

## 2016 年上半年软件设计师考试真题

### 上午试题

●VLIW 是 ( 1 ) 的简称。

A . 复杂指令系统计算机

B . 超大规模集成电路

C . 单指令流多数据流

D . 超长指令字

●主存与 Cache 的地址映射方式中, ( 2 ) 方式可以实现主存任意一块装入 Cache 中任意位置, 只有装满才需要替换。

A . 全相联

B . 直接映射

C . 组相联

D . 串并联

●如果 “2X” 的补码是 “90H”, 那么 X 的真值是 ( 3 )。

A . 72

B . -56

C . 56

D . 1111

●移位指令中的 ( 4 ) 指令的操作结果相当于对操作数进行乘 2 操作。

- A . 算术左移
- B . 逻辑右移
- C . 算术右移
- D . 带进位循环左移

●内存按字节编址，从 A1000H 到 B13FFH 的区域的存储容量为 ( 5 ) KB。

- A . 32
- B . 34
- C . 65
- D . 67

●以下关于总线的叙述中，不正确的是 ( 6 )。

- A . 并行总线适合近距离高速数据传输
- B . 串行总线适合长距离数据传输
- C . 单总线结构在一个总线上适应不同种类的设备，设计简单且性能很高
- D . 专用总线在设计上可以与连接设备实现最佳匹配

●以下关于网络层次与主要设备对应关系的叙述中，配对正确的是 ( 7 )。

- A . 网络层——集线器
- B . 数据链路层——网桥

C . 传输层——路由器

D . 会话层——防火墙

●传输经过 SSL 加密的网页所采用的协议是 ( 8 )。

A . HTTP

B . HTTPS

C . S-HTTP

D . HTTP-S

●为了攻击远程主机，通常利用 ( 9 ) 技术检测远程主状态。

A . 病毒查杀

B . 端口扫描

C . QQ 聊天

D . 身份认证

●某软件公司参与开发管理系统软件的程序员张某，辞职到另一公司任职，于是该项目负责人将该管理系统软件上开发者的署名更改为李某 ( 接张某工作 )。该项目负责人的行为 ( 10 ) 。

A . 侵犯了张某开发者身份权 ( 署名权 )

B . 不构成侵权，因为程序员张某不是软件著作权人

C . 只是行使管理者的权利，不构成侵权

D . 不构成侵权 , 因为程序员张某现已不是项目组成员

●美国某公司与中国某企业谈技术合作 , 合同约定使用 1 项美国专利 ( 获得批准并在有效期内 ) , 该项技术未在中国和其他国家申请专利。依照该专利生产的产品 ( 11 ) 需要向美国公司支付这件美国专利的许可使用费。

A . 在中国销售 , 中国企业

B . 如果返销美国 , 中国企业不

C . 在其他国家销售 , 中国企业

D . 在中国销售 , 中国企业不

●以下媒体文件格式中 , ( 12 ) 是视频文件格式。

A . WAV

B . BMP

C . MP3

D . MOV

●以下软件产品中 , 属于图像编辑处理工具的软件是 ( 13 ) 。

A . Powerpoint

B . Photoshop

C . Premiere

D . Acrobat

●使用 150DPI 的扫描分辨率扫描一幅 3×4 英寸的彩色照片，得到原始的 24 位真彩色图像的数据量是 ( 14 ) Byte。

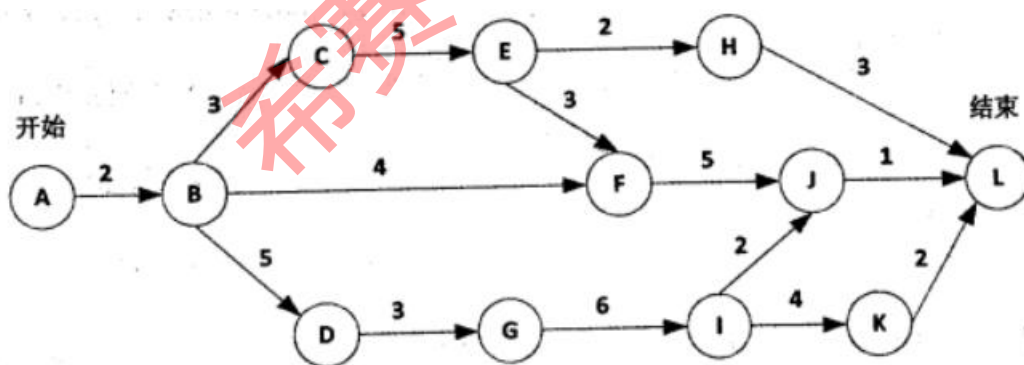
A . 1800

B . 90000

C . 270000

D . 810000

●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，连接顶点的边表示包含的活动，边上的数字表示活动的持续时间(天)，则完成该项目的最少时间为 ( 15 ) 天。活动 BD 最多可以晚开始 ( 16 ) 天而不会影响整个项目的进度。



A . 15

B . 21

C . 22

D . 24

( 16 ) A . 0

B . 2

C . 3

D . 5

●在结构化分析中，用数据流图描述（17）。当采用数据流图对一个图书馆管理系统进行分析时，（18）是一个外部实体。

（17）A . 数据对象之间的关系，用于对数据建模

B . 数据在系统中如何被传送或变换，以及如何对数据流进行变换的功能或子功能，用于对功能建模

C . 系统对外部事件如何响应，如何动作，用于对行为建模

D . 数据流图中的各个组成部分

（18）A . 读者

B . 图书

C . 借书证

D . 借阅

●软件开发过程中，需求分析阶段的输出不包括（19）。

A . 数据流图

B . 实体联系图

C . 数据字典

D . 软件体系结构图

●以下关于高级程序设计语言实现的编译和解释方式的叙述中，正确的是（20）。

- A．编译程序不参与用户程序的运行控制，而解释程序则参与
- B．编译程序可以用高级语言编写，而解释程序只能用汇编语言编写
- C．编译方式处理源程序时不进行优化，而解释方式则进行优化
- D．编译方式不生成源程序的目标程序，而解释方式则生成

●以下关于脚本语言的叙述中，正确的是（21）。

脚本语言是通用的程序设计语言

- B．脚本语言更适合应用在系统级程序开发中
- C．脚本语言主要采用解释方式实现
- D．脚本语言中不能定义函数和调用函数

●将高级语言源程序先转化为一种中间代码是现代编译器的常见处理方式。常用的中间代码有后缀式、（22）树等。

前缀码

三地址码

符号表

补码和移码

●当用户通过键盘或鼠标进入某应用系统时，通常最先获得键盘或鼠标输入信息的是（23）

程序。

- A . 命令解释
- B . 中断处理
- C . 用户登录
- D . 系统调用

●在 Windows 操作系统中，当用户双击“IMG\_20160122\_103.jpg”文件名时，系统会自动通过建立的（24）来决定使用什么程序打开该图像文件。

