软考设计师模拟试题 3（上午题）

●一个 3.5 英寸的磁盘,最小磁道的直径为 4 厘米,最大磁道直径为 8 厘米,每分钟 10000 转,共有 30 记录面,每个记录面有 8000 个磁道,每条磁道上有 511 个扇区,每个扇区实际记录有 600 个字节,其中有效数据为 512 个字节.则这个磁盘存储器的有效存储容量是 (1) GB,磁道密度是每毫米 (2) 跳磁道。

(1) A．60
B．58
C．63
D．30

(2) A．350
B．400
C．800
D．200

●

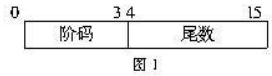


图 1

如图 1 所示为计算机中 16 位浮点数的表示格式。
某机器码为 1110001010000000。
若阶码为移码且尾数为反码，其十进制真值为 (3) ；
若阶码为移码且尾数为原码，其十进制真值为 (4) ；
若阶码为补码且尾数为反码，其十进制真值为 (5) ；
若阶码为补码且尾数为原码，其十进制真值为 (6) ，将其规格化后的机器码为 (7) 。

(3) ～ (6) A．0.078125
B．20
C．1.25
D．20.969375

(7) A．1110001010000000
B．11110101000000
C．1101010100000000
D．11110001010000

●UML 称为统一的建模语言，它把 Booch、Rumbaugh 和 Jacobson 等各自独立的 OOA 和 OOD 方法中最优秀的特色组合成一个统一的方法。UML 允许软件工程师使用由一组语法的语义的实用规则所支配的符号来表示分析模型。
在 UML 中用 5 种不同的视图来表示一个系统，这些视图从不同的侧面描述系统。每一个视图由一组图形来定义。这些视图概述如下：

(8) 用使用实例(use case)来建立模型，并用它来描述来自终端用户方面的可用的场景。
(9) 对静态结构(类、对象和关系)模型化。
(10) 描述了在用户模型视图和结构模型视图中所描述的各种结构元素之间的交互和协作。
(11) 将系统的结构和行为表达成为易于转换为实现的方式。
(12) 表示系统实现环境的结构和行为。

(8) ～(10) A．环境模型视图
B．行为模型视图
C．用户模型视图
D．结构模型视图

(11)，(12) A．环境模型视图

B．实现模型视图
C．结构模型视图
D．行为模型视图

●商品条码是在流通领域中用于标识商品的 (13) 通用的条码。条码中的 (14) 供人们直接识读，或通过键盘向计算机输入数据。

(13) A．行业
B．国际
C．国内
D．企业

(14) A．商品代码
B．条码符号
C．条码代码
D．商品条码

●在CORBA 体系结构中，负责屏蔽底层网络通信细节的协议是 (15) 。

(15) A．IDL
B． RPC
C． ORB
D．GIOP

●电子商务具有 (16) 的运作模式。

(16) A．B2C
B．C2B
C．C2C
D．A2B

●人们对软件存在着许多错误的观点，这些观点表面上看起来很有道理，符合人们的直觉，但实际上给管理者和开发人员带来了严重的问题。下述关于软件开发的观点中正确的是 (17) 。

(17) A．我们拥有一套讲述如何开发软件的书籍，书中充满了标准与示例，可以帮助我们解决软件开发中遇到的任何问题
B. 如果我们已经落后于计划，可以增加更多的程序员和使用更多的 CASE 工具来赶上进度
C. 项目需求总是在不断变化，我们可以采用瀑布模型来解决此类问题
D．需要得多是软件项目失败的主要原因

●下面是关于树和线性结构的描述：

线性结构存在惟一的没有前驱的 (18) ，树存在惟一的没有前驱的 (19) ：线性结构存在惟一的没有后继的 (20) ，树存在多个没有后继的 (21) ；线性结构其余元素均存在 (22) ，树其余结点均存在惟一的前驱(双亲)结点和多个后继(孩子)结点。

