软考设计师模拟试题 4（上午题）

●已知文法 G[A]，它定义的语言描述为　(1)　。

G[A]:A → 0B｜1C

B → 1｜1A｜0BB

C → 0｜0A｜1CC

(1) A．G[A]定义的语言由 0、1 符号串组成，或者串中 1 的个数是 0 的个数 2 倍，或者串中 0 的个数是 1 的个数 2 倍

B．G[A]定义的语言由 0、l 符号串组成，串中 0 的个数是 1 的个数 2 倍

C．G[A]定义的语言由 0、1 符号串组成，串中 1 的个数是 0 的个数 2 倍

D．G[A]定义的语言由 0、1 符号串组成，串中 0 和 1 的个数相同

●利用并行处理技术可以缩短计算机的处理时间，所谓并行性是指　(2)　。可以采用多种措施来提高计算机系统的并行性，它们可分成三类，即　(3)　。

提供专门用途的一类并行处理机(亦称阵列处理机)以　(4)　方式工作，它适用于　(5)　。多处理机是目前较高性能计算机的基本结构，它的并行任务的派生是　(6)　。

(2) A．多道程序工作

B．多用户工作

C．非单指令流单数据流方式工作

D．在同一时间完成两种或两种以上工作

(3) A．多处理机、多级存储器和互连网络

B．流水结构、高速缓存和精简指令集

C．微指令、虚拟存储和 I / O 通道

D．资源重复、资源共享和时间重叠。

(4) A．SISD

B．SIMD

C．MISD

D．MIMD

(5) A．事务处理

B．工业控制

C．矩阵运算

D．大量浮点计算

(6) A．需要专门的指令来表示程序中并发关系和控制并发执行

B．靠指令本身就可以启动多个处理单元并行工作

C．只执行没有并发约束关系的程序

D．先并行执行，事后再用专门程序去解决并发约束

●软件的易维护性是指理解、改正、改进软件的难易程度。通常影响软件易维护性的因素有易理解性、易修改性和　(7)　。在软件的开发过程中往往采取各种措施来提高软件的易维护性。如采用　(8)　有助于提高软件的易理解性；　(9)　有助于提高软件的易修改性。在软件质量特性中，　(10)　是指在规定的一段时间和条件下，与软件维持其性能水平的能力有关的一组属性；　(11)　是指防止对程序及数据的非授权访问的能力。

(7) A．易使用性

B．易恢复性

C．易替换性

D．易测试性

(8) A．增强健壮性

B．信息隐蔽原则

C．良好的编程风格

D．高效的算法

(9) A．高效的算法

B．信息隐蔽原则

C．增强健壮性

D．身份认证

(10) A．正确性

B．准确性

C．可靠性

D．易使用性

(11) A．安全性

B．适应性

C．灵活性

D．容错性

●在 CORBA 体系结构中， (12) 属于客户端接口。

(12) A．静态 IDLSkeletons

B．POA

C．静态 IDLStubs

D．动态 Skeletons

●（13）是以科学、技术和实践经验的综合成果为基础，对重复性事物和概念所做的统一规定，经有关方面协商一致，由一个公认机构或主管机构的批准，以特定形式发布作为共同遵守的准则和依据。

(13) A.标准化

B.协议

C.标准

D.工作流程

●从信息资源管理的观点出发，信息系统的逻辑结构一般由四部分组成，其中 (14) 利用信息系统提供的信息进行决策和选择，是信息系统服务的对象。

(14) A．信息源

B．信息处理器

C．信息使用者

D．信息管理者

●桌上有一个空盒，盒内只允许放一个水果。爸爸专向盒内放苹果，妈妈专向盒内放桔子，儿子等着吃盒中的水果(苹果或桔子)。若盒内已有水果，放者必须等待，若盒内没有水果，吃者必须等待。用 PV 操作来协调 3 人的关系。请回答下列问题：

①应设置的信号量及其初值为 SP 和 1。

②在 3 组工作流程的虚线位置填上适当的 PV 操作，实现 3 人正确的活动。

爸：准备

P(SP)