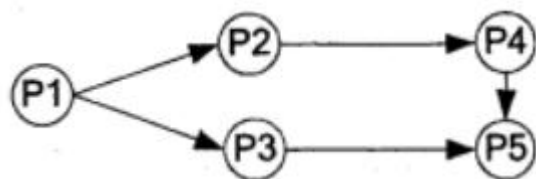
- A . 文件
- B . 文件关联
- C . 文件目录
- D . 临时文件

●某磁盘有 100 个磁道，磁头从一个磁道移至另一个磁道需要 6ms。文件在磁盘上非连续存放，逻辑上相邻数据块的平均距离为 10 个磁道，每块的旋转延迟时间及传输时间分别为 100ms 和 20ms，则读取一个 100 块的文件需要（25）ms。

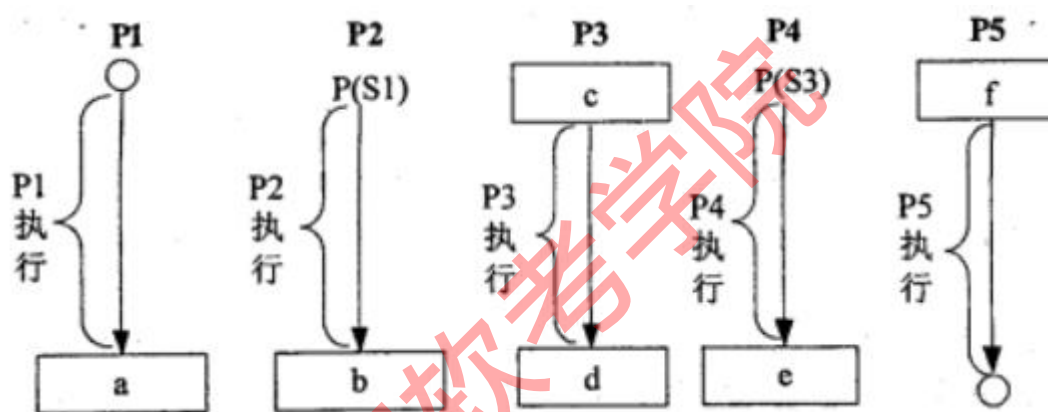
- A . 12060
- B . 12600
- C . 18000
- D . 186000

●进程 P1、P2、P3、P4 和 P5 的前趋图如下图所示：





若用 PV 操作控制进程 P1、P2、P3、P4 和 P5 并发执行的过程，则需要设置 5 个信号 S1、S2、S3、S4 和 S5，且信号量 S1~S5 的初值都等于零。下图中 a 和 b 处应分别填 ( 26 )；c 和 d 处应分别填写 ( 27 )；e 和 f 处应分别填写 ( 28 )。



A .  $V(S1)$   $P(S2)$  和  $V(S3)$

B .  $P(S1)$   $V(S2)$  和  $V(S3)$

C .  $V(S1)$   $V(S2)$  和  $V(S3)$

D .  $P(S1)$   $P(S2)$  和  $V(S3)$

( 27 ) A .  $P(S2)$  和  $P(S4)$

B .  $P(S2)$  和  $V(S4)$

C .  $V(S2)$  和  $P(S4)$

D .  $V(S2)$  和  $V(S4)$

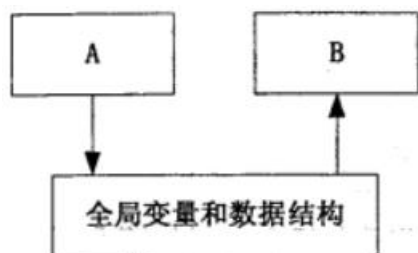
( 28 ) A .  $P(S4)$  和  $V(S4) V(S5)$

B .  $V(S5)$  和  $P(S4) P(S5)$

C .  $V(S3)$  和  $V(S4) V(S5)$

D.  $P(S3)$  和  $P(S4) \vee P(S5)$

●如下图所示，模块 A 和模块 B 都访问相同的全局变量和数据结构，则这两个模块之间的耦合类型为 (29) 耦合。



公共

B. 控制

C. 标记

D. 数据

●以下关于增量开发模型的叙述中，不正确的是 (30)。

A. 不必等到整个系统开发完成就可以使用

B. 可以使用较早的增量构件作为原型，从而获得稍后的增量构件需求

C. 优先级最高的服务先交付，这样最重要的服务接受最多的测试

D. 有利于进行好的模块划分

●在设计软件的模块结构时，(31) 不能改进设计质量。

A. 模块的作用范围应在其控制范围之内

B. 模块的大小适中

C．避免或减少使用病态连接（从中部进入或访问一个模块）

D．模块的功能越单纯越好

●软件体系结构的各种风格中，仓库风格包含一个数据仓库和若干个其他构件。数据仓库位于该体系结构的中心，其他构件访问该数据仓库并对其中的数据进行增、删、改等操作。以下关于该风格的叙述中，不正确的是（32）。（33）不属于仓库风格。

A．支持可更改性和可维护性

B．具有可复用的知识源

C．支持容错性和健壮性

D．测试简单

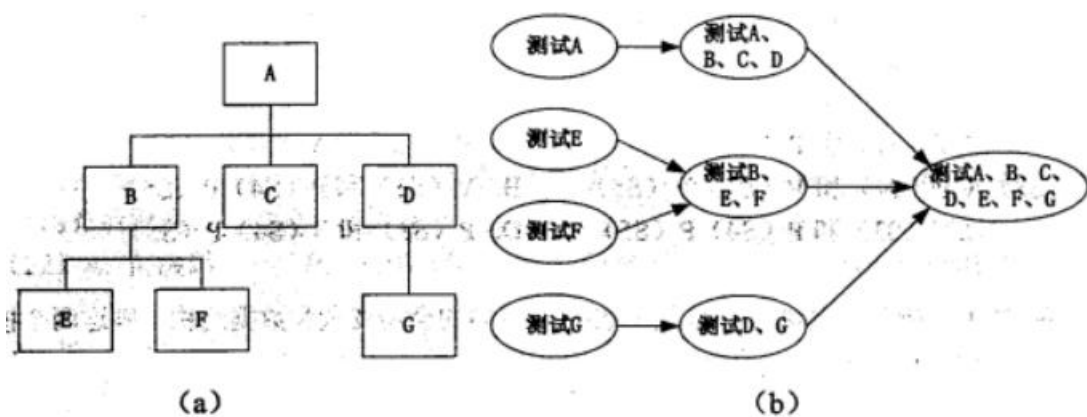
A．数据库系统

B．超文本系统

C．黑板系统

D．编译器

●下图（a）所示为一个模块层次结构的例子，图（b）所示为对其进行集成测试的顺序，则此测试采用了（34）测试策略。该测试策略的优点不包括（35）。



A . 自底向上

B . 自顶向下

C . 三明治

D . 一次性

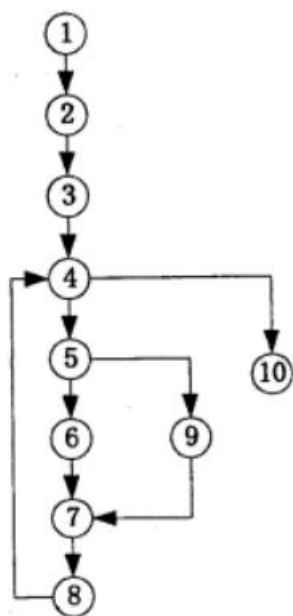
( 35 ) A . 较早地验证了主要的控制和判断点

B . 较早地验证了底层模块

C . 测试的并行程度较高

D . 较少的驱动模块和桩模块的编写工作量

●采用 McCabe 度量法计算下图所示程序的环路复杂性为 ( 36 )。



(36) A . 1

B . 2

C . 3

D . 4

●在面向对象方法中，( 37 ) 是父类和子类之间共享数据和方法的机制。子类在原有父类接口的基础上，用适合于自己要求的实现去置换父类中的相应实现称为 ( 38 ) 。