由此可见，由于线性结构是一个顺序结构，元素之间存在的是一对一的关系，而树是一个层次结构，元素之间存在的是一对多的关系。

(18) ～(21) A．根结点
B．首元素
C．尾元素
D．叶子

(22) A．惟一的前驱元素和后继元素
B．惟一的前驱(双亲)结点和多个后继(孩子)结点
C．叶子
D．一对一

●下面是关于树和线性结构的描述：

线性结构存在惟一的没有前驱的首元素，树存在惟一的没有前驱的根结点：线性结构存在惟一的没有后继的尾元

素，树存在多个没有后继的叶子；线性结构其余元素均存在惟一的前驱元素和后继元素，树其余结点均存在(23)　。

由此可见，由于线性结构是一个　(24)　结构，元素之间存在的是　(25)　的关系，而树是一个　(26)　结构，元素之间存在的是　(27)　的关系。

(23) A．惟一的前驱元素和后继元素

B．惟一的前驱(双亲)结点和多个后继(孩子)结点

C．叶子

D．一对一

(24) ~(27) A．一对一

B．一对多

C．顺序

D．层次

●软件开发模型用于指导软件开发。演化模型是在快速开发一个　(28)　的基础上，逐步演化成最终的软件。螺旋模型综合了　(29)　的优点，并增加了　(30)　。

喷泉模型描述的是面向　(31)　的开发过程，反映了该开发过程的　(32)　特征。

(28) A．模块

B．运行平台

C．　原　型

D．主程序

(29) A．瀑布模型和演化模型

B．瀑布模型和喷泉模型

C．演化模型和喷泉模型

D．原型和喷泉模型

(30) A．质量评价

B．进度控制

C．版本控制

D．风险分析

(31) A．数据流

B．数据结构

C．对象

D．构件(Component)

(32) A．迭代和有间隙

B．迭代和无间隙

C．无迭代和有间隙

D．无迭代和无间隙

●ISO 为运输层定义了 4 种类型的服务原语，由运输层服务用户产生的原语是　(33)　。

(33) A．请求原语指示原语

B．请求原语响应原语

C．指示原语确认原语

D．相应原语确认原语

●IEEE 802 规范主要与 OSI 模型的　(34)　有关。

(34)A.较低的 4 层　B.转输层和网络层

●因为 ATM　(35)　，即信元沿同一条路径走，所以，信元一般不会失序。

(35) A．是异步的

B．采用了分组交换的技术

C．采用电路交换的技术

D．用虚电路

●当存储器采用段页式管理时，主存被划分为定长的 (36) ，程序按逻辑模块分成 (37) 。在某机器的多道程序环境下，每道程序还需要一个 (38) 作为有用户标志号，每道程序都有对应 (39) 。一个逻辑地址包括 (38) ，x、段号 s、页号 p 和页内地址 d 等 4 个部分。

设逻辑地址长度分配如下，其中 x、s、p、d 均以二进制数表示。

212019141311100

xSpd

其转换后的地址为 (40) 。

(36) A．段

B．页

C．区域

D．块

(37) A．区域

B．页

C．块

D．段

(38) A．模块号

B．区域号

C．基号

D．区域

(39) A．一个段表和一个页表

B．一个段表和一组页表

C．一组段表和一个页表

D．一组段表和一组页表

(40) A．x*220+s*214+p*211+d

B．(((x)+s)+p+d

C．(((x)+s)+p)*211+(d)

D．(((x)+s)+p*211+d

●程序设计语言包括 (41) 等几个方面，它的基本成分包括 (42) 。Chomsky(乔姆斯基)提出了形式语言的分层理论，他定义了四类文法：短语结构文法、上下文有关文法、上下文无关文法和正则文法。一个文法可以用一个四元组 G=(Σ,V,S,P)表示，其中，Σ是终结符的有限字符表，V 是非终结符的有限字母表，S(∈V)是开始符号，P 是生成式的有限非空集。在短语文法中，P 中的生成式都是 α→β 的形式，其中 a∈ (43) ，β∈(Σ∪V)*。在上下文有关文法中，P 中的生成式都是 α1Aα2→α1βα2 的形式，其中 A∈ (44) ，β∈(Σ∪V*),β≠ε。在上下文无关文法中，P 中的生成式的左部∈ (45) 。

(41) A．语法、语义

B．语法、语用

C．语义、语用

D．语法、语义、语用

(42) A．数据、传输、运算

B．数据、运算、控制

C．数据、运算、控制、传输

D．顺序、分支、循环

(43)A．V+

B．(Σ∪V) C．(Σ∪

V)* D．(Σ∪V)*V(Σ

∪V)*

(44) A．V

B. V+

C. ∑∪V

D．(∑∪V)*

(45) A．V

B. V+

C. ∑∪V

D．(∑∪V)*

●设有关系模式 S (Sno，Sname，Pno，Pname，Q，A)表示销售员销售商品情况，其中各属性的含义是：Sno 为销售员员工号，Sname 为销售员姓名，Pno 为商品号，Pname 为商品名称，Q 为销售商品数目，A 为销售商品总金额。根据定义有如下函数依赖集：P={Sno→Sname，Sno→Q，Sno→A，Pno→Pname}。