向盒内放苹果

(15)

妈：准备

(16)

向盒内放桔子

(17)

儿：

(18)

拿盒中的水果(苹果或桔子)

(19)
吃水果(苹果或桔子)
(15) ~(19) A．P(SP)
B．P(SG)
C．V(SG)
D．V(SP)
●一进程刚获得 3 个主存块的使用权，若该进程访问页面的次序是{1，3，2，1，2，1，5，1，2，3}。当采用先进先出调度算法时，发生缺页次数是 (20) 次，而采用 LRU 算法时，缺页数是 (21) 次。
(20) ,(21) A．3
B．4
C．5
D．6
●试对各种内部排序算法进行比较(见表 1)。

表 1 排序算法比较表

排序算法	最好情况及时间	最坏情况及时间	平均时间	稳定性
直插	正序：$O(n)$	反序：$O(n^2)$	$O(n^2)$	稳定
直选	与初始序列无关。$O(n^2)$	与初始序列无关。$O(n^2)$	$O(n^2)$	不稳
冒泡	(25)	反序：$O(n^2)$	$O(n^2)$	(22)
快速	杂乱无序：$O(n\log n)$	正、反序：$O(n^2)$	(23)	不稳
堆	(26)	初始序列关系不大。$O(n\log n)$	$O(n\log n)$	不稳
归并	与初始序列无关。$O(n\log n)$	与初始序列无关。$O(n\log n)$	$O(n\log n)$	(24)
基数	与初始序列无关。$O(d,n+d,rd)$	与序列无关。$O(d.n+d.rd)$	$O(d.n+d.rd)$	稳定
Shell	正序：移动次数为 0，比较约 $n^{1.25}$	反序：约 $1.6n^{1.3}$	$n^{1.3}$	不稳

(22),(24) A．稳定
B．不稳定
C．正序
D．反序
(23) A．O(n2)
B．O(nlogn)
C．O(d,n+d,rd)
D．O(n)
(25) ,(26) A．O(n)
B．O(n2
C．O(nlogn)
D．(n2-1)
●属于局域网功能的是 (27) 。
(27) A．内部网络之间的信息共享
B．系统的信息处理
C．视频电影的观看
D．文件的传输
●当数据分组从低层向高层传送时，分组的头要被 (28) 。
(28) A．加上
B．去掉
C．重新处置
D．修改
●千兆以太网比快速以太网有 (29) 数据传输率和 (30) 的碰撞域。
(29) A．相同
B．小

C．高
D．不能确定
(30) A．大
B．小
C．相同
D．不能确定
●语言 L={ambn｜m≥0,n≥1}的正规表达式是 (31) 。
(31) A．a*bb*
B．aa*bb*
C．aa*b*
D．a*b*
●从下面的选项中选出正确的答案在宏定义：#define MAXINT 324 中，宏名 MAXINT 代替的是 (32) 。
(32) A．整型数
B．实型数
C．常量
D．一串字符
●面向对象技术中，对象是类的实例。对象有 3 种成分： (33) 、属性和方法(或操作)。
(33) A．标识
B．继承
C．封装
D．消息
●下列叙述中正确的是 (34) 。
(34) A．宏替换不占用运行时间
B．在带参的宏定义中，要定义其形式参数的类型
C．在带参的宏定义中，形式参数是变量
D．在带参的宏定义中，形式参数是常量
●设学生 S、课程 C、学生选课 SC 的关系模式分别为：
S(Sno，Sname，Sage，Saddr)、C(Cno，Cname，Pcno)以及 SC(Sno，Cno，Grade)与关系代数表达式 π Sno,Sname,Gr(β Sname=′数据库′(S｜×｜SC｜×｜C))等价的元组演算表达式为：
{ (35) S(u)∧SC(v)∧C(w)∧ (36) ∧ (37) }
(35) A．(u)(瑎 v)(瑎 w)
B．(瑎 u)(v)(瑎 w)
C．(瑎 u)(瑎 v)(瑎 w)
D．(瑎 u)(瑎 v)(w)
(36) A．u［1］=v［1］∧v［1］=w［1］∧w［1］=′数据库′
B．u［1］=v［2］∧v［2］=w［1］∧w［3］=′数据库′
C．u［1］=v［1］∧v［2］=w［1］∧w［2］=′数据库′
D．u［2］=v［2］∧v［1］=w［2］∧w［2］=′数据库′
(37) A．t［1］=u［1］∧t［2］=u［2］∧t［3］=v［3］
B．t［1］=u［1］∧t［2］=u［2］∧t［3］=v［2］
C．t［1］=u［1］∧t［2］=w［1］∧t［3］=v［2］
D．t［1］=u［1］∧t［2］=w［2］∧t［3］=v［3］
●若有关系模式 R(A，B，C)和 s(C，D，E)，对于如下的关系代数表达式：
E1=∏A，D(σB＜′2003′∧R.
C．=S.C∧E=′80′∧∧(R×S))
E2=∏A，D(σR.C=S.C(σB＞′2003′)(R)×σE=′80′(S))

