

【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）

●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

●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

●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
- 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
- (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
- 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
- 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
- (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) 。在软件开发的各种资源中， (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

●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

●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

●xDSL 技术包括了 (30) 。

(30) A. SDH, VDSL, HDSL

B. ADSL, VDSL, HDSL

C. ISDN, HDSL, ADSL

D. HFC, VDSL, ADSL

●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

●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

●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{caic}

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

●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

●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

●开发软件时对提高软件开发人员工作效率至关重要是 (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

●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. $O(e+10)$

●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. $2i+1$
C. $2i$
D. $2i-1$

●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~z中的任意一个英文字母，d表示0~9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

●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)

●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

●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

●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

●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

● 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

● 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. weak

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

软考设计师模拟试题 2（下午题）

● 试题一

阅读下列说明以及图示(如图 1 所示)，回答问题 1~3。

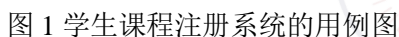
【说明】

某大学准备开发一个学生课程注册系统，学生可以使用该系统查询新学期将开设的课程和讲课教师情况，选择自己要学习的课程进行登记注册，并可以查询成绩单；教师可以使用该系统查询新学期将开设的课程和选课学生情况，并可以登记成绩单；注册管理员使用该系统进行注册管理，包括维护教师信息、学生信息和课程信息等。在每个学期的开始，学生可以获得该学期的课程目录表，课程目录表列出每门课程的所有信息，诸如基本信息、教师、开课系和选课条件等。

新学期开始前两周为选课注册时间，在此期间学生可以选课注册，并且允许改变或取消注册申请，开学两周后注册管理员负责关闭课程注册。每个学生可以选择不超过 4 门课程，同时指定 2 门侯选课程以备主选课程未选上。每门课程最多不能超过 10 人，最少不能低于 3 人，低于 3 人选课的课程将被取消。一旦学生的注册过程完毕，注册系统将有关信息提交收费系统以便学生付费。如果在实际注册过程中名额已满，系统将通知学生在提交课程表之前予以更改。

在学期结束时，学生可以存取系统查看电子成绩单。由于学生成绩属于敏感信息，系统必须提供必要的安全措施以防非法存取。

【用例图】



控制类	说明
RegisterCourseControl	负责新生选课生的选课登记
ViewReportControl	负责新生成绩的查询
SelectTeachCourseControl	负责教师课程的学生选修情况
EnrollControl	负责新生成绩的登记
ClassRegistrationControl	负责新生选课注册

```
graph TD
    Student1((:Student))
    RegisterForm[:Register Course Form]
    RegController[:Registration Controller]
    CourseCatalog[:Course Catalog]
    Student2[:Student]
    Schedule[:Schedule]

    Student1 -- "1:// (4)  
6://select 4 primary and 2  
altmate offerings" --> RegisterForm
    RegisterForm -- "4://display course offerings  
5:// (5)" --> RegController
    RegisterForm -- "2://ger course offerings()  
7://create schedule with offerings" --> RegController
    RegController -- "3:// (6)" --> CourseCatalog
    RegController -- "9:// (8)" --> Student2
    RegController -- "8:// (7)" --> Schedule
```

The diagram illustrates the interactions for a registration system. A student (**:Student**) interacts with a **:Register Course Form** and a **:Registration Controller**. The **:Register Course Form** interacts with the **:Registration Controller** and a **:Course Catalog**. The **:Registration Controller** interacts with the **:Course Catalog** and two other **:Student** and **:Schedule** objects.

Use Cases and Messages:

- 1:// (4)**: Message from **:Student** to **:Register Course Form**.
- 6://select 4 primary and 2 altmate offerings**: Message from **:Student** to **:Register Course Form**.
- 4://display course offerings**: Message from **:Register Course Form** to **:Registration Controller**.
- 5:// (5)**: Message from **:Register Course Form** to **:Registration Controller**.
- 2://ger course offerings()**: Message from **:Register Course Form** to **:Registration Controller**.
- 7://create schedule with offerings**: Message from **:Register Course Form** to **:Registration Controller**.
- 3:// (6)**: Message from **:Registration Controller** to **:Course Catalog**.
- 9:// (8)**: Message from **:Registration Controller** to **:Student**.
- 8:// (7)**: Message from **:Registration Controller** to **:Schedule**.

