 的压缩比
C. 无失真地传输视频信号
D. 提供大量基本模板

- (60) A. 更高的压缩比
B. 更多的基本图形的模板
C. 更强的交互能力
D. 更快的运算速度

【解析】MPEG 的目标是为视频、音频制定数据压缩标准。MPEG 的各种版本已陆续面世。MPEG 相比于其他数据压缩方案有这样一些优点：兼容性强；压缩比高；在高压比下的数据失真相对很小。MPEG 的视频压缩技术是针对动态图像的。为了提高压缩比，帧内图像的数据压缩与帧间图像的数据压缩同时进行，它采用 DCT 变换技术、预测压缩算法和插补法等，使数据的冗余大为减少，在失真很小的情况下，压缩比提高到 200。MPEG 于 1988 年开始工作，1992 年制定出 MPEG-I，主要是考虑到工业级标准，用于数字电话网络上的视频传输，也可在 Internet 上传输音频用。对动态图像的压缩编码速率可达 1.5Mbit / s 到 5Mbit / s。1994 年制定出 MPEG-II，有更高的图像质量，也有更高的速率(达到 10Mbit / s)，可播放 VCD，提供 CD 级的音质，还可提供较广的范围

改变压缩比，以适应不同画面质量，存储容量及带宽要求。MPEG-III在设计过程中，尚未推出就被抛弃了。MPEG-IV的设计是为了适应交互计算机和交互性电视技术的发展。为了具有更高的数据压缩比，更高的图像质量，更快的传输速率，突破了 MPEG-I 和 MPEG-II 中的文法，采用基于内容的压缩编码方法，使 MPEG-IV 具有更强有交互能力的特点。

● 计算机网络中对等实体间通信时必须遵循约定的标准协议。不同的协议有不同的功能，如：

SNMP: (61) ；

WAP: (62) ；

PPP (63) ；

HDLC (64) ；

OSPF (65) 。

(61) A. 简单网管协议

B. 邮件传输协议

C. 文件传输协议

D. 远程登陆

(62) A. 无线上网协议

B. 有线上网协议

C. 网络层协议

D. 路由协议

(63) A. 点到多点协议

B. 点到点协议

C. 广播协议

D. 网络层协议

(64) A. 物理层协议

B. 网络层协议

C. 链路层协议

D. 传输层协议

(65) A. 文件传输协议

B. 路由协议

C. 面向连接的协议

D. 面向非连接的协议

【解析】SNMP 为简单网络管理协议，用于网络的管理。WAP 是无线应用协议，用于手机的无线上网。PPP 是点对点协议，用于拨号上网。HDLC 是高级数据链路控制协议，OSPF 是开放最短路径优先协议，是一种路由选择协议。

● Because Web servers are platform and application (66) , they can send or request data from legacy or external applications including databases. All replies, once converted into (67) mark-up language, can then be transmitted to a (68) . Used in this way, Intranets can (69) lower desktop support costs, easy links with legacy applications and databases and, (70) all, ease of use.

(66) A. coupled

B. dependent

C. independent

D. related

(67) A. ciphertext

B. hypertext

C. paintext

D. supertext

- (68) A. browser
B. repeater
C. router
D. server

- (69) A. off
B. offer
C. office
D. officer

- (70) A. abort
B. about
C. above
D. Around

【解析】(参考译文)因为网络服务器是平台独立而且应用独立，他们能从原有或者外部应用(包括数据库)那里发送或者获取数据。全部答复一旦转换为超文本格式，就能被提交到浏览器中显示。利用这种方法，企业内部网能提供低廉的桌面支持费用，容易与原有应用系统及其数据库连接。总而言之，易于使用。

●Most computer systems are (71) to two different groups of attacks: Insider attacks and outsider attacks. A system that is known to be (72) to an outsider attack by preventing (73) from outside can still be vulnerable to the insider attacks accomplished by abusive usage of (74) users. Detecting such abusive usage as well as attacks by outsiders not only provides information on damage assessment, but also helps to prevent future attacks. These attacks are usually (75) by tools referred to as Intrusion Detection Systems.