E3=∏A，D(σB＜′2003′(R)驮妒σE=′80′(S))

E4=∏A，D(σB＜′2003′∧E=′80′∧(R 驮妒 S))正确的结论是 (38) ，表达式 (39) 的查询效率最高。

(38) A．E1=E2=E3=E4

B．E3=E4 但 E1≠E2

C．E1=E2 但 E3≠E4

D．E3≠E4 但 E2=E4

(39) A．E1

B．E2

C．E3

D．E4

●对长度为 n 的顺序存储的有序表进行二分查找时，其对应的判定树的高度为 (40) 。

(40) A．n

B．［log2n］

C．［log2(n+1)］

D．［log2n+1］

●已知一个有序表为(13，18，24，35，47，50，62，83，90，115，134)，当二分查找值为 90 的元素时，查找成功的比较次数为 (41) 。

(41) A．1

B．2

C．3

D．4

●对于一个线性表既要求能够进行较快的插入和删除，又要求存储结构能够反应数据之间的逻辑关系，则应该用 (42) 。

(42) A．顺序方式存储

B．链接方式存储

C．散列方式存储

D．以上方式均可

●在含 n 个顶点和 e 条边的无向图的邻接矩阵中，零元素的个数为 (43) 。

(43) A．e

B．2e

C．n2-e

D．n2-2e

●模块内聚度用于衡量模块内部各成分之间彼此结合的紧密程度。

一个语句在程序的多处出现，为了节省内存空间把这些语句放在一个模块中，该模块的内聚度是 (44) 的。

(44) A．偶然性

B．顺序性

C．通信性

D．过程性

●在结构测试用例设计中，有语句覆盖、条件覆盖、判定覆盖(即分支覆盖)、路径覆盖等，其 (45) 是最强的覆盖准则。为了对图 1 所示的程序段进行覆盖测试，必须适当地选取测试数据组。若 x 和 y 是两个变量，可供选择的测试数据组共有 I、Ⅱ、Ⅲ、Ⅳ四组(见表 2)，则实现判定覆盖至少应采用的测试数据组是 (46) ;实现条件覆盖至少采用的测试数据组是 (47) ，实现路径覆盖至少应采用的测试数据组是 (48) 或 (49) 。