图 2 创建课程登记表的协作图

注释 1: 学生打算注册新的课程。
注释 2: 一张这学期可选择的课程列表。
注释 3: 显示一张为学生选课用的空白登记表。

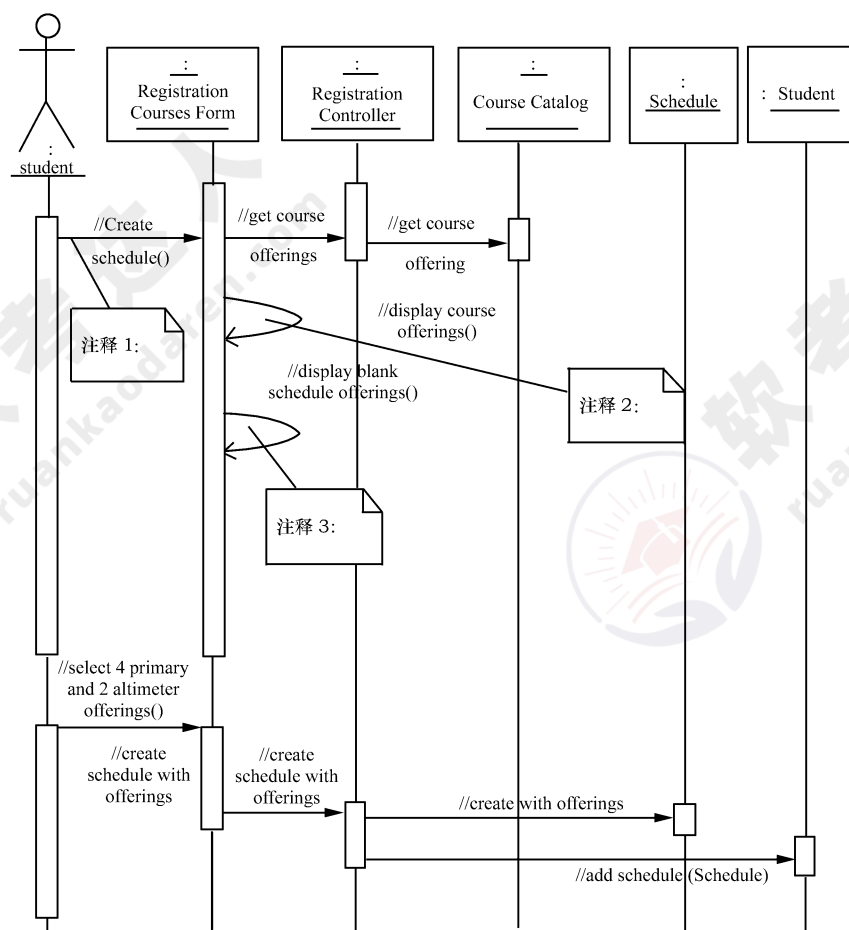


图 3 创建课程登记表的时序图

【问题 1】

在 UML 中，用例代表一个完整的功能，如与角色通信、进行计算或在系统内工作等。请简要说明用例具有哪些的特征，并指出用例图中 (1) ~ (3) 处表示的内容。

【问题 2】

协作图与时序图是同构的，二者表示的都是同样的系统交互活动，只是各自的侧重点不同而已。根据题目提供的信息，指出协作图中 (4) ~ (8) 处表示的内容。

【问题 3】

UML 采用 5 个互联的视图来描述软件系统的体系结构，即用例视图(Use-case View)、设计视图(Design View)、进程视图(Process View)、实现视图(Implementation View)和展开视图(Deployment View)。系统模型中每一个视图的内容是由一些图来描述的，UML 中包含用例图、类图、对象图、状态图、时序图、协作图、活动图、组件图、分布图等 9 种图。对整个系统而言，其功能由用例图描述，静态结构由类图和对对象图描述，动态行为由状态图、时序图、协作图和活动图描述，而物理架构则是由组件图和分布图描述。请分别指出用例图、类图、对象图、状态图、时序图、协作图、活动图、组件图、分布图的作用。