关系模式S 的关键字是 (46) ，W 的规范化程度最高达到 (47) 。若将关系模式S 分解为3 个关系模式S1(Sno，Sname，Q，A)，S2(Sno，Pno，Pname)，则 S1 的规范化程度最高达到 (48) ，S2 的规范化程度最高达到 (49) 。SQL 中集合成员资格的比较操作"元组 IN(集合)"中的"IN"与 (50) 操作符等价。

(46) A．(Sno，Q)

B．(Pno，A)

C．(Sno，Pno)

D．(Sno，Pno，Q)

(47) A．1NF

B．2NF

C．3NF

D．BCNF

(48) A．1NF

B．2NF

C．3NF

D．BCNF

(49) A．1NF

B．2NF

C．3NF

D．BCNF

(50) A．<>ANY

B．=ANY

C．<>Like

D．=Like

●为了保证数据库的完整性(正确性)，数据库系统必须维护事务的以下特性 (51) 。

(51) A．原子性、一致性、隔离性、持久性

B．原子性、一致性、隔离性、闭包性

C．一致性、隔离性、持久性、完整性

D．隔离性、闭包性、时间性、适用性

●在平衡二叉排序树上进行查找时，其时间复杂度为 (52) 。

(52) A．O(log2n+1)

B．O(log2n)

C．O(log2n-1)

D．log22n

●各种需求方法都有它们共同适用的 (53) 。

(53) A．说明方法

B．描述方式

C． 准 则

D．基本原则

●对于单链表，如果仅仅知道一个指向链表中某结点的指针 p， (54) 将 p 所指结点的数据元素与其确实存在的直接前驱交换，对于单循环链表来说 (55) ，而对双向链表来说 (56) 。

(54) ~(56) A．可以

B．不可以

C．不确定

D．仅能一次

●采用邻接表存储的图的深度优先遍历算法类似于二叉树的 (57) 。

(57) A．中序遍历

B．前序遍历

C．后序遍历

D．按层遍历

●采用邻接表存储的图的广度优先遍历算法类似于二叉树的 (58) 。

(58) A．中序遍历

B．前序遍历

C．后序遍历

D．按层遍历

●用顺序存储的方法将完全二叉树中的所有结点逐层存放在一维数组R[1]到 R[n]中，那么，结点 R[i]若有左子树，则左子树是结点〖ZZ(Z〗 (59) 。

(59) A．R[2i+1]

B．R[2i-1]

C．R[i／2]

D．R[2f]

●假定一棵三叉树的结点数为 50，则它的最小高度为 (60) 。

(60) A．3

B．4

C．5

D．6

●任何一棵二叉树的叶结点在前序、中序、后序序列中的相对次序 (61) 。

(61) A．不发生改变

B．发生改变

C．不能确定

D．以上都不对

●多媒体电子出版物创作的主要过程可分为 (62) 。基于内容检索的体系结构可分为两个子系统： (63) 。

(62) A．应用目标分析、脚本编写、各种媒体数据准备、设计框架、制作合成、测试

B．应用目标分析、设计框架、脚本编写、各种媒体数据准备、制作合成、测试

C．应用目标分析、脚本编写、设计框架、各种媒体数据准备、制作合成、测试

D．应用目标分析、各种媒体数据准备、脚本编写、设计框架、制作合成、测试

(63) A．用户访问和数据库管理子系统

B．多媒体数据管理和调度子系统

C．特征抽取和查询子系统

D．多媒体数据查询和用户访问子系统

●MIDI 是一种数字音乐的国际标准，MIDI 文件存储的 (64) 。它的重要特色是 (65) 。

(64) A．不是乐谱而是波形

B．不是波形而是指令序列
C．不是指令序列而是波形
D．不是指令序列而是乐谱

(65) A．占用的存储空间少
B．乐曲的失真度少
C．读写速度快
D．修改方便

● (66) is a protocol that a host uses to inform a router when it joins or leaves anInternet multicast group.
(67) is an error detection code that most data conlmunication networks use.
(68) is an interior gateway protocol that uses a distance vector algorithm to propagaterouting information.
(69) is a transfer mode in which all types of information are organized into fixed form cells on all asynchronous or nonperiodic basis over a range of mediA．
(70) is an identifier of a web page.