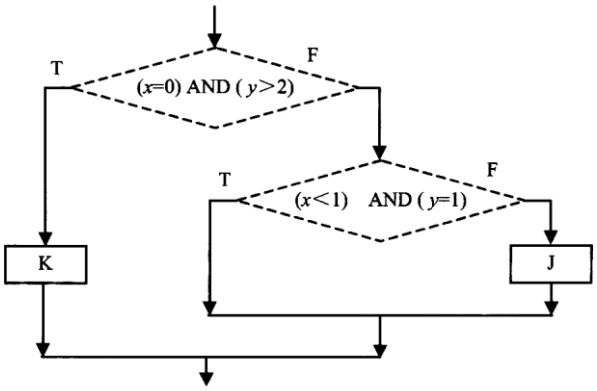

图 1　程序段

(45) A．语句覆盖
B．条件覆盖
C．判定覆盖
D．路径覆盖

(46) ，(47) A．I 和Ⅱ组
B．Ⅱ和Ⅲ组
C．Ⅲ和Ⅳ组
D．I 和Ⅳ组

(48) ，(49) A．I、Ⅱ和Ⅲ组
B．Ⅱ、Ⅲ和Ⅳ组
C．I、Ⅲ和Ⅳ组
D．I、Ⅱ和Ⅳ组

●对长度为 n 的顺序表进行顺序查找的时间复杂度为 (50) 。

(50) A．O(n)
B．O(［log2n］)
C．O(［log2］(n+1))
D．O(n2)

●多媒体音频处理中，人所敏感的声频最高为 (51) (Hz)，因此数字音频文件中对音频的采样频率为 (52) (Hz)。对一个双声道的立体声，保持 1 秒钟声音，波形文件所需的字节数为 (53) ，这里假设每个采样点的量化数为 8 位。MIDI 文件是最常用的数字音频文件之一，MIDI 是一种 (54) ，它是该领域国际上的一个 (55) 。

(51) A．50k
B．10k
C．22k
D．44k

(52) A．44.1k
B．20.05k
C．10k
D．88k

(53) A．22050
B．88200
C．176400
D．44100

(54) A．语音数字接口
B．乐器数字接口

C．语音模拟接口
D．乐器模拟接口
(55) A．控制方式
B．管理规范
C．通信标准
D．输入格式

●采用可变长子网掩码技术可以把大的网络分成小的子网，例如把子网掩码为 255.255.0.0 的网络 40.15.0.0 分为两个子网，假设第一个子网为 40.15.0.0／17，则第二个子网为 (56) 。假设用户 X1 有 2000 台主机，则至少应给他分配 (57) 个 C 类网络，如果分配给用户 X1 的网络号为 196.25.64.0，则指定给 X1 的子网掩码为 (58) ；假设给用户 X2 分配的 C 类网络号为 196.25.16.0～196.25.31.0，则 X2 的子网掩码应为 (59) ：如果路由器收到一个目标地址为 11000100.00011001.01000011.00100001 的数据报，则该数据报应送给 (60) 用户。

(56)A.40.15.1.0／17B.40.15.2.0／17
C.40.15.100.0／17D.40.15.128.0／17

●考查下列文法：G(VT，VN，E，P)
其中：VT={+，*，(，)，i}
VN={E,T,F}
E 是开始符号
P：
E→E+T|T
T→T*F|F
F→(E)|i
F*F+T 是该文法的一个句型，其中， (61) 是句柄， (62) 是素短语。 (63) 是该句型的直接推导， (64) 是该句型的最左推导。 (65) 是该文法的一个句子。

(61) A．F
B．F*F
C．F+T
D．F*F+，T
(62) A．F
B．F*F
C．F+T
D．F*F+T
(63) A．F*F+i
B．F*F+T*F
C．F*F+F*F
D．i*i+T
(64) A．F*F+T*F
B．F*F+T
C．F*(E)+T
D．(E)*F+T
(65) A．T+(i+i)
B．i+(i+F)
C．i
D．(E)

●Prior to the UML，there was no clearleading (66) language.Users had to choose from among many similar modeling languages with minor differences in overall (67) power.Most of the modeling languages shared a set of commonly accepted concepts that are expressed slightly differently in various languages.This lack of (68)

discouraged new users from entering the OO market and from doing OO modeling，without greatly expanding the power of modeling.Users longed for the industry to adopt one，or a very few，broadly supported modelinglanguages suitable for (69) usage.

Some Vendors were discouraged from entering the OO modeling area because of the need to support many similar，but slightly different，modeling languages.In particular，the supply of add-ontools has been depressed because small vendors cannot afford to support many different formatsfrom many different (70) modeling tools.It is important to the entire OOindustry to encourage broadly based tools and vendors，as well as niche products that cater to the needs of specialized groups.