( 37 ) A . 封装

B . 继承

C . 覆盖

D . 多态

( 38 ) A . 封装

B . 继承

C . 覆盖

D . 多态

●在 UML 用例图中，参与者表示 ( 39 )。

A . 人、硬件或其他系统可以扮演的角色

B . 可以完成多种动作的相同用户

C . 不管角色的实际物理用户

D . 带接口的物理系统或者硬件设计

●UML 中关联是一个结构关系，描述了一组链。两个类之间 ( 40 ) 关联。

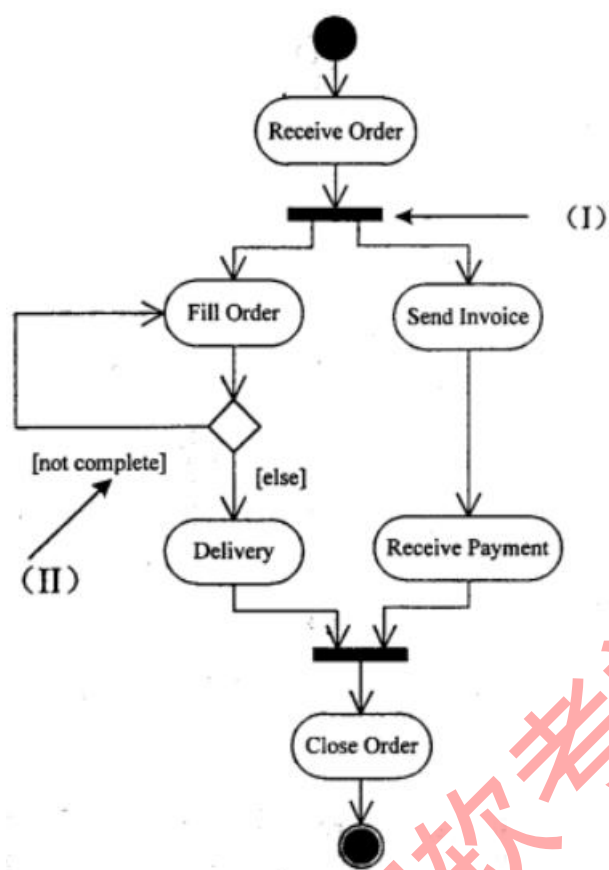
不能有多

可以有多个由不同角色标识的

C . 可以有任意多个

D . 的多个关联必须聚合成一个

●如下所示的 UML 图是 ( 41 )，图中 ( I ) 表示 ( 42 )，( II ) 表示 ( 43 )。



(41) A. 序列图

B. 状态图

C. 通信图

D. 活动图

(42) A. 合并分叉

B. 分支

C. 合并汇合

D. 流

(43) A. 分支条件

B. 监护表达式

C. 动作名

D . 流名称

●为图形用户界面 ( GUI ) 组件定义不同平台的并行类层次结构, 适合采用 ( 44 ) 模式。

A . 享元 ( Flyweight )

B . 抽象工厂 ( Abstract Factory )

C . 外观 ( Facade )

D . 装饰器 ( Decorator )

● ( 45 ) 设计模式将一个请求封装为一个对象, 从而使得可以用不同的请求对客户进行参数化, 对请求排队或记录请求日志, 以及支持可撤销的操作。

命令 ( Command )

B . 责任链 ( Chain of Responsibility )

C . 观察者 ( Observer )

D . 策略 ( Strategy )

● ( 46 ) 设计模式最适合用于发布/订阅消息模型, 即当订阅者注册一个主题后, 此主题有新消息到来时订阅者就会收到通知。

适配器 ( Adapter )

B . 通知 ( Notifier )

C . 观察者 ( Observer )

D . 状态 ( State )



- 因使用大量的对象而造成很大的存储开销时，适合采用（47）模式进行对象共享，以减少对象数量从而达到较少的内存占用并提升性能。

组合（Composite）

B．享元（Flyweight）

C．迭代器（Iterator）

D．备忘（Memento）

- 移进--归约分析法是编译程序（或解释程序）对高级语言源程序进行语法分析的一种方法，属于（48）的语法分析方法。

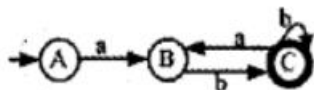
自顶向下（或自上而下）

B．自底向上（或自下而上）

C．自左向右

D．自右向左

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



A．aabb

B．abab

C．baba

D . Abba

- 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 reference ) 方式，main 函数中 “print(x)” 执行后输出的值为 ( 50 )。

main( )	f(int x, int &a)
<pre>int x = 1; f(5, x); print(x);</pre>	<pre>x = 2*x + 1 ; a = a + x; return;</pre>

- A . 1
- B . 6
- C . 11
- D . 12

- 数据的物理独立性和逻辑独立性分别是通过修改 ( 51 ) 来完成的。

- A . 外模式与内模式之间的映像、模式与内模式之间的映像
- B . 外模式与内模式之间的映像、外模式与模式之间的映像
- C . 外模式与模式之间的映像、模式与内模式之间的映像
- D . 模式与内模式之间的映像、外模式与模式之间的映像

- 关系规范化在数据库设计的 ( 52 ) 阶段进行。

需求分析

B . 概念设计

C . 逻辑设计

D . 物理设计

●若给定的关系模式为  $R\langle U, F \rangle$  ,  $U = \{A, B, C\}$  ,  $F = \{AB \rightarrow C, C \rightarrow B\}$  , 则关系 R ( 53 ) 。

有 2 个候选关键字 AC 和 BC , 并且有 3 个主属性

B . 有 2 个候选关键字 AC 和 AB , 并且有 3 个属性

C . 只有一个候选关键字 AC , 并且有 1 个非主属性和 2 个主属性

D . 只有一个候选关键字 AB , 并且有 1 个非主属性和 2 个主属性

●某公司数据库中的元件关系模式为 P ( 元件号 , 元件名称 , 供应商 , 供应商所在地 , 库存量 ) , 函数依赖集 F 如下所示 :

$F = \{\text{元件号} \rightarrow \text{元件名称}, (\text{元件号}, \text{供应商}) \rightarrow \text{库存量}, \text{供应商} \rightarrow \text{供应商所在地}\}$

元件关系的主键为 ( 54 ) , 该关系存在冗余以及插入异常和删除异常等问题。为了解决这一问题需要将元件关系分解 ( 55 ) , 分解后的关系模式可以达到 ( 56 ) 。