(66) A．ICMP
B．SMTP
C．IGMP
D．ARP

(67) A．4B／5B
B．CRC
C．Manchester Code
D．Huffman Code

(68) A．OSPF
B．RIP
C．RARP
D．BGP

(69) A．ISDN
B．x.25
C．Frame Relay
D．ATM

(70) A．HTTP
B．URL
C．HTML
D．TAG

● Network managers have long awaited practical voice-over IP(VOIP)solutions.VOIP promises(71)network management and decreases costs by(72)a company's telephony and data infrastructures into one network.And a VOIP solution implemented at a company's head-quarters with far-reaching branch offices can(73)tremendous amounts of(74)in long distance phone bills，provided that solution delivers POTS-like voice(75)over the Internet.

(71)A.complicated B.useful C.ease D.orderly
(72)A.converging B.dividing C.combine D.bringing
(73)A.get B.put C.save D.waste
(74)A.cash B.money C.space D.time
(75)A.qualtity B.quality C.volune D.speed

软考设计师模拟试题 3（下午题）

●试题一

阅读下列说明和数据流图，回答问题 1～问题 3。

【说明】

某医院收费系统的主要功能是收取病人门诊的各项费用。系统的收费功能分为 3 个方面：病历收费、挂号收费和根据处方单内容收取检查或药物费用。

1.病人初次来该医院看病，首先购买病历，记录病人基本情况。

2.病人看病前要挂号。根据病人的病历和门诊部门(内科、外科等)，系统提供相应的挂号单和处方单，并收取费用。

3.病人根据处方单进行进一步检查或取药前需交纳各项费用。系统首先根据病人基本情况检查处方单中病历号是否正确，记录合格的处方单，并提供收据。

4.所有收费都必须依据定价表中的定价来计算，且所有收费都必须写入收费记录中。

医院收费系统的顶层图如图 2 所示；医院收费系统的第 O 层 DFD 图如图 3 所示。其中，加工 1 的细化图如图 4 所示，加工 2 的细化图如图 5 所示。

假定顶层图是正确的，″定价表″文件已由其他系统生成。

【数据流图】

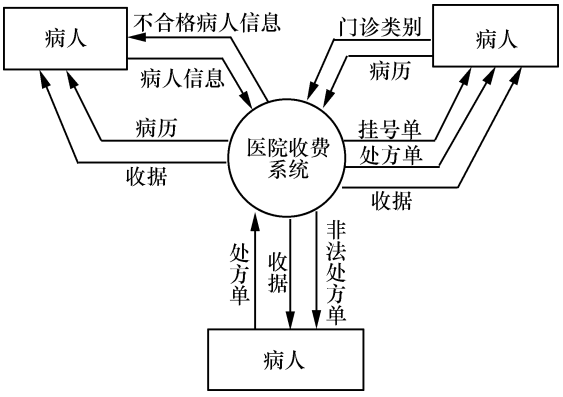


图 2 医院收费系统的顶层图

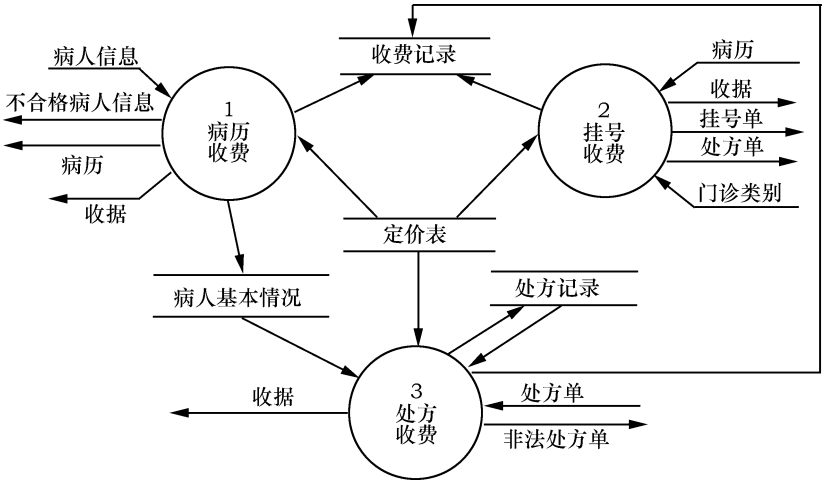
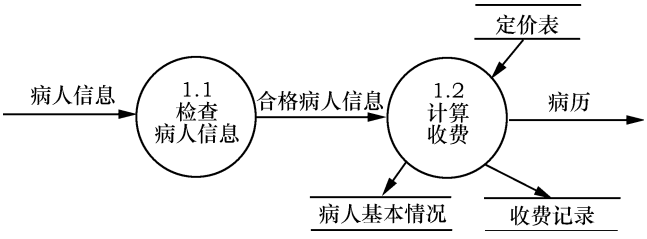