- (71) A. vulnerable
B. week
C. sensitively
D. helpless

- (72) A. reliable
B. secure
C. indestructible
D. steady

- (73) A. visit
B. access
C. I/O
D. read/write

- (74) A. power
B. rights
C. authorized
D. common

- (75) A. searched
B. checked
C. tested
D. detected

【解析】(参考译文)大多数计算机系统主要受到两类不同类型的攻击：内部攻击和外部攻击。系统虽然可以通过阻止外部访问防止受到外部攻击，但是，仍然容易受到伴随着内部合法用户的特权滥用而产生的内部攻击。通过检测那些授权滥用的内部攻击和外部攻击，不但可以提供信息损坏的评估，而且还能免于受到以后的攻击。这些攻击通常通过来自攻击监测系统的工具来发现。

软考设计师模拟试题 2（下午题答案）

● 试题一

[问题 1]

【答案】用例的特征如下：用例总是由角色初始化；用例为角色提供值；用例具有完全性，即不管其内部是如何实现的，只有最终产生了返回角色的结果，用例的执行才能完毕。

(1)登记成绩(2)查询成绩单(3)关闭注册

[问题 2]

【答案】(4)create Schedule(5)display blank schedule()(6)get course offerings()

(7)create with offerings()(8)add schedule(Schedule)

[问题 3]

【答案】

1)用例图定义了系统的功能需求，它完全是从系统的外部观看系统功能，并不描述系统内部对功能的具体实现。在用例图中，角色代表触发系统功能的用户或其他系统，用例代表具体的功能描述。

2)类图描述系统的静态结构，表示系统中的类以及类与类之间的关系。

3)对象图描述了一组对象以及它们之间的关系，表示类的对象实例。

4)状态图表示一个状态机，强调对象行为的事件顺序。

5)时序图和协作图均表示一组对象之间的动态协作关系。其中，时序图反映对象之间发送消息的时间顺序，协作图反映收发消息的对象的结构组织。时序图和协作图是同构的，即两者之间可以相互转换。

6)活动图反映系统中从一个活动到另一个活动的流程，强调对象间的控制流程。

7)组件图描述组件以及它们之间的关系，表示系统的静态实现视图。

8)分布图反映了系统中软件和硬件的物理架构，表示系统运行时的处理节点以及节点中组件的配置。

【解析】用例描述了它所代表的功能的各个方面，即包含了用例执行期间可能发生的各种情况。用例和角色之间具有“关联”的连接关系，表示什么角色与该用例进行通信。在 UML 语言中，用例用一个椭圆图形和名称表示。

在本题中，我们通过题目说明可以识别以下用例：

1.与教师有关的用例

- 1)选择课程--选择所教的课程，并获得学生名册。
- 2)登记成绩--在学期结束时，提交学生的课程成绩。

2.与学生有关的用例

- 1)注册课程--在学期开始进行选课注册，允许在一段时间内更改或删除，课程目录系统提供当前学期的所有可选课程列表。
- 2)查看成绩单--学生可以查看以前学期的电子成绩单。

3.与注册管理员有关的用例

- 1)维护课程信息--在系统中增加、修改和删除课程信息。
- 2)维护学生信息--在系统中增加、修改和删除学生信息。
- 3)维护教师信息--在系统中增加、修改和删除教师信息。
- 4)关闭注册--删除少于3人的课程，并由付费系统通知学生缴费。