● 试题二

阅读下列说明和算法，回答问题 1 和问题 2，将解答填入答题纸的对应栏内。

【说明】

算法 2-1 是用来检查文本文件中的圆括号是否匹配。若文件中存在圆括号没有对应的左括号或者右括号，则给出相应的提示信息，如下所示：

文件提示信息

(1+2)

abc)缺少对应左括号：第 2 行，第 4 列

((def)gx)缺少对应左括号：第 3 行，第 10 列

((h)

ij)(k

(1ml)缺少对应右括号：第 5 行，第 4 列;第 4 行，第 1 列

在算法 2-1 中，stack 为一整数栈。算法中各函数的说明见表 4。

表 4	
函数名	函数功能
push(m)	将整数 m 压入栈 stack 中
pop()	stack 的栈顶元素出栈
empty()	判断 stack 栈是否为空，若为空，函数返回 1，否则函数返回 0
nextch()	读取文本文件中的下一个字符，并返回该字符的 ASCII 码值，将字符的 ASCII 码值存放在变量 ch 中，并将文件指针移动到下一个字符，若遇到文件结束符，则返回 EOF
kind(char ch)	判断字符 ch 是左括号还是右括号，若是左括号，函数返回 1，若是右括号，函数返回 2，否则函数返回 0

【算法 2-1】

将栈 stack 置空，置 EOF 为 false

ch<-nextch();

while(not EOF)

k<-kind(CH);

if(k== (1))

push((2));push((3));

elseif(k== (4))

if(not empty())

pop();pop();

else

显示错误信息(缺少对应左括号或右括号);

显示行号 row;显示列号 col;

endif

endif

ch<-nextch();

endwhile

if(not empty())

显示错误信息(缺少对应左括号或右括号);

while(not empty())

row<-pop();col<-pop();

显示行号 row;显示列号 col;

endwhile

endif

为了识别更多种类的括号，对算法 2-1 加以改进后得到算法 2-2。算法 2-2 能够识别圆括号、方括号和花括号(不同类型的括号不能互相匹配)。改进后，函数 kind(char ch)的参数及其对应的返回值见表 5。

表 5							
ch	{	}	[]	()	其他
返回值	1	2	3	4	5	6	0

【算法 2-2】

将栈 stack 置空，置 EOF 为 false

ch<-nextch();

while(not EOF)

k<-kind(ch);

if(k>0)

if(判断条件 1)


```

push(__ (5) __);push(__ (6) __);push(__ (7) __);
elseif(判断条件 2and 判断条件 3)
pop();pop();pop();
else
显示错误信息(缺少对应左括号或右括号);
显示行号 row;显示列号 col;
endif
endif
ch<-nextch();
endwhile
if(not empty())
显示错误信息(缺少对应左括号或右括号);
while(not empty())

pop();row    pop();col    pop();

显示行号 row;显示列号 col;
endwhile
endif

```

【问题 1】

请将【算法 2-1】和【算法 2-2】中__(1)~__(7)处补充完整。

【问题 2】

请从下面的选项中选择相应的判断逻辑填补【算法 2-2】中的“判断条件 1”至“判断条件 3”。注意，若“判断条件 2”的逻辑判断结果为假，就无需对“判断条件 3”进行判断。

- (a)字符是括号
- (b)字符是左括号
- (c)字符是右括号
- (d)栈空
- (e)栈不空
- (f)栈顶元素表示的是与当前字符匹配的左括号
- (g)栈顶元素表示的是与当前字符匹配的右括号

● 试题三

阅读下列说明，回答问题 1~问题 3，将解答填入答题纸的对应栏内。

【说明】

某超市的销售业务由一个销售业务管理系统进行管理，该系统每完成一次交易都需要提供顾客发票，其格式见表 6 所示。

表 6 超市购物发票

顾客信息:		收银员信息:	
顾客代码:	姓名:	收银员代码:	姓名:
商品名称	单价	数量	金额
总计:		件	元

对于这样一个销售业务管理系统，分别给出了以下两种关系数据库的设计(下划线表示主关键字)

设计一:

顾客 Customer(顾客代码 Cno, 姓名 name, 住址 address, 联系电话 phone)

收银员 Salesman(收银员代码 Sno, 身份证号 idno, 姓名 name, 住址 address, 联系电话 phone)

商品 Merchandise(商品代码 Mno, 商品名称 Mname, 价格 price)

发票 Invoice(发票号码 Ino, 交易日期 Idatetime, 顾客代码 Cno, 收银员代码 Sno, 商品代码 Mno, 单价 unitprice, 数量 amount)

设计二:

顾客 Customer(顾客代码 Cno, 姓名 name, 住址 address, 联系电话 phone)

收银员 Salesman(收银员代码 Sno, 身份证号 idno, 姓名 name, 住址 address, 联系电话 phone)

商品 Merchandise(商品代码 Mno, 商品名称 Mname, 价格 price)

发票 Invoice(发票号码 Ino, 交易日期 Idatetime, 顾客代码 Cno, 收银员代码 Sno)

发票明细 Invoicedetail(发票号码 Ino, 商品代码 Mno, 单价 unitprice, 数量 amount)

【问题 1】

设计一的关系模式 Invoice 最高满足第几范式?为什么?设计一和设计二哪个更加合理?为什么?