A . 元件号 , 元件名称

B . 元件号 , 供应商

C . 元件号 , 供应商所在地

D . 供应商 , 供应商所在地

( 55 ) A . 元件 1 ( 元件号 , 元件名称 , 库存量 ) 元件 2 ( 供应商 , 供应商所在地 )

B . 元件 1 ( 元件号 , 元件名称 ) 元件 2 ( 供应商 , 供应商所在地 , 库存量 )

C. 元件 1 ( 元件号, 元件名称 ) 元件 2 ( 元件号, 供应商, 库存量 ) 元件 3 ( 供应商, 供应商所在地 )

D. 元件 1 ( 元件号, 元件名称 ) 元件 2 ( 元件号, 库存量 ) 元件 3 ( 供应商, 供应商所在地 ) 元件 4 ( 供应商所在地, 库存量 )

( 56 ) A . 1NF

B . 2NF

C . 3NF

D . 4NF

●若元素以 a , b , c , d , 的顺序进入一个初始为空的栈中 , 每个元素进栈、出栈各 1 次 , 要求出栈的第一个元素为 d , 则合法的出栈序列共有 ( 57 ) 种。

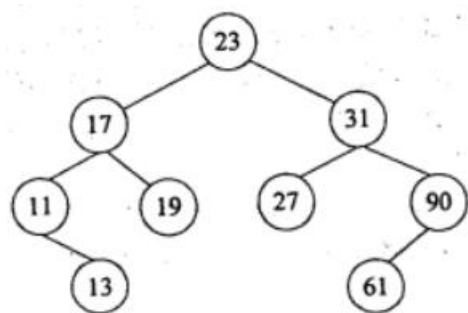
A . 4

B . 5

C . 6

D . 24

●设有二叉排序树( 或二叉查找树 )如下图所示 , 建立该二叉树的关键码序列不可能是( 58 )。



A . 23 31 17 19 11 27 13 90 61

B . 23 17 19 31 27 90 61 11 13

C . 23 17 27 19 31 13 11 90 61

D . 23 31 90 61 27 17 19 11 13

●若一棵二叉树的高度 ( 即层数 ) 为  $h$  , 则该二叉树 ( 59 )。

有  $2h$  个结点

B . 有  $2h-1$  个结点

C . 最少有  $2h-1$  个结点

D . 最多有  $2h-1$  个结点

●在 13 个元素构成的有序表  $A[1..13]$  中进行折半查找 ( 或称为二分查找 , 向下取整 )。那么以下叙述中 , 错误的是 ( 60 )。

无论要查找哪个元素 , 都是先与  $A[7]$  进行比较

B . 若要查找的元素等于  $A[9]$  , 则分别需与  $A[7]$ 、 $A[11]$ 、 $A[9]$  进行比较

C . 无论要查找的元素是否在  $A[]$  中 , 最多与表中的 4 个元素比较即可

D . 若待查找的元素不在  $A[]$  中 , 最少需要与表中的 3 个元素进行比较

●以下关于图的遍历的叙述中 , 正确的是 ( 61 )。

A . 图的遍历是从给定的源点出发对每一个顶点仅访问一次的过程

B . 图的深度优先遍历方法不适用于无向图

C . 使用队列对图进行广度优先遍历

D . 图中有回路时则无法进行遍历

●考虑一个背包问题，共有  $n=5$  个物品，背包容量为  $W=10$ ，物品的重量和价值分别为：

$w=\{2, 2, 6, 5, 4\}$ ， $v=\{6, 3, 5, 4, 6\}$ ，求背包问题的最大装包价值。若此为 0-1 背包

问题，分析该问题具有最优子结构，定义递归式为

$$c(i, j) = \begin{cases} 0 & \text{若 } i = 0 \text{ 或 } j = 0 \\ c(i-1, j) & \text{若 } w[i] > j \\ \max\{c(i-1, j), c(i-1, j-w(i))\} & \text{其它} \end{cases}$$

其中  $c(i, j)$  表示  $i$  个物品、容量为  $j$  的 0-1 背包问题的最大装包价值，最终要求解  $c(n, W)$ 。

采用自底向上的动态规划方法求解，得到最大装包价值为 ( 62 )，算法的时间复杂度为 ( 63 )。

若此为部分背包问题，首先采用归并排序算法，根据物品的单位重量价值从大到小排序，然后依次将物品放入背包直至所有物品放入背包中或者背包再无容量，则得到的最大装包价值为 ( 64 )，算法的时间复杂度为 ( 65 )。