图 3 医院收费系统的 0 层图



软考学习交流群：782973214

图 4 医院收费系统的加工 1 子图

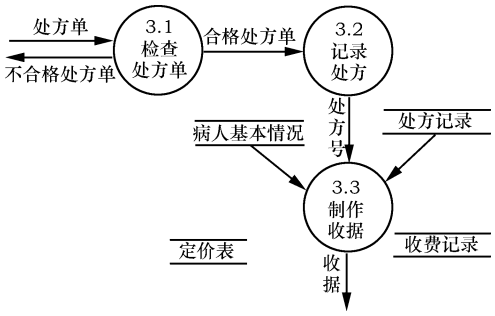


图 5 医院收费系统的加工 2 子图

【问题 1】

指出哪张图的哪些文件可以不必画出。

【问题 2】

数据流图 4 中缺少 2 条数据流，请直接在图中添加。

【问题 3】

数据流图 5 中缺少 4 条数据流，请直接在图中添加。

●试题二

阅读以下说明和流程图，回答问题 1 和问题 2，将答案写在答卷的对应栏内。

【说明】

某供销系统接受顾客的订货单，当库存中某配件的数量小于订购量或库存量低于一定数量时，向供应商发出采购单；当某配件的库存量大于或等于定购粮食，或者收到供应商的送货单并更新了库存后，向顾客发出提货单。该系统还可随时向总经理提供销售和库存情况表。该供销系统的分层数据流图中部分数据流和文件的组成如下： 文件

配件库存=配件号+配件名+规格+数量+允许的最低库存量

数据流

订货单=配件号+配件名+规格+数量+顾客名+地址

提货单=订货单+金额

采购单=配件号+配件名+规格+数量+供应商名+地址

送货单=配件号+配件名+规格+数量+金额

假定顶层图(如图 6 所示)是正确的，″供应商″文件已由其他系统生成。

【问题 1】

指出哪张图中的哪些文件可不必画出。

【问题 2】

指出在哪些图中遗漏了哪些数据流。回答时使用如下形式之一：

(1) XX 图中遗漏了 XX 加工(或文件)流向 XX 加工(或文件)的 XX 数据流；

(2) XX 图中 XX 加工遗漏了 XX 输入(或输出)数据流。

【流程图】

顶层图

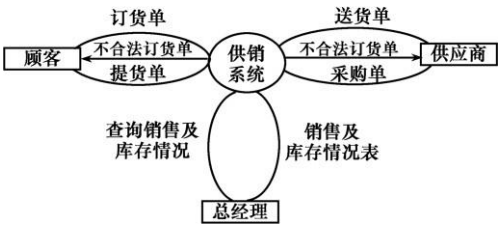


图 6

0 层图

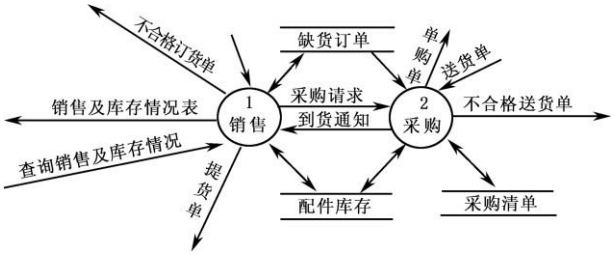


图 7

加工 1 子图

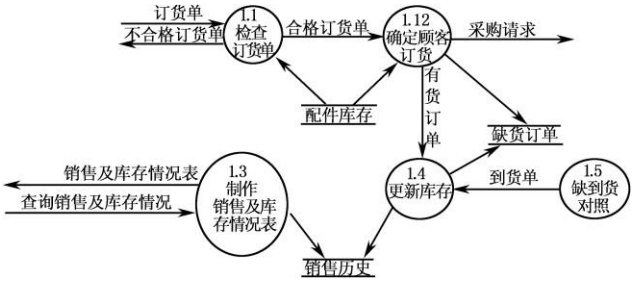


图 8

加工 2 子图

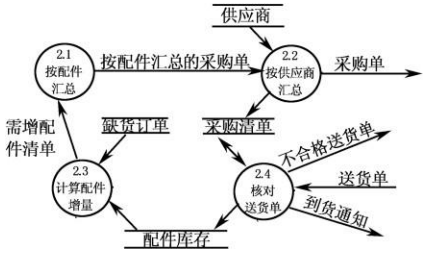


图 9

●试题三

有下列关于运动会管理系统的 ER 图，如图 10 所示。图中矩形表示实体，圆表示属性，双圆表示关键字属性，菱形表示实体之间的关系。假定已通过下列 SQL 语言建立了基本表。