【问题 2】

根据设计二中关系模式, 以下 SQL 语句是用于“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”的不完整语句, 请填补其中的空缺。

CREATE VIEW Invoice -total (1)

SELECT Invoice.Ino, Idatetime, (2), (3)

FROM Invoice, Invoicedetail

WHERE (4) AND

Idatetime BETWEEN '2005-01-01' AND '2005-01-31'

GROUP BY (5);

【问题 3】

根据设计二中关系模式, 以下 SQL 语句是用于“查询从未售出的商品信息”的不完整语句, 请填补其中的空缺。

SELECT Mno, Mname, price

FROM Merchandise (1)

WHERE (2)

(SELECT (3)

FROM Invoicedetail

WHERE A. Mno=Invoicedetail.Mno);

【问题 4】

设计二中关系 Merchandise 中由属性 price 表示商品价格, 关系 Invoicedetail 中的属性 unitprice 也表示商品价格。两个是否有必要同时存在?为什么?

● 试题四

阅读下列函数说明, 将应填入(n)处的字句写在答卷纸的对应栏内。

【函数 1 说明】

函数 compare(SqlList A, SqlList B)的功能是: 设 $A=(a_1, \dots, a_m)$ 和 $B=(b_1, \dots, b_n)$ 均为顺序表, “比较”, 两个顺序表 A 和 B 的大小。设 A' 和 B' 分别为 A 和 B 中除去最大共同前缀后的子表(例如, $A=(y, x, x, z, x, z)$, $B=(y, x, x, z, y, x, x, z)$, 则两者中最大的共同前缀为 (y, x, x, z) , 在两表中除去最大共同前缀后的子表分别为 $A'=(x, z)$ 和 $B'=(y, x, x, z)$)。若 $A'=B'=\text{空表}$, 则 $A=B$; 若 $A'=\text{空表}$, 而 $B' \neq \text{空表}$, 或者两者均不为空表, 且 A' 的首元小于 B' 的首元, 则 $A < B$; 否则 $A > B$ 。

提示: 算法的基本思想为: 若相等, 则 $j+1$, 之后继续比较后继元素; 否则即可得出比较结果。显然, j 的初值应为 0, 循环的条件是 j 不超出其中任何一个表的范围。若在循环内不能得出比较结果, 则循环结束时有 3 种可能出现的情况需要区分。

【函数 1】

int compare(SqlListA, SqlList B)

```

{
//若 A<B, 则返回-1; 若 A=B, 则返回 0;若 A>B, 则返回 1
j=0;
while(i<__ (1) __ && j<
    B. length)
if(A. elem[j]<
    B. elem[j])return(-1);
else if(A. elem[j]>
    B. elem[j])return__ (1) __;
else__ (2) __;
if(A. length==
    B. length)return(0);
else if(A. length<
    B. length)return(-1);
else return__ (1) __;
} //compare
//函数 1 的时间复杂度是__ (3) __。

```

【函数 2 说明】

函数 exchange_L(SLink&L, int m)的功能是: 用尽可能少的辅助空间将单链表中前 m 个结点和后 n 个结点的互换。即将单链表(a1, a2..., am, b1, b2, ..., bn)改变成(b1, b2, ..., bn, a1, a2, ..., am)。

【函数 2】

```

void exchange_L(SLink &L, int m)
{
if(__ (4) __&&L->next)//链表不空且 m!=0
{
P=L->next; k=1;
while(k<m&& p)//查找 am 所在结点
{
P=__ (5) __; ++k;
}
if(__ (6) __&&p->next)//n!=0 时才需要修改指针
{
ha=L->next; //以指针 ha 记 a1 结点的位置
L->next=p->next; //将 b1 结点链接在头结点之后
p->next=NULL; //设 am 的后继为空
q=__ (7) __; //令 q 指向 b1 结点
while(q->next)q=__ (8) __; //查找 bn 结点
q->next=__ (9) __; //将 a1 结点链接到 bn 结点之后
}
}
}
//函数 2 的时间复杂度是__ (10) __。