( 62 ) A . 11

B . 14

C . 15

D . 16.67

( 63 ) A .  $\Theta(nW)$

B .  $\Theta(n \lg n)$

C .  $\Theta(n^2)$

D .  $\Theta(\lg nW)$

( 64 ) A . 11

B . 14

C . 15

D . 16.67

( 65 ) A .  $\Theta(nW)$

B .  $\Theta(\lg n)$

C .  $\Theta(n^2)$

D .  $\Theta(\lg nW)$

●默认情况下，FTP 服务器的控制端口为 ( 66 )，上传文件时的端口为 ( 67 )。

( 66 ) A . 大于 1024 的端口

B . 20

C . 80

D . 21

( 67 ) A . 大于 1024 的端口

B . 20

C . 80

D . 21

●使用 ping 命令可以进行网络检测，在进行一系列检测时，按照由近及远原则，首先执行的

是 ( 68 )。

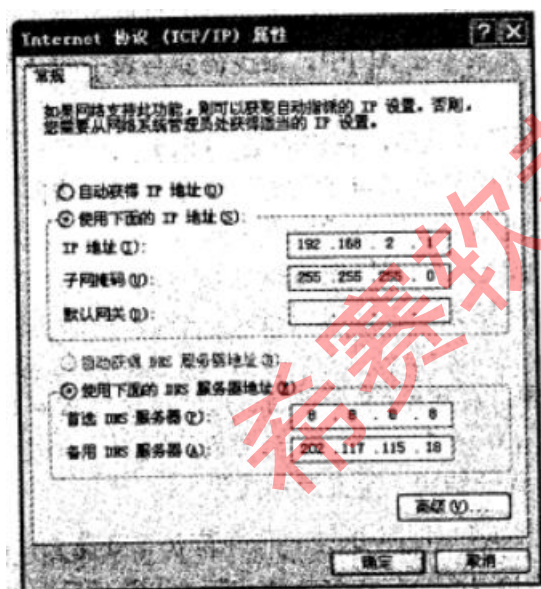
A . ping 默认网关

B . ping 本地 IP

C . ping127.0.0.1

D . ping 远程主机

●某 PC 的 Internet 协议属性参数如下图所示，默认网关的 IP 地址是 ( 69 )。



A . 8.8.8.8

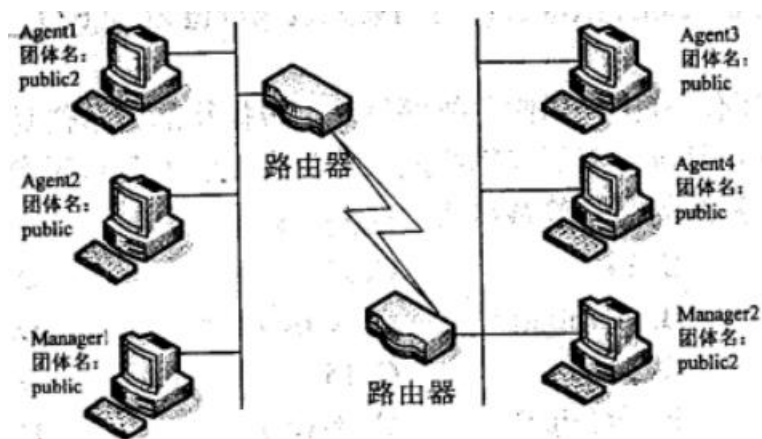
B . 202.117.115.3

C . 192.168.2.254

D . 202.117.115.18

●在下图的 SNMP 配置中，能够响应 Manager2 的 getRequest 请求的是 ( 70 )。





- A . Agent1
- B . Agent2
- C . Agent3
- D . Agent4

● In the fields of physical security and information security, access control is the selective restriction of access to a place or other resource. The act of accessing may mean consuming, entering, or using. Permission to access a resource is called authorization (授权).

An access control mechanism ( 71 ) between a user (or a process executing on behalf of a user) and system resources, such as applications, operating systems, firewalls, routers, files, and databases. The system must first authenticate (验证) a user seeking access. Typically the authentication function determines whether the user is ( 72 ) to access the system at all. Then the access control function determines if the specific requested access by this user is permitted. A security administrator maintains an authorization database that specifies what type of

access to which resources is allowed for this user. The access control function consults this database to determine whether to ( 73 ) access. An auditing function monitors and keeps a record of user accesses to system resources.

In practice, a number of ( 74 ) may cooperatively share the access control function.

All operating systems have at least a rudimentary ( 基本的 ) , and in many cases a quite robust, access control component. Add-on security packages can add to the ( 75 ) access control capabilities of the OS. Particular applications or utilities, such as a database management system, also incorporate access control functions. External devices, such as firewalls, can also provide access control services.

( 71 ) A . cooperates

B . coordinates

C . connects

D . mediates

( 72 ) A . denied

B . permitted

C . prohibited

D . rejected

( 73 ) A . open

B . monitor

C . grant

D . seek

( 74 ) A . components

B . users

C . mechanisms

D . algorithms

( 75 ) A . remote

B . native

C . controlled

D . Automated

### 下午试题

试题一 ( 共 15 分 )

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】

某会议中心提供举办会议的场地设施和各种设备，供公司与各类组织机构租用。场地包括一个大型报告厅、一个小型报告厅以及诸多会议室。这些报告厅和会议室可提供的设备有投影仪、白板、视频播放/回放设备、计算机等。为了加强管理，该中心欲开发一会议预订系统，系统的主要功能如下。

( 1 ) 检查可用性。客户提交预订请求后，检查预订表，判定所申请的场地是否在申请日期内可用；如果不可用，返回不可用信息。

( 2 ) 临时预订。会议中心管理员收到客户预定请求的通知之后，提交确认。系统生成新临时预订存入预订表，并对新客户创建一条客户信息记录加以保存。根据客户记录给客户

发送临时预订确认信息和支付定金要求。

(3) 分配设施与设备。根据临时预订或变更预定的设备和设施需求,分配所需设备(均能满足用户要求)和设施,更新相应的表和预订表。

(4) 确认预订。管理员收到客户支付定金的通知后,检查确认,更新预订表,根据客户记录给客户发送预订确认信息。

(5) 变更预订。客户还可以在支付余款前提交变更预订请求,对变更的预订请求检查可用性,如果可用,分配设施和设各;如果不可用,返回不可用信息。管理员确认变更后,根据客户记录给客户发送确认信息。

(6) 要求付款。管理员从预订表中查询距预订的会议时间两周内的预定,根据客户记录给满足条件的客户发送支付余款要求。

(7) 支付余款。管理员收到客户余款支付的通知后,检查确认,更新预订表中的已支付余款信息。

现采用结构化方法对会议预定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(不完整)。

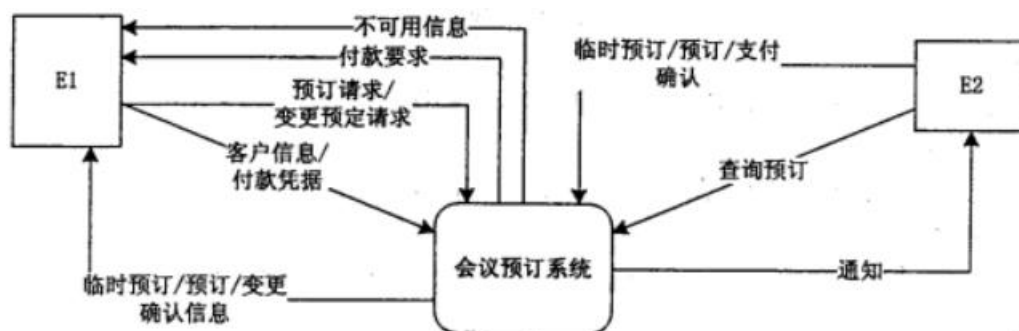
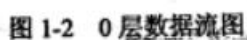


图 1-1 上下文数据流图



使用说明中的词语，给出图 1-1 中的实体 E1 ~ E2 的名称。

使用说明中的词语，给出图 1-2 中的数据存储 D1~D4 的名称。

根据说明和图中术语，补充图 1-2 之中缺失的数据流及其起点和终点。

【问题 4】(3 分)

如果发送给客户的确认信息是通过 Email 系统向客户信息中的电子邮件地址进行发送的，那么需要对图 1-1 和 1-2 进行哪些修改？用 150 字以内文字加以说明。

## 试题二（共 15 分）

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】

某销售公司当前的销售业务为商城实体店销售。现该公司拟开展网络销售业务，需要开发一个信息化管理系统。请根据公司现有业务及需求完成该系统的数据库设计。

### 【需求描述】

（1）记录公司所有员工的信息。员工信息包括工号、身份证号、姓名、性别、出生日期和电话，并只登记一部电话。

（2）记录所有商品的信息。商品信息包括商品名称、生产厂家、销售价格和商品介绍。系统内部用商品条码唯一区别每种商品。

（3）记录所有顾客的信息。顾客信息包括顾客姓名、身份证号、登录名、登录密码、和电话号码。一位顾客只能提供一个电话号码。系统自动生成唯一的顾客编号。

（4）顾客登录系统之后，在网上商城购买商品。顾客可将选购的商品置入虚拟的购物车内，购物车可长期存放顾客选购的所有商品。顾客可在购物车内选择商品、修改商品数量后生成网购订单。订单生成后，由顾客选择系统提供的备选第三方支付平台进行电子支付，支付成功后系统需要记录唯一的支付凭证编号，然后由商城根据订单进行线下配送。