4.与安全性要求有关的用例

登录--使用此系统的人员需要进行登录，以验证其身份和权限。

发现和定义对象类应以问题域和系统责任为出发点，正确地运用抽象原则，尽可能全面地发现对象的因素，并对其进行检查和整理，最终得到系统的对象类。我们可以在用例模型的基础上，通过识别实体类、边界类和控制类，从而发现和定义系统中的对象类。识别上述对象类之后，通过建立交互图，将用例的行为分布到这些对象类中。时序图表示完成某项行为的对象类和这些对象类之间传递消息的时间顺序，其中，对象生命线是一条垂直的虚线，表示对象存在的时间；控制焦点是一个细长的矩形，表示对象执行一个所经历的时间段；消息是对象之间的一条水平箭头线，表示对象之间的通信。协作图包含一组对象和以消息交换为纽带的关联，用于描述系统的行为是如何由系统的成分合作实现的。

● 试题二

[问题 1]

【答案】(1) 1 (2) col (3) row (4) 2 (5) col (6) row (7) k

【解析】(1) 栈置空，置 EOF 为 false，并从文件中读取第一个字符到 ch，然后进入循环，循环体执行一次处理一个 ch。进入循环，利用 kind 函数算出 ch 的类型 k。(2) 算法 2-1 大致流程如下：当 k 等于什么的时候把什么入栈；当 k 等于什么的时候且栈不为空的时候出栈，如果栈为空，打印错误消息，如果都不是，则读取文件的下一个字符再次进入循环。由上看出，入栈应是类型 k 为 1，而出栈应该是类型 k 为 2。在 (4) 下面出栈时，并没有用到栈的内容。在此就有些人认为栈中的内容没什么用了，随便压个 ch，而且两个写的都是 ch。其实从逻辑上就可以理解这种解答，如果压的是同样的数据，又是在同一位置出栈，算法大可置用一个 push、pop 就可以了。所以应该继续看后面的内容，以寻找正确的答案。当看到"row<-pop() ;col<-pop();"时，所有的疑惑就迎刃而解了，应把 row 和 col 压入栈。由于是先弹出 row 后弹出 col，按栈的后进先出的规则，可知压入栈的是 col，再压 row。

[问题 2]

【答案】判断条件 1: b 判断条件 2: e 判断条件 3: f

【解析】因为后面有"pop () ;row<-pop () ;col<-pop () ;",所以空 (5) 应填 col, (6) 应填 row。又因为判断条件 1 为真时需要入栈,所以判断条件 1 应是判断字符是不是左括号,是就入栈,所以判断条件选 b。判断条件 2 和 3 是相关联的,当 2 和 3 都为真时,要进行出栈操作,因此要判断栈是否为空。由此可以得到待判断条件 2 和 3 中,有一个必定是用来判断栈是否为空的。可以用栈顶元素来确定当前括号是否和栈中压入括号是同一类型的。前提是左括号类型已经入栈了,且在栈底,如果 (7) 压入的是 k,即吻合。所以 (7) 应填 k,这样判断括号是否匹配的条件就可以确定了,如果当前 ch 是右括号且当前栈顶的左括号类型与 ch 匹配,则匹配成功。应把"栈不空"作为判断条件 2,"栈顶元素表示的是与当前字符匹配的左括号"作为判断条件 3。这样判断条件 2 填 e,判断条件 3 填 f。

● 试题三

[问题 1]

【答案】设计一中 Invoice 最高满足第一范式。

根据题意可得出以下函数依赖:

Ino→Sno, Cno, Idate

而关系 Invoice 的主码是 Ino 和 Mno。非主属性 Sno、Cno 和 Idate 并非完全依赖于主码,因此关系 Invoice 不满足第二范式,最高满足第一范式。

设计二更加合理。因为设计二解决了设计一中由于非主属性不完全依赖于主码而造成的数据冗余等问题。

[问题 2]

【答案】(1)AS(2)SUM(amount)(3)SUM(unitprice*amount)

(4)Invoice.Ino=Invoicedetail.Ino(5)Invoice.Ino, Idate 或 Invoicedetail.Ino.Idate

[问题 3]

【答案】(1)A 和 AS A(2)NOT EXISTS(3)*

[问题 4]

【答案】有必要。Merchandise 中由属性 price 表示的是商品的当前价格,超市中的价格是有可能变动的,而关系 Invoicedetail 中的属性 unitprice 表示的是在开具发票时该商品的单价。