CREATE TABLE ATHLETE
(ANO CHAR (6) NOT NULL，
ANAME CHAR (20) ，
ASEX CHAR (1) ，
ATEAM CHAR (20))；
CREATE TABLE ITEM
(INO CHAR (6) NOT NULL，
INAME CHAR (20) ，
ITIME CHAR (12) ，
IPLACE CHAR (20) ；
CREATE TABLE GAMES
(ANO CHAR (6) NOT NULL，

INO CHAR (6) NOT NULL，

SCORRE CHAR (10));

为了答题的方便，图中的实体和属性同时给出了中英文两种文字，回答问题时只需写出英文名即可。

【E-R 图】

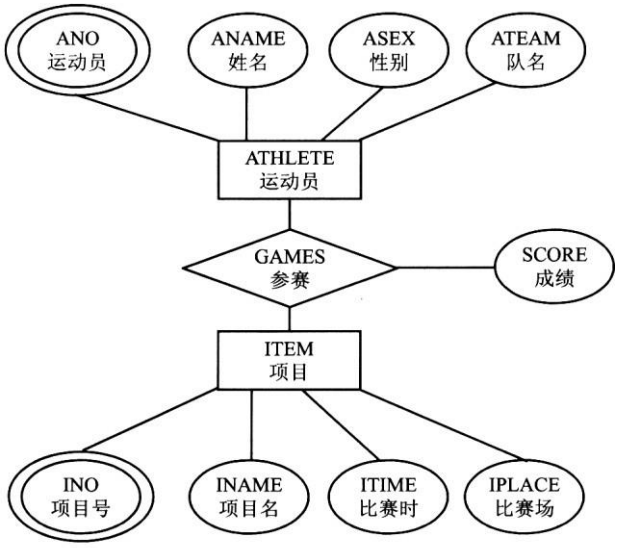


图 10 E-R 图

【问题】

填充下列 SQL 程序 1～4 中的 (1) ～ (7) ，使它们分别完成相应的功能：

程序 1：统计参加比赛时男运动员人数。

SELECT (1)

FROM ATHLETE

WHERE ASEX=′ M′ ;

程序 2：查 100872 号运动员参加的所有项目及其比赛时间和地点。

SELECT ITEM，INO，IN AME，ITIME，IPLACE

FROM GAMES，ITEM

WHERE (2) ;

AND (3) ;

程序 3：查参加 100035 项目的所有运动员名单。

SELECT ANO，ANAME，ATEAM

FROM ATHLETE

WHERE (4) ;

(SELECT (4) (5)

FROM GAMES

WHERE GAMES.ANO=ATHLETE.ANO AND INO=’100035’);

程序 4：建立运动员成绩视图。

(6) ATHLETE-SCORE

AS SELECT ATHLETE，ANO，ANAME，ATEAM，INAME，SCORE

FORM (7) WHERE ATHLETE.ANO=GAMES.ANO AND GAMES.INO=ITEM.INO;

●试题四

阅读下列程序说明和 C 程序，将应填入(n)处的字句写在答卷纸的对应栏内。

【程序说明】

该程序定义了两个子函数 strsort 和 strmerge。它们分别实现了将一个字符串按字母顺序排序和将两个字符串合并排序，并删去相同字符。在主函数里，先输入两个字符串 s1 和 s2，然后调用 strsort 函数对它们分别排序，然后