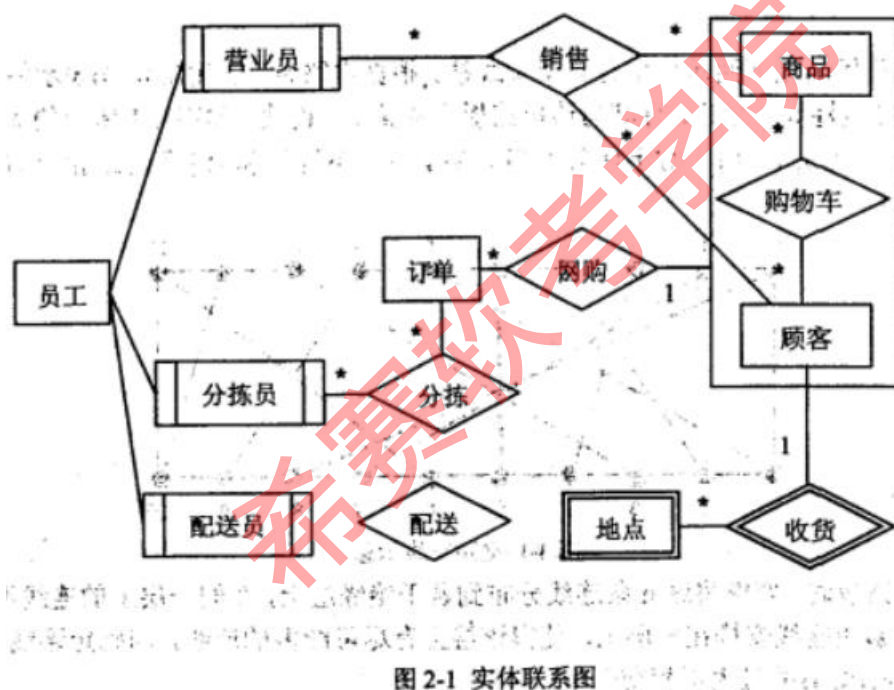
（5）所有的配送商品均由仓库统一出库。为方便顾客，允许每位顾客在系统中提供多组收货地址、收货人及联系电话。一份订单所含的多个商品可能由多名分检员根据商品所在

仓库信息从仓库中进行分拣操作,分拣后的商品交由配送员根据配送单上的收货地址进行配送。

(6) 新设计的系统要求记录实体店的每笔销售信息,包括营业员、顾客、所售商品及其数量。

### 【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。



### 【逻辑结构设计】

根据概念模型设计阶段完成的实体联系图,得出如下关系模式(不完整):

员工(工号,身份证号,姓名,性别,出生日期,电话)

商品(商品条码,商品名称,生产厂家,销售价格,商品介绍, (a) )

顾客(顾客编号,姓名,身份证号,登录名,登录密码,电话)

收货地点(收货 ID,顾客编号,收货地址,收货人,联系电话)

购物车(顾客编号,商品条码,商品数量)



订单（订单 ID，顾客编号，商品条码，商品数量，（b））

分拣（分拣 ID，分拣员工号，（c），分拣时间）

配送（配送 ID，分拣 ID，配送员工号，收货 ID，配送时间，签收时间，签收快照）

销售（销售 ID，营业员员工号，顾客编号，商品条码，商品数量）

【问题 1】（4 分）

补充图 2-1 中的“配送”联系所关联的对象及联系类型。

【问题 2】（6 分）

补充逻辑结构设计中的（a）（b）和（c）三处空缺。

【问题 3】（5 分）

对于实体店销售，若要增加送货上门服务，由营业员在系统中下订单，与网购的订单进行后续的统一管理。请根据该需求，对图 2-1 进行补充，并修改订单关系模式。

试题三（共 15 分）

阅读下列说明和图，回答问题 1 至问题 3，将解答填入答题纸的对应栏内。

【说明】

某软件公司欲设计实现一个虚拟世界仿真系统。系统中的虚拟世界用于模拟现实世界中的不同环境（由用户设置并创建），用户通过操作仿真系统中的 1~2 个机器人来探索虚拟世界。机器人维护着两个变量 b1 和 b2，用来保存从虚拟世界中读取的字符。

该系统的主要功能描述如下：

（1）机器人探索虚拟世界（Run Robots）。用户使用编辑器（Editor）编写文件以设置想要模拟的环境，将文件导入系统（Load File）从而在仿真系统中建立虚拟世界（Setup



World)。机器人在虚拟世界中的行为也在文件中进行定义，建立机器人的探索行为程序（Setup Program）。机器人在虚拟世界中探索时（Run Program），有 2 种运行模式：

①自动控制（Run）：事先编排好机器人的动作序列（指令（Instruction）），执行指令，使机器人可以连续动作。若干条指令构成机器人的指令集（Instruction Set）。

②单步控制（Step）：自动控制方式的一种特殊形式，只执行指定指令中的一个动作。

（2）手动控制机器人（Manipulate Robots）。选定 1 个机器人后（Select Robot），可以采用手动方式控制它。手动控制有 4 种方式：

①Move：机器人朝着正前方移动一个交叉点。

②Left：机器人原地沿逆时针方向旋转 90 度。

③Read：机器人读取其所在位置的字符，并将这个字符的值赋给 b1；如果这个位置上没有字符，则不改变 b1 的当前值。

④Write：将 b1 中的字符写入机器人当前所在的位置，如果这个位置上已经有字符，该字符的值将会被 b1 的值替代。如果这时 b1 没有值，即在执行 Write 动作之前没有执行过任何 Read 动作，那么需要提示用户相应的错误信息（Show Errors）。

手动控制与单步控制的区别在于，单步控制时执行的是指令中的动作，只有一种控制方式，即执行下个动作；而手动控制时有 4 种动作。

现采用面向对象方法设计并实现该仿真系统，得到如图 3-1 所示的用例图和图 3-2 所示的初始类图。图 3-2 中的类“Interpreter”和“Parser”用于解析描述虚拟世界的文件以及机器人行为文件中的指令集。