【解析】设计一中根据题意可得出以下函数依赖:

Ino→Sno, Cno, Idate

而关系 Invoice 的主码是 Ino 和 Mno。非主属性 Sno、Cno 和 Idate 并非完全依赖于主码，因此关系 Invoice 不满足第二范式，最高满足第一范式。

关系 Invoice 的设计的不合理在于该关系中将发票的单值属性(发票号码 Ino，交易日期 Idate，顾客代码 Cno，收银员代码 Sno)和多值属性(商品代码 Mno，单价 unitprice，数量 amount)混合在一个关系中，造成关系 Invoice 的冗余异常、修改异常和删除异常。而设计二则将设计一中关系 Invoice 分解，使得发票的单值属性和多值属性分开，避免了异常。因此，设计二明显比设计一要好。

问题 2 是要建立 2005 年 1 月期间每张发票的发票号，交易日期，交易商品件数和交易总金额的视图。

首先建立视图的格式为 CREATE VIEW<视图名>AS<视图定义>，因此(1)空的答案为 AS。

本查询是从 Invoice 和 Invoicedetail 两个关系中查询，两关系的连接条件是两关系的 Ino 相等，因此(4)空的答案是 Invoice.Ino=Invoicedetail.Ino。

统计每张发票的信息需要按发票将数据分组，也就是按发票号 Ino 分组，但因为查询关系 Invoice 和 Invoicedetail 都有属性 Ino。为了避免二义性，所以分组属性是 Invoice.Ino 或者是 Invoicedetail.Ino。因为在包含聚合运算的 Select 子句中，只有在 Group By 子句中出现的属性才能在 SELECT 子句中以非聚合形式出现，而 SELECT 子句中有非聚合形式的属性 Idate 出现，所以(5)空的答案是 Invoice.Ino.Idate 或 Invoicedetail.Ino.Idate。

需要查询的是每张发票的交易商品件数和交易总金额。交易商品件数是发票商品数量的总和，因此(2)空的答案是 SUM(amount)。交易总金额是每条交易商品明细中每条记录商品金额的总和，每条记录商品金额是 unitprice*amount，因此(3)空的答案是 SUM(unitprice*amount)。

问题 3 是查询从未售出的商品信息。

SQL 语句中有两种格式为表或视图取别名："表名 AS 别名"或"表名别名"。由题中可以看出 Merchandise 的别名是 A，因此填空(1)的答案是 A 或者 AS A。

要查询"从未出售"的商品，也就是要查询的商品在交易记录中不存在，因此(2)空的答案是 NOT EXISTS。

Merchandise 中由属性 price 表示的是商品的当前价格，超市中的价格是有可能变动的，而关系 Invoicedetail 中的属性 unitprice 表示的是在开具发票时该商品的单价。如果缺少其中任意一个，将导致商品单价不能进行调整，否则，当商品的单价发生变化时，销售历史中的商品价格就随着发生变化。

● 试题四

【答案】(1)A.length(2)j++(3)O(Min(A.length, B.length))(4)m
(5)p->next(6)p(7)L->next(8)q->next(9)ha(10)O(ListLength(L))

【解析】函数 1 中，算法要求对两个顺序表进行“比较”，是一种“引用型”操作，因此在算法中不应该破坏已知表。按题目中的规定，只有在两个表的长度相等，且每个对应元素都相同时才相等；否则，两个顺序表的大小主要取决于两表中除去最大公共前缀后的第一个元素。因此，比较两表的大小不应该先比较它们的长度，而应该设一个

下标变量 j 同时控制两个表，即对两表中“位序相同”的元素进行比较。

上述算法中只有一个 `while` 循环，它的执行次数依赖于待比较的顺序表的表长，因此，算法的时间复杂度为 $O(\text{Min}(A.\text{length}, B.\text{length}))$ 。