```

● 试题五

阅读下列程序说明和 C 程序, 将应填入程序中(n)处的字句, 写在答卷纸的对应栏内。

【程序说明】

本程序先从文件读入各考生的准考证号(设为整型数)及成绩，并将其存放在一棵检索二叉树上，二叉树结点的键值是成绩，每个结点带一链表，链表结点存放取得该成绩的考生的准考证号。然后，程序按中序遍历检索二叉树，从高分到低分输出结果，使每行输出成绩及其取得成绩的考生的准考证号。

【程序】

```
#include <stdio.h>
typedef struct idnode {
    int id;
    struct idnode * next;
} IdNode;
typedef struct marknode {
    int mark;
    IdNode *head;
    struct marknode *left, *right;
} MarkNode;
char fname [  ] ="sp07.dat";
main()
{ int id, mark;
  MarkNode *root=null;
  FILE *fp=fopen(fname,"r");
  if(!fp) {
    printf("file%s open error. \n", fname);
    exit(0);
  }
  while (!feof(fp)) {
    fscanf(fp,"%d%d", &id, &mark);
    btree(&root, id, mark);
  }
  fclose(fp);
  print(root);
}
btree(MarkNode**mpptr, int id, int mark)
{ IdNode *ip;

  MarkNode *mp=*mpptr;
  if __ (1) __ {
    if (mark==p->mark) addIdNODE (__ (2) __, id);
    else if (mark>mp->mark) btree (&mp->left, id, mark);
    else btree(&mp->right, id, mark);
  }else
  { mp=(marknode *) malloc(sizeof (marknode));
    mp->mark=mark;
    mp->left=mp->right=NULL;
    __ (3) __
    addIdNode(&mp->head, id);
```

```

__ (4) __;
}
}
addIdNode(IdNode **ipp, int id)
{ IdNode *ip=*ipp;
if ( __ (5) __)addIdNode ( __ (6) __), id;
else{
ip=(IdNode *)malloc(sizeof(IdNode));
sp->id=id;
ip->next=NULL;
__ (7) __
}
}
print(MarkNode *mp)
{ IdNode *ip, *ip0;
if (mp){
print (mp->left);
printf("%6d: \t",mp->mark);
ip=mp->head;
while(ip){
printf("%6d",ip->id);
ip0=ip;
ip=ip->next;
free(ip0);
}
printf(" \n");printf(mp->right);free(mp);
}
}

```

● 试题六

阅读以下说明和 C++ 代码，将应填入(n)处的字句写在答题纸的对应栏内。

【说明】

本题将有向网(带权有向图)定义为类 AdjacencyWDigraph。类中的数据成员 n 表示有向网中的顶点数;a 为带权邻接矩阵, 用于存储有向网中每一对顶点间弧上的权值;c 为二维数组, 存储有向网中每一对顶点间的最短路径长度;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。类中的主要成员函数有:

Input():输入有向网的顶点数、各条弧及权值, 建立带权邻接矩阵 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。

AllPairs():用弗洛伊德(Floyd)算法求有向网中每一对顶点间的最短路径长度。

OutShortestPath(int i,int j):计算顶点 i 到顶点 j 的最短路径。

outputPath(int i,int j):输出顶点 i 到顶点 j 的最短路径上的顶点。

Floyd 算法的基本思想是递推地产生一个矩阵序列 C0,C1, C2, ..., Cn,其中 C0 是已知的带权邻接矩阵, a,Ck(i,j)(0 ≤ i,j < n)表示从顶点 i 到顶点 j 的中间顶点序号不大于 k 的最短路径长度。如果 i 到 j 的路径没有中间顶点, 则对于 0 ≤ k < n,有 Ck(i,j)=C0(i,j)=a [i] [j]。递推地产生 C1, C2, ..., Cn 的过程就是逐步将可能是最短路径上的顶点作为路径上的中间顶点进行试探, 直到为全部路径都找遍了所有可能成为最短路径上的中间顶点, 所有的最短