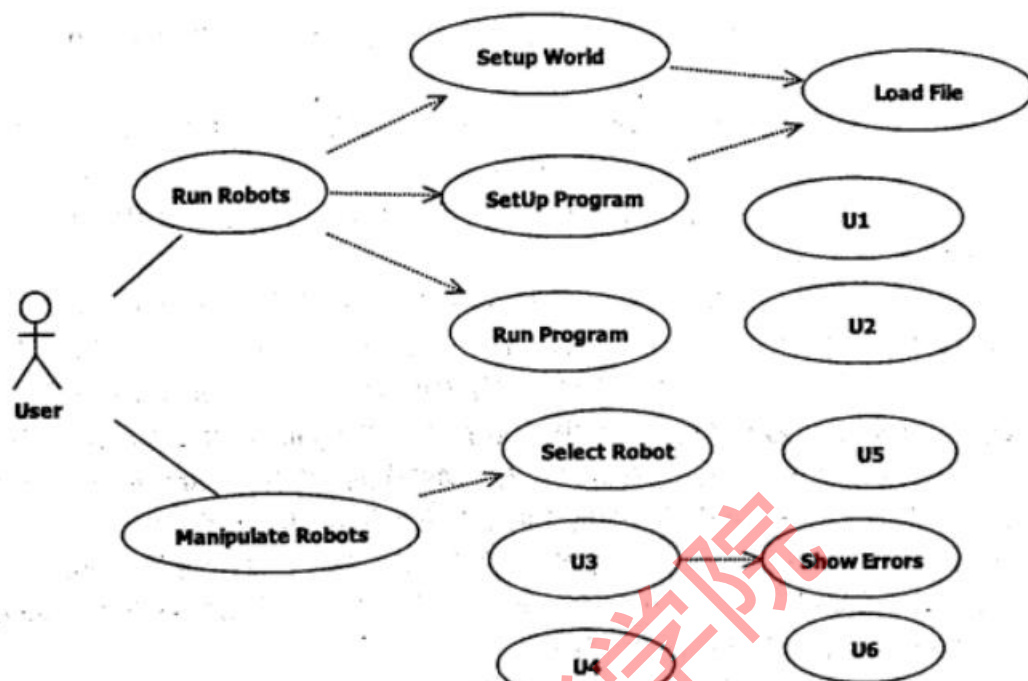


图 3-1 用例图

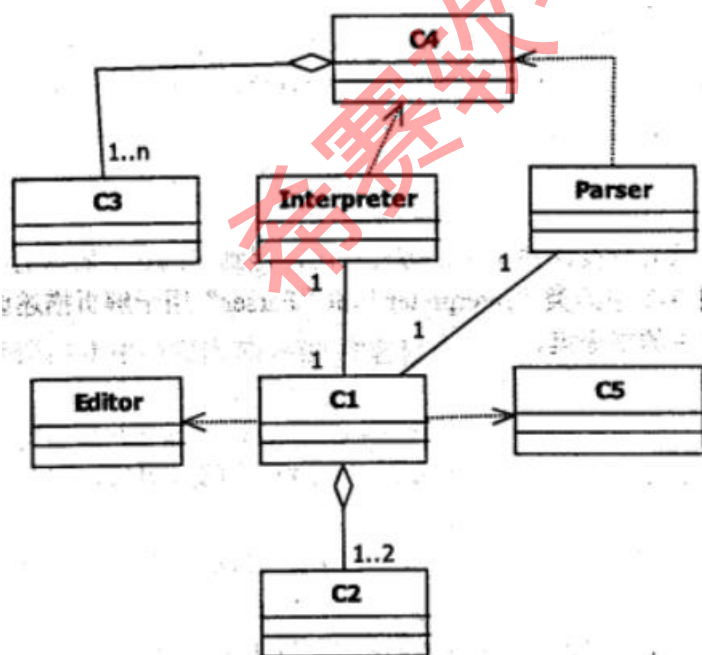


图 3-2 初始类图

【问题 1】(6 分)

根据说明中的描述，给出图 3-1 中 U1~U6 所对应的用例名。

【问题 2】(4 分)

图 3-1 中用例 U1 ~ U6 分别与哪个 ( 哪些 ) 用例之间有关系 , 是何种关系 ?

【问题 3】( 5 分 )

根据说明中的描述 , 给出图 3-2 中 C1 ~ C5 所对应的类名。

试题四 ( 共 15 分 )

阅读下列说明和 C 代码 , 回答问题 1 至问题 3 , 将解答写在答题纸的对应栏内。

【说明】

在一块电路板的上下两端分别有  $n$  个接线柱。根据电路设计 , 用  $(i, \pi(i))$  表示将上端接线柱  $i$  与下端接线柱  $\pi(i)$  相连 , 称其为该电路板上的第  $i$  条连线。如图 4-1 所示的  $\pi(i)$  排列为  $\{8, 7, 4, 2, 5, 1, 9, 3, 10, 6\}$ 。对于任何  $1 \leq i < j \leq n$  , 第  $i$  条连线和第  $j$  条连线相交的充要条件是  $\pi(i) > \pi(j)$ 。

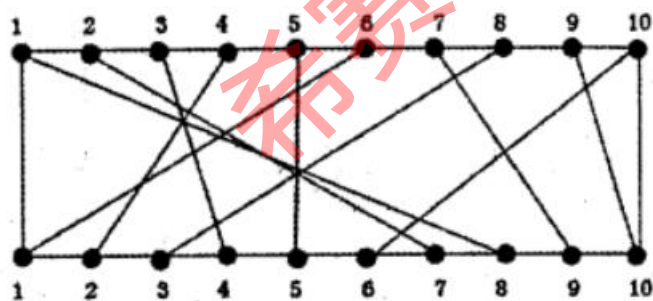


图 4-1 电路布线示意

在制作电路板时 , 要求将这  $n$  条连线分布到若干绝缘层上 , 在同一层上的连线不相交。

现在要确定将哪些连线安排在一层上 , 使得该层上有尽可能多的连线 , 即确定连线集

$Nets = \{(i, \pi(i)) , 1 \leq i \leq n\}$  的最大不相交子集。

【分析问题】

记  $N(i, j) = \{(t, \pi(t)) \in Nets, t \leq i, \pi(t) \leq j\}$ 。  $N(i, j)$  的最大不相交子集为  $MNS(i, j)$  ,

$size(i, j) = |MNS(i, j)|$ 。

经分析，该问题具有最优子结构性质。对规模为  $n$  的电路布线问题，可以构造如下递归式：

$$(1) \text{ 当 } i=1 \text{ 时, } size(1, j) = \begin{cases} 0 & j < \pi(1) \\ 1 & \text{其它情况} \end{cases}$$
$$(2) \text{ 当 } i > 1 \text{ 时, } size(i, j) = \begin{cases} size(i-1, j) & j < \pi(i) \\ \max\{size(i-1, j), size(i-1, \pi(i)-1) + 1\} & \text{其它情况} \end{cases}$$

### 【C 代码】

下面是算法的 C 语言实现。

#### (1) 变量说明

$size[i][j]$ ：上下端分别有  $i$  个和  $j$  个接线柱的电路板的第一层最大不相交连接数

$pi[i]$ ： $\pi(i)$ ，下标从 1 开始

#### (2) C 程序

```
#include "stdlib.h"
```

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

```
#define N 10 /*问题规模*/
```

```
Int m=0; /*牢记最大连接集合中的接线柱*/
```

```
Void maxNum(int pi[],int size[N+1][N+1],int n) /*求最大不相交连接数*/
```

```
int i, j;
```

```
for(j=0; j < pi[1]; j++) size[1][j] = 0; /*当  $j < \pi(1)$  时 */
```

```
for(j=pi[1]; j <= n; j++) (1) ; /*当  $j \geq \pi(1)$  时 */
```

```
for(i=2; i < n; i++) {
```

```
for(j=0; j < pi[i]; j++) (2) ; /*当  $j < \pi(i)$  时 */
```

```
for(j=pi[i];j<=n ; j++) { /*当 j>=c[i]时,考虑两种情况*/

    size[i][j]=size[i-1][j]>=size[i-1][pi[i]-1]+1 ?size[i-1][j]:size[i-1][pi[i]-1]+1;

}

}

/*最大连接数 */

size[n][n]=size[n-1][n]>=size[n-1][pi[n]-1]+1 ? size[n-1][n]:size[n-1][pi[n]-1]+1;

}

/*构造最大不相交连接集合，net[i]表示最大不相交子集中第 i 条连线的上端接线柱的序号

*/

void constructSet ( int pi[],int size[N+1][N+1],int n,int net[n] ) {

    int i,j=n;

    m=0 ;

    for(i=n ; i>1;i--) { /*从后往前*/

        if(size[i][j]!=size[i-1][j]){ /*(i,pi[i])是最大不相交子集的一条连线*/

            ( 3 ) ; /*将 i 记录到数组 net 中，连接线数自增 1*/

            j= pi[i]-1; /*更新扩展连线柱区间*/

        }

    }

    if(j>=pi[1]) net[m++]=1; /*当 i=1 时*/

}
```