(66) A．programming
B．modeling
C．formal
D．intelligent

(67) A．control
B．expressive
C．conductive
D．interactive

(68) A．agreement
B．understanding
C．characteristic
D．diversitv

(69) A．distincitive
B．special-purpose
C．separate
D．general-purpose

(70) A．internal
B．external
C．front-end
D．back-end

●MIMD systems can be classified into(71) oriented systems,high availability systems and response oriented systems.The goal of(71).orlented multiprocessing is to obtain high(71)(72)minimal computing cost.The techniques employed by multiprocessor operating systems to achieve this goal take advantage of an inherent processing versus input/output balance in the workload to produce(73)and(74)loading of system(75).

(71)A.though B.through C.throughout D.throughput

(72)A.at B.of C.on D.to

(73)A.balance B.balanced C.balances D.balancing

(74)A.uniform B.unique C.unit D.united

(75)A.resource B.resources C.source D.sources

resource B.resources C.source D.sources

软考设计师模拟试题 4（下午题）

●试题一

阅读以下算法说明和流程图，回答问题 1 和问题 2。

【算法说明】

下面是一段插入排序的程序，将 R[k+1]插入到 R[1…k]的适当位置。　　R[0]=R[k+1]；　　j=k；

```
while (R[j]>R[0])
{
R[j+1]=R[j]；　　j--；
}
R[j+1]=R[0]；
```

【流程图】

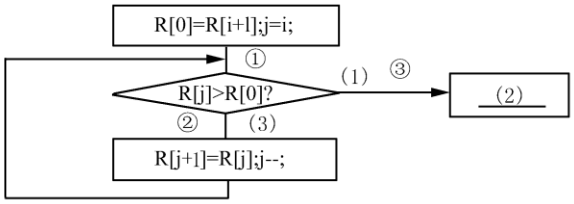

图 2　算法的流程图

【测试用例设计】

(while 循环次数为 0、1、2 次)

表 3　测试用例表

循环次数	输入数据					R[0]	预期结果					覆盖路径	
	j	R[j-2]	R[j-1]	R[j]	R[j+1]		j	R[j-2]	R[j-1]	R[j]	R[j+1]	判断	路径
0	i	–	–	1	2	2	i	–	–	1	2	<	(4)
	i	–	–	1	1	2	i	–	–	1	1	=	⑤
1	i	–	1	3	2	2	i-1	–	1	2	3	(7)	②④⑥
	i	–	3	3	2	2	i-1	–	(8)	2	(9)	>=	②④⑤
2	i	1	3	4	2	2	i-2	1	2	3	4	> > <	(3)
	i	2	3	4	2	2	i-2	2	2	3	4	> > =	(6)

【问题 1】

指出算法的流程图中 (1) ～ (3) 处的内容。

【问题 2】

指出测试用例设计中 (4) ～ (9) 处的内容。

●试题二

阅读以下说明和流程图，回答问题 1 至问题 3，将解答填入答题纸的对应栏内。

【说明】

某城市电信局受理了许多用户申请在指定电话上开设长话业务。长话包括国内长途和国际长途。电信局保存了长话用户档案和长话业务档案。

长话用户档案的记录格式为

用户编码	用户名	用户地址

长话业务档案的记录格式为

电话号码	用户编码	国内长途许可标志	国际长途许可标志

电话用户每次通话的计费数据都自动地记录在电信局程控交换机的磁带上。计费数据的记录格式为

日期	电话号码	受话号码	通话开始时间	通话持续时间

该电信局为了用计算机自动处理长话收费以提高工作效率，开发了长话计费管理系统。该系统每月能为每个长话用户打印出长话缴费通知单。长话缴费通知单的记录格式为

用户名	用户地址	国内长途话费	国际长途话费	话费合计

流程图描述了该系统的数据处理过程。

该系统每天对原始的计费数据进行分类排序，并确定每个通话记录的通话类型(市话/国内长途/国际长途)，再根据话费单价文件算出每个通话记录应收取的话费。因此，形成的日计费文件中，增加了两个数据项：通话类型和话费。该系统每日对日计费文件进行累计(按电话号码和通话类型，对该类型的话费进行累计，得到该电话号码