函数 2 中，对链表来说，“插入”和“删除”仅需修改指针即可完成，并且由于前 m 个元素之间和后 n 个元素之间的链接关系都不需要改变，则算法的实际操作为：

先从链表中删除 (a_1, a_2, \dots, a_m) ，然后将 (b_1, b_2, \dots, b_n) 链接到头结点之后，再将 (a_1, a_2, \dots, a_m) 链接到 b_n 之后。

算法的时间复杂度为 $O(\text{ListLength}(L))$ 。

● 试题五

【答案】(1) `mp` 或 `mp!=NULL` (2) `mp->head` 或 `&(mp->head)`
(3) `&mp->head=NULL` (4) `*mptr=mp` (5) `ip` 或 `ip!=NULL`
(6) `&ip->next` 或 `&(ip->next)` (7) `*ipp=ip`

● 试题六

【答案】(1) `a[i][j]` (2) `t1+t2`, 其中 `t1` 可以写成 `c[i][k]`, `t2` 可以写成 `c[k][j]`
(3) `k` (4) `kay[i][j]` (5) `kay[i][j]` , `j`

【解析】(1) 此处的双层循环的作用是给数组 `c` 赋初值。即把最初的 i 号结点到 j 号结点的路径长度存入 `c`。由题目中已经有说明：“Input ()：输入有向图的顶点数、各条弧及权值，建立带权邻近矩阵 `a`。若顶点 i 到顶点 j 有弧，则 `a[i][j]` 取弧上的权值，否则 `a[i][j]` 的值取 `NoEdge`。”所以应填 `a[i][j]`。(2) 首先应该说明的是此处的三层循环所完成的功能是用递推的方式，在 i 号结点和 j 号结点中插入一个 k 号结点，然后比较 `c[i][j]` 与 `c[i][k] + c[k][j]`，如果 `c[i][k] + c[k][j]` 小于 `c[i][j]`，则用 `c[i][k] + c[k][j]` 代替 `c[i][j]`。这里用到的原则就是：`c[i][k]`，`c[k][j]` 分别是 i 到 k ， k 到 j 的最短路径，若 i 到 j 要经过 k ，则 `c[i][k] + c[k][j]` 就是 i 到 j 过结点 k 的最短路径。(3) 由于题目中提到“`kay` 为二维数组，存储最短路径，`kay[i][j] = k` 表示顶点 i 到达顶点 j 的最短路径必须经过顶点 k 。”所以，应填 `k`。(5) 此处用到了程序的递归，其实这个过程很好理解，也就是判断当中间结点为 0，表示 i, j 直接为最短路径，则直接打印即可。如果有中间结点 k ，则先打印从 i 到 k 的路径，再打印从 k 到 j 的路径。此处的中间结点存在 `kay[i][j]` 里，所以 (4) 填 `kay[i][j]`。

● 试题七

【答案】(1) `add(showBtn)` (2) `frm != null` (3) `frm.show()` (4) `frm = null` (5) `frm.hide()`

【解析】本题主要考查 Applet 窗口及 Applet 程序的执行过程，和图形用户界面程序的设计。解题关键是熟悉 Applet 窗口程序设计及其执行过程，这里要求加入一个 `Button` 构件；同时需要熟悉图形用户界面的设计，这里要求通过 Applet 中的 `Button` 构件控制一个 `Frame` 窗口的显示与隐藏，要熟悉如何建立一个新窗口，如何设置窗口的标题，如何利用 `Component` 类的方法将窗口关闭。本题中，要正确解答，需要明确程序的控制流程：首先判断用户是否按下按钮，其次判断当前窗口对象是否存在，如果窗口不存在则定义一个窗口，反之将窗口隐藏，并同时修改按钮文字作为提示，要明白提示作用是如果存在则提示关掉，如果不存在则提示显示。判断当前窗口对象是否存在的语句，存在的情况下才应该隐藏；用来隐藏窗口的语句，是 `frm.hide()`；用来显示的语句是 `frm.show()`。