【问题 1】( 6 分 )

根据以上说明和 C 代码，填充 C 代码中的空 (1) ~ (3)。

【问题 2】(6 分)

根据题干说明和以上 C 代码，算法采用了 (4) 算法设计策略。

函数 maxNum 和 constructSet 的时间复杂度分别为 (5) 和 (6) (用 O 表示)。

【问题 3】(3 分)

若连接排列为 {8,7,4,2,5,1,9,3,10,6}，即如图 4-1 所示，则最大不相交连接数为 (7)，包含的连线为 (8) (用  $(i, \pi(i))$  的形式给出)。

试题五 (共 15 分)

阅读下列说明和 C++ 代码，将应填入 (n) 处的字句写在答题纸的对应栏内。

【说明】

某软件系统中，已设计并实现了用于显示地址信息的类 Address (如图 5-1 所示)，现要求提供基于 Dutch 语言的地址信息显示接口。为了实现该要求并考虑到以后可能还会出现新的语言的接口，决定采用适配器 (Adapter) 模式实现该要求，得到如图 5-1 所示的类图。

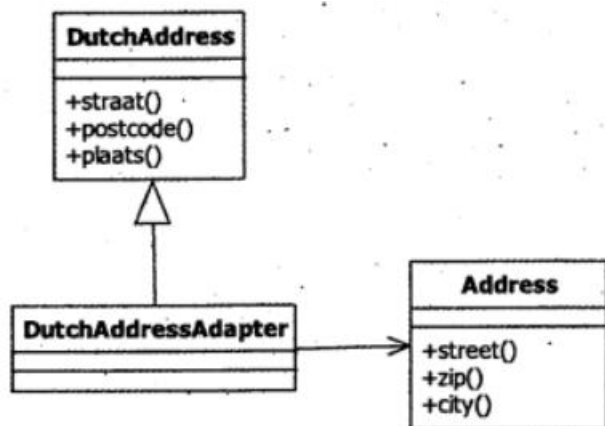


图 5-1 适配器模式类图

**【C++代码】**

```
#include <iostream>
```

```
using namespace std;
```

```
class Address{
```

```
public:
```

```
    void stree()    { /* 实现代码省略 */ }
```

```
    void zip()      { /* 实现代码省略 */ }
```

```
    void city()     { /* 实现代码省略 */ }
```

```
    //其他成员省略
```

```
};
```

```
class DutchAddress {
```

```
public:
```

```
    virtual void straat()=0;
```

```
    virtual void postcode()=0;
```

```
    virtual void plaats()=0;
```

```
    //其他成员省略
```

```
};
```

```
class DutchAddressAdapter : public DutchAddress {
```

private:

( 1 ) ;

public:

DutchAddressAdapter(Address \*addr) {

    address = addr;

}

void straat() {

    ( 2 ) ;

}

void postcode(){

    ( 3 ) ;

}

void plaat(){

    ( 4 ) ;

}

//其他成员省略

};

void testDutch(DutchAddress \*addr){

    addr->straat();

    addr->postcode();



```
        addr->plaats();  
  
    }  
  
int main(){  
  
    Address*addr = new Address();  
  
    ( 5 )    ;  
  
    Cout<< "\n The DutchAddress\n"<< endl;  
  
    testDutch(addrAdapter);  
  
    return 0;  
  
}
```

#### 试题六 ( 共 15 分 )

阅读下列说明和 Java 代码，将应填入 ( n ) 处的字句写在答题纸的对应栏内。

##### 【说明】

某软件系统中，已设计并实现了用于显示地址信息的类 Address ( 如图 6-1 所示 )，现要求提供基于 Dutch 语言的地址信息显示接口。为了实现该要求并考虑到以后可能还会出现新的语言的接口，决定采用适配器 ( Adapter ) 模式实现该要求，得到如图 6-1 所示的类图。

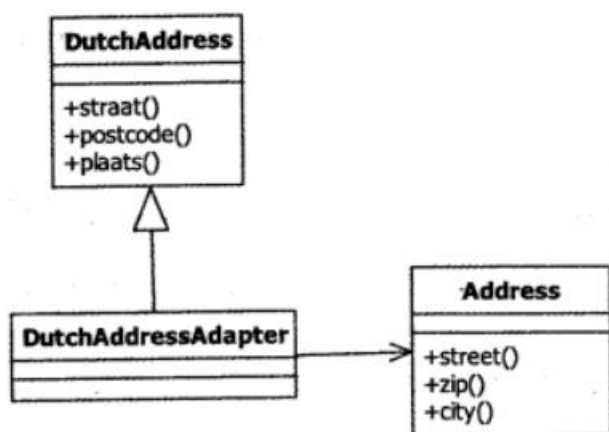


图 6-1 适配器模式类图

【Java 代码】

```
import java.util.*;
```

```
Class Address{
```

```
    public void street() { //实现代码省略 }
```

```
    public void zip() { //实现代码省略 }
```

```
    public void city() { //实现代码省略 }
```

```
    //其他成员省略
```

```
}
```

```
class DutchAddress{
```

```
    public void straat() { //实现代码省略 }
```

```
    public void postcode() { //实现代码省略 }
```

```
    public void plaats() { //实现代码省略 }
```

```
    //其他成员省略
```

```
}
```

```
class DutchAddressAdapter extends DutchAddress {
```

```
    private    ( 1 )    ;
```

```
    public DutchAddressAdapter (Address addr){
```

```
        address= addr;
```

```
    }
```

```
    public void straat() {
```

```
        ( 2 )    ;
```

```
    }
```

```
    public void postcode() {
```

```
        ( 3 )    ;
```

```
    }
```

```
    public void plaats(){
```

```
        ( 4 )    ;
```

```
    }
```

```
//其他成员省略
```

```
}
```

```
class Test {
```

```
    public static void main(String[] args) {
```

```
        Address addr= new Address();
```

```
        ( 5 )    ;
```

```
        System.out.println("\n The DutchAddress\n");
```

```
        testDutch(addrAdapter);
```

```
    }
```

```
    Static void  testDutch(DutchAddressaddr){
```

```
        addr.straat();
```

```
        addr.postcode();
```

```
        Addr.plaats();
```

```
    }
```

```
}
```