该通话类型的当月话费总计)，形成月计费文件。
月计费文件经过长话出账处理形成长话账单文件。长话账单文件的记录格式为

月份	用户编码	电话号码	国内长途话费	国际长途话费	话费总额

长话账单文件经过处理 5 和处理 6 的处理后，就可以形成长话缴费通知单。
【问题 1】
(1) 请说明流程图 1 中的文件 F0、F1 分别是哪个文件。
(2) 处理 1 和处理 5 分别按照哪些数据项进行分类?
【问题 2】
处理 4 能发现哪些错误(不需考虑设备故障错误)?
【问题 3】
说明处理 6 的功能。
【流程图】(如图 3 所示)

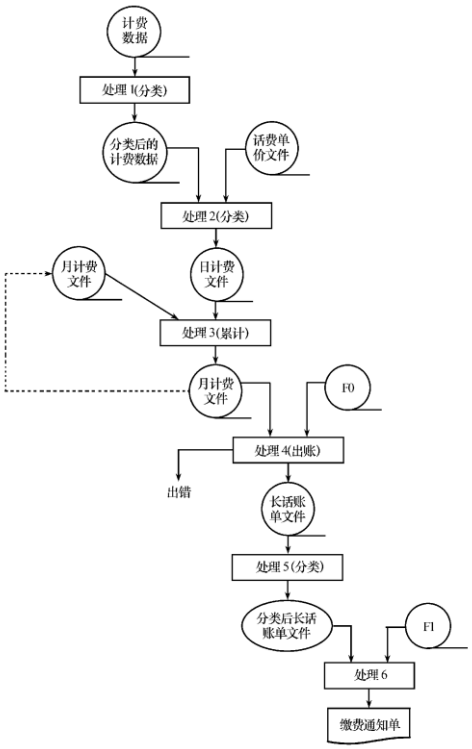

图 3
s●试题三
阅读下列函数说明和 C 函数，将应填入(n)处的字句写在答题纸的对应栏内。
【函数 3 说明】
函数 DeleteNode(Bitree*r,int e)的功能是：在树根结点指针为 r 的二叉查找(排序)树上删除键值为 e 的结点，若删除成功，则函数返回 0，否则函数返回-1。二叉查找树结点的类型定义为：

```
typedef struct Tnode{
int data;/*结点的键值*/
struct Tnode*Lchild,*Rchild;/*指向左、右子树的指针*/
}*Bitree;
```

在二叉查找树上删除一个结点时，要考虑三种情况：
①若待删除的结点 p 是叶子结点，则直接删除该结点;
②若待删除的结点 p 只有一个子结点，则将这个子结点与待删除结点的父结点直接连接，然后删除结点 p;
③若待删除的结点 p 有两个子结点，则在其左子树上，用中序遍历寻找关键值最大的结点 s ,用结点 s 的值代替结点 p 的值，然后删除结点 s，结点 s 必属于上述①、②情况之一。

【函数 3】

```
int DeleteNode(Bitree*r,int e){
Bitree p=*r,pp,s,c;
while(  (1)  ){/*从树根结点出发查找键值为 e 的结点*/
pp=p;
if(e＜p-＞data)p=p-＞Lchild;
else p=p-＞Rchild;
}
if(!p)return-1;/*查找失败*/
if(p-＞Lchild &&p-＞Rchild) { /*处理情况③*/
s=  (2)  ;pp=p;
while(  (3)  ){pp=s;s=s-＞Rchild;}
p-＞data=s-＞data;p=s;
}
/*处理情况①、②*/
if(  (4)  )c=p-＞Lchild;
else c=p-＞Rchild;
if(p==*r)*r=c;
else if(  (5)  )pp-＞Lchild=c;
else pp-＞Rchild=c;
free(p);
return 0;
}
```

●试题四

请补充函数 fun()，该函数的功能是将字符串 tt 中的大写字母都改为对应的小写字母，其他字符不变。例如，若输入"Are you come from Sichuan?"，则输入"are you come from sichuan?"。