调用 strmerge 函数将 s1 和 s2 合并，将合并后的字符串赋给字符串 s3，最后输出字符串 s3。

【程序】

```
#include<stdio.h>
void strmerge(char*a，char*b，char*c)//将字符串 a，b 合并到字符串c 中
    {   ch
ar t,*w；
w=c；
while(  (1)  )
 {
//找到字符串 a，b 当前字符中较小的字符
if(*a<*b)
{
t=*a；
  (2)  ；
}
else if(*a>*b)
{
t=*b;
  (3)  ；
}
else//字符串 a，b 当前字符相等
{  t=*a ；
a++;
b++;
}
if(  (4)  )//开始，可直接赋值
*w=t;
else if(t!=*w)
//如果 a，b 中较小的当前字符与 c 中当前字符不相等，才赋值  (5)  ；
 }
if(*a!=\′ ＼0′ )//如果字符串 a 还没有结束，则将 a 的剩余部分赋给 c while(*a!=′
＼0′ )
if(*a!=*w)
{
*(++w)=*a；
a++;
}
else
  (6)  ；
 if(*6!=′ ＼0′ )//如果字符串 b 还没有结束，则将b 的剩余部分赋给 c
while(*b!=′ ＼0′ )
if(*b!=*w)
{
*(++w)=*b；
b++;
```

```
}
else
b++;
  (7)  ;
}
void strsort(char*s)//将字符串 S 中的字符排序
{
int i，j，n；
char t，*w;
W=S;
for(n=0；*w!=′ ＼0′ ； n++)//得到字符串长度 n
w++;
for(i=0；i<n-1；i++)//对字符串 s 进行排序，按字母先后顺序
for(j=i+1；j<n；j++)
if(  (8)  )
{
t=s[i]；    s[i]=s[j]；    (9)  ；
}
}
void main()
{
char s1[100]，s2[100]，s3[100]；    printf("＼nPlease，input the first string：")；
scanf("％s"，s1)；
printf("＼nPlease input the second string：")；
scanf("%s"，s2)；
strsort(s1)；//将字符串 s1 排序
strsort(s2)；//将字符串 s2 排序
printf("％s＼n"，s1)；  printf("％
s＼n"，s2)；
s3[0]=′ ＼0′ ； //字符串 s3 的第一个字符先置′ ＼0′结束标志
    (10)  //将 s1 和 s2 合并，按照字母顺序排列，
 //且要删去相同字符，存入 s3 中
printf("%s"，s3)；
 }
```

●试题五

阅读下列程序说明和 C 代码，将应填入(n)处的字句写在答题纸的对应栏内。

【程序 5 说明】

著名的四色定理指出任何平面区域图均可用四种颜色着色，使相邻区域着不同的颜色。本程序对给定的区域图找出所有可能的不超过四种颜色的着色方案。

程序中用 1~4 表示四种颜色。要着色的N 个区域用 0~N-1 编号，区域相邻关系用 adj［］［］矩阵表示，矩阵的 i 行 j 列的元素为 1，表示区域 i 与区域 j 相邻；矩阵的 i 行 j 列的元素为 0，表示区域 i 与区域 j 不相邻。数组 color［］用来存储着色结果，color［i］的值为区域 i 所着颜色。

【程序 5】

```
#include<stdio.h>
#define N 10
```

```
void output(int color［］)/*输出一种着色方案*/
{int i;
for(i=0;i<N;i++)
printf("%4d"，color［i］);
printf("＼n");
}
int back(int*ip,int color［］)/*回溯*/
{int c=4;
while(c==4){ if(*i
p<=0)return 0;
--(*ip);
c=  (1)  ;
color［*ip］=-1;
}
return c;
}
/*检查区域 i，对 c 种颜色的可用性*/
int color0k(int i,int c,int［］［N］,int color［］}
{int j;
for(j=0;j<i;j++)
if(  (2)  )
return 0;
return 1;
}
/*为区域 i 选一种可着的颜色*/
int select(int i,int c,int adj［］［N］,int color［］)
{int k;
for(k=c;k<=4;k++)
if(colorOK(  (3)  ))
return k;
return 0;
}
int coloring(int adj［］［N］)/*寻找各种着色方案*/
{int color［N］,i,c,cnt;
for(i=0;i<N;i++)color［i］=-1;
i=c=0;cnt=0;
while  (1)
_____{ if((c
=  (4)  )==0){
c=back(&i,color);
if(c==0)return cnt;
}else{  (5)  ;i++;
if(i==N){ output(c
olor);
++cnt;
c=back(&i,color);
```

```
}else c=0;
```

```
}
}
}
void main()
{int adj［N］［N］=
{{0，1，0，1，1，1，1，1，1，1},
{1，0，1，1，0，1，1，1，1，0},
{0，1，0，1，0，1，1，0，1，1},
{1，1，1，0，1，1，0，0，1，1},
{1，0，0，1，0，1，0，0，0，0},
{1，1，1，1，1，0，1，0，0，1},
{1，1，1，0，0，1，0，0，1，0},
{1，1，0，0，0，0，0，0，1，1},
{1，1，1，1，0，0，1，1，0，1},
{1，0，1，1，0，1，0，1，1，0}
};
printf("共有%d 组解.＼n",coloring(adj));
}
```