路径也就全部求出，算法就此结束。

【C++代码】

```
#include<iostream. h>
#define NoEdge 10000 //当两个顶点之间没有边相连时，在邻接矩阵中用 NoEdge 表示
void Make2DArray(int ** &x,int rows,int cols);
class AdjacencyWDigraph{
private
int n;//有向网中的顶点数目
int**a;//存储顶点间弧上的权值
int**c;//存储计算出的最短路径长度
int**kay;//存储求出的最短路径
public:
int Vertices()const {return n;}
void AllPairs();
void Input();//输入有向网的顶点数、各条弧及权值，建立邻接矩阵 a
void OutShortestPath(int i,int j);//计算顶点 i 到 j 的最短路径(试卷中未列出)
~AdjacencyWDigraph();//析构函数(试卷中未列出)
private:
void outputPath(int i,int j);
};
void AdjacencyWDigraph::AllPairs()
{ int i,j,k,t1,t2,t3;
for(i=1;i<=n;k++)
for(j=1;j<=n;j++)
{ c [i] [j] = __ (1) __; kay [i] [j] =0;}
for(k=1;k<=n;k++)
for(i=1;i<=n;i++){
if(i==k) continue;
t1=c [i] [k] ;
for(j=1;j<=n;j++){
if(j==k||j==i)continue;
t2=c [k] [j] ;t3=c [i] [j] ;
if(t1!=NoEdge && t2!=NoEdge &&(t3==NoEdge||t1+t2<t3))
{ c [i] [j] = __ (2) __; kay [i] [j] = __ (3) __;}
} //for
} //for
}
void AdjacencyWDigraph::outputPath(int i,int j)
{//输出顶点 i 到 j 的最短路径上的顶点
if(i==j)return;
if(kay [i] [j] ==0)cout<<j<<' ' ;
else { outputPath(i,__ (4) __); outputPath(__ (5) __);}
}
void AdjacencyWDigraph::Input()
{ int i,j,u,v,w,E;
```

```

cout<<" 输入网中顶点个数: " ;cin>>n;
cout<<" 输入网中弧的个数:" ;cin>>E;
Make2DArray(a,n+1,n+1);
for(i=1;i<=n;i++)
for(j=1;j<=n;j++)a [i] [j] =NoEdge;
for(i=1;i<=n;i++)a [i] [i] =0;
Make2DArray(c,n+1,n+1);
Make2DArray(kay,n+1,n+1);
for(i=1;i<=E;i++){
cout<<" 输入弧的信息(起点终点权值): " ; cin>>u>>v>>w;a [u] [v] =w;
}
}

void Make2DArray(int**&x,int rows,int cols)
{int i,j;
x=new int* [rows+1] ;
for(i=0;i<rows+1;i++)x [i] =new int [cols+1] ;
for(i=1;i<=rows;i++)
for(j=1;j<=cols;j++)x [i] [j] =0;
}

```

● 试题七

【说明】

下面是一个 Applet 程序，其功能是通过一个按钮控制一个窗口的创建，显示与隐藏，并且以按钮文字作为提示，可以随着窗口的状态改变，即如果窗口出现，则按钮文字为"Hide myFrm"，提示用户点击按钮，则隐藏窗口，反之亦然。请将横线处语句补充完整。

程序运行结果如图 5 所示：



图 5

```

import java. awt.*;
import java. applet.*;
/*
<applet code="ex8_7.class" width=800 height=400 >
</applet>
*/
public class ex8_7 extends Applet{
private Frame frm;

```

```
private Button showBtn;  
public void init(){  
    showBtn = new Button( "Show Frame" );  
    __ (1) __;  
}  
public boolean action( Event e, Object o ){  
    if( e.target == showBtn ){  
        if( __ (2) __ ){  
            __ (3) __;  
            frm.dispose();  
            __ (4) __;  
            showBtn.setLabel("Show myFrm");  
        }  
        else{  
            frm = new Frame( "myFrm" );  
            frm.resize( 200, 150 );  
            frm.setBackground( Color.gray );  
            __ (5) __;  
            showBtn.setLabel("Hide myFrm");  
        }  
    }  
    return true;  
}  
}
```

ex8_7.html

```
<HTML>  
<HEAD>  
<TITLE>ex8_7</TITLE>  
</HEAD>  
<BODY>  
<applet  code="ex8_7.class" width=800 height=400 >  
</applet>  
</BODY>  
</HTML>
```