注意：部分源程序给出如下。

请勿改动主函数 main 和其他函数中的任何内容，仅在函数 fun()的横线上填入所编写的若干表达式或语句。

试题程序：

```
#include<stdio.h>
#include<string.h>
#include<conio.h>
char *fun(char tt［］)
{
int i;
for(i=0;tt［i］;i++)
{
if((tt［i］>=′A′)&&(  (1)  ))
  (2)  ;
}
return (  (3)  )  ;
}
main()
{
char tt［81］;
```

```
printf(" \ nPlease enter a string:");
gets(tt);
printf(" \ nThe result string is:  \ n%s",
fun(tt));
}
```

●试题五

阅读下列程序说明和 C++代码，将应填入(n)处的字句写在答卷的对应栏内。

【说明】

①在类体中添加函数 move(double ax, double ay)的定义，使得点的坐标 x 和 y 分别移动 ax 和 ay 个单位。

②在类定义外完成重载的两个构造函数 CPosition()和 CPosition(double dx, double dy)，其中前者为不带参数的构造函数，使 CPosition 对象的默认值为 x=0，y=0，后者为带参数的构造函数，把数据成员 x 和 y 分别初始化为参数 dx 和 dy 的值。

③完成函数 double distance(double bx, double by)的定义，该函数返回*this 和点(bx,by)的距离。

注意：除在指定的位置添加语句外，请不要改动程序中的其他语句。

源程序文件 test5.cpp 清单如下：

```
#include<iostream.h>
#include <math.h>
class CPosition
{
public:
CPosition();
CPosition(double dx, double dy);
double getx();
double gety();
  (1)
double distance(double bx, double by);
private:
double x;
double y;
};
  (2)
{
x=0; y=0;
}
CPosition::CPosition(double dx, double dy)
{
x=dx; y=dy;
}
double CPosition::getx()
{
return x;
}
double CPosition::gety()
{
return y;
```

```
}
double CPosition::distance(double bx, double by)
{
   (3)
}
void main()
{
double a,b;
cout << "Input x, y position of a point: ";
cin >> a >> b;
CPosition psA(a, b);
cout << "Input x, y position of another point: ";
cin >> a >> b;
cout << "The distance is " << psA．distance(a,b) <<endl;
}
```

●试题六

【说明】

下面是一个 Applet 程序，其功能是建立一个图形用户界面的窗口，包括一个文本显示区和一个按钮，点击按钮，可以在文本区已有的文本基础上追加显示 10 条"Welcome to China"信息，并且文本区由滚动条控制文本的上下滚动。

程序运行结果如图 4 所示。

```
import javax.swing.*;
import javA．awt.*;
import javA．awt.event.*;
/*
<applet    code="ex5_6.class" width=800 height=400 >
</applet>
```

图 4

```
*/
public class ex5_6 extends JApplet {
JButton jb = new JButton("Add Text");
JTextPane jtp = new JTextPane();
public void init() {
jb.addActionListener(new ActionListener() {
public void actionPerformed(ActionEvent e){
```

```
for(int i = 1; i < 10; i++)
  (1)  + "Welcome to China! \ n");
}
});
Container cp =  (2)  ;
cp.add(new JScrollPane(jtp));
cp.add(  (3)  );
}
public static void main(String [ ]  args) {
ex5_6 obj5_6=new ex5_6();
String str = obj5_6.getClass().toString();
if(str.indexOf("class") !=-1)
str = str.substring  (6)  ;
JFrame frm = new JFrame(str);
frm.addWindowListener(new  (4)  {
public void windowClosing(WindowEvent we) {
System.exit(0);
}
});
  (5)  .add(ex5_6);
frm.setSize(300, 400);
frm.setVisible(true);
}
}
```

ex5_6.html

```
<HTML>
<HEAD>
<TITLE>ex5_6</TITLE>
</HEAD>
<BODY>
<applet    code="ex5_6.class" width=800 height=400 >
</applet>
</BODY>
</HTML>
```