●试题六

阅读下列程序说明和 C++代码，将应填入(n)处的字句写在答卷的对应栏内。

【程序 6 说明】

本程序实现两个多项式的乘积运算。多项式的每一项由类 Item 描述，而多项式由类 List 描述。类 List 的成员函数有：

createList()：创建按指数降序链接的多项式链表，以表示多项式。

reverseList()：将多项式链表的表元链接顺序颠倒。

multiplyList(List L1，List L2)：计算多项式 L1 和多项式 L2 的乘积多项式。

【程序 6】

```
#include<iostream.h>
class List；
class ltem{
friend class List；
private：
double quot；
int exp；
Item*next；
public：
Item(double_quot，int_exp)
{  (1)  ；}
}；
class
List{ priva
te:
Item*list；
public：
```

```
List(){list=NULL；}
void reverseList()；
```

```
void multiplyList(List L1，List L2)；
void createList()；
}；
void List::createList()
{Item*p，*u，*pre；
int exp；
double quot；
list=NULL；
while  (1)  {
cout<<"输入多项式中的一项(系数、指数)："<<endl；
cin>>quot>>exp；
if(exp<0)break；//指数小于零，结束输入
if(quot==0)continue；
p=list；
while(  (2)  ){//查找插入点
pre=p；p=p->next；}
if(p!=NULL&&exp==p->exp){ p->quot+=quot；continue；}
u=  (3)  ；
if(p==list)  list=u ；
else  pre->next=u ；
u->next=p；}
}
void List::reverseList()
{Item*p，*u；
if(list＝＝NULL)return；
p=list->next；list->next=NULL；
while(p!=NULL){
u=p->next；p->next=list；
list=p；p=u；}
}
void List::multiplyList(List L1，List L2)
{Item*pLl，*pL2，*u；
int  k，maxExp；
double quot；
maxExp=  (4)  ；
L2．reverseList()；list=NULL;
for(k=maxExp；k>=0；k--
){ pL1=L1．list；
while(pL1!=NULL&&pL1->exp>k)pL1=pL1->next；
pL2=L2．list；
while(pL2!=NULL&&  (5)  pL2=pL2->next；
quot=0．0；
while(pL1!=NULL&&pL2!=NULL){
if(pL1->exp+pL2->exp==k){
  (6)  ；
```

pL1=pL1->next；pL2=pL2->next；

```
}else if(pL1->exp+pL2->exp>k)pL1=pL1->next；
else pL2=pL2->next；
}
if(quot!=0．0){
u=new Item(quot，k)；
u->next=list；list=u；}
}
reverseList()；L2．reverseList()；
}
void main()
{ListL1，L2，L；
cout<<"创建第一个多项式链表＼n"；L1．createList()；
cout<<"创建第二个多项式链表＼n"；L2．createList()；
L．multiplyList(L1，L2)；
}
```

●试题七

【说明】

下面是一个 Applet 程序，其功能是根据给出的小时，分钟和秒数计算相等的秒数，即将 1 分钟化为 60 秒，依此类推。要求建立一个时间类，时间参数均作为类的成员变量，并且给出换算时间的方法，也作为这个类的成员函数，可以供外部对象进行调用。同时还需要在输出窗口中显示换算结果，并且将结果写到 out3_3.txt 文件中，本题给出确定的时间为 4 小时 23 分 47 秒，要求换算成以秒做单位的时间。

程序运行结果如图 11 所示。

图 11

```
import javA．io.*;
import javA．awt.*;
import javA．applet.*;
/*
<applet code=ex7_7.class width=800 height=400>
</applet>
*/
public class ex7_7 extends
Applet{ public void paint(Graphics
g){
int nSum;
class
```

```
myTime7_7{ publi
c int h;
```

```
public int m;
public int s;
public int out;
public int caculateSecond(){
  (1)  ;
return out;
}
}
myTime7_7 objTime7_7 = new myTime7_7();
objTime7_7.h = 4;
objTime7_7.m = 23;
objTime7_7.s = 47;
nSum = objTime7_7.  (2)  ;
g.drawString ("时："+objTime7_7.h, 20, 30);
g.drawString ("分："+objTime7_7.m, 20, 50);
g.drawString ("秒："+objTime7_7.s, 20, 70);
g.drawString (  (3)  );
try {
FileOutputStream fos7_7 = new FileOutputStream("out7_7.txt");
BufferedOutputStream bos7_7=new BufferedOutputStream(fos7_7,1024);
PrintStream ps7_7=new PrintStream(bos7_7,false); System.setOut(ps7_7);
System.out.println(  (4)  );
ps7_7.close();
} catch(IOException ioe) {
  (5)  (ioe);
}
}
}
```

ex7_7.html

```
<HTML>
<HEAD>
<TITLE>ex7_7</TITLE>
</HEAD>
<BODY>
<applet    code="ex7_7.class" width=800 height=400 >
</applet>
</BODY>
</HTML>
```