CPU 在执行指令的过程中，会自动修改（1）的内容，以使其保存的总是将要执行的下一条指令的地址。

（1）A.指令寄存器　　B.程序计数器　　C.地址寄存器　　D.指令译码器

【答案】B

【解析】

CPU 执行指令的过程中，会自动修改 PC 的内容，PC 是指令计数器，用来存放将要执行的下一条指令。

对于指令寄存器（IR）存放即将执行的指令，指令译码器（ID）对指令中的操作码字段进行分析和解释，地址寄存器（AR），不是我们常用的 CPU 内部部件，其作用是是用来保存当前 CPU 所要访问的内存单元或 I/O 设备的地址。

在微机系统中，BIOS（基本输入输出系统）保存在（2）中。

（2）A.主板上的 ROM　　B.CPU 的寄存器　　C.主板上的 RAM　D.虚拟存储器

【答案】A

【解析】

BIOS（Basic Input Output System）（基本输入输出系统）是一组固化到计算机内主板上一个 ROM 芯片上的程序，它保存着计算机最重要的基本输入输出的程序、开机后自检程序和系统自启动程序，它可从 CMOS 中读写系统设置的具体信息。

采用 n 位补码（包含一个符号位）表示数据，可以直接表示数值（3）。

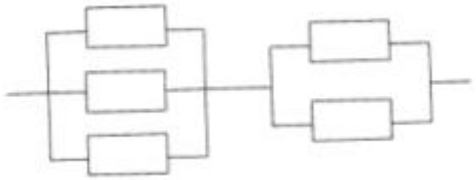
（3）A.2^n　　B.-2^n　　C.2^{n-1}　　D. -2^{n-1}

【答案】D

【解析】

在计算机中，n 位补码（表示数据位），表示范围是 -2^{n-1} ~ $-2^{n-1}-1$，其中最小值为认为定义，以 n=8 为例，其中-128 的补码是人为定义的 1000 0000。

某系统由下图所示的部件构成，每个部件的千小时可靠度都为 R，该系统的千小时可靠度为（4）。



（4）A.（3R+2R）/2　　B. R/3+R/2

C.（1-（1-R）^{3}）（1-（1-R）^{2}）　　D.（1-（1-R）^{3}-（1-R）^{2}）

【答案】C

【解析】

对于可靠度计算，串联系统可靠度为 R1*R2，并联系统 R_1=1-(1-R)*(1-R)*(1-R)，并联系统 R_2=1-(1-R)*(1-R)，因此答案为（1-（1-R）^{3}）（1-（1-R）^{2}）。

以下关于采用一位奇校验方法的叙述中，正确的是（5）。

（5）A.若所有奇数位出错，则可以检测出该错误但无法纠正错误

B.若所有偶数位出错，则可以检测出该错误并加以纠正

C.若有奇数个数据位出错，则可以检测出该错误但无法纠正错误

D.若有偶数个数据位出错，则可以检测出该错误并加以纠正

【答案】C

【解析】

对于奇偶校验，是由若干位有效信息，再加上一个二进制位（校验位）组成校验码，其中奇校验“1”的个数为奇数，而偶校验“1”的个数为偶数，以此完整校验，如果其中传输过程中有偶数个数发生错误（即 1 变成 0 或 0 变成 1），则“1”的个数，其奇偶就不会发生改变，也就无法发现错误了，只有奇数个数据位发生错误，才能发现错误。同时，奇偶校验只能查错不能纠错。

下列关于流水线方式执行指令的叙述中，不正确的是（6）。

（6）A.流水线方式可提高单条指令的执行速度

B.流水线方式下可同时执行多条指令

C.流水线方式提高了各部件的利用率

D.流水线方式提高了系统的吞吐率

【答案】A

【解析】本题要求选择不正确的叙述。

其中 A 流水线方式可提高单条指令的执行速度是不正确的，对于只有单条指令的情况下，流水线方式与顺序执行时没有区别的。流水线的原理是在某一时刻可以让多个部件同时处理多条指令，避免各部件等待空闲，由此提高了各部件的利用率，也提高了系统的吞吐率。

DES 是（7）算法。

（7）A.公开密钥加密　　B.共享密钥加密　　C.数字签名　　D.认证

【答案】B

【解析】

非对称加密又称为公开密钥加密，而共享密钥加密指对称加密。常见的对称加密算法有：DES，三重 DES、RC-5、IDEA、AES。

计算机病毒的特征不包括（8）。

（8）A.传染性　　B.触发性　　C.隐蔽性　　D.自毁性

【答案】D

【解析】

计算机病毒具有隐蔽性、传染性、潜伏性、触发性和破坏性等特定。因此自毁性不属于计算机病毒的特征。

MD5 是（9）算法，对任意长度的输入计算得到的结果长度为（10）位。

（9）A.路由选择　　B.摘要　　C.共享密钥　　D.公开密钥

（10）A.56　　B.128　　C.140　　D.160

【答案】B　B

【解析】

MD5 是一种摘要算法，经过一系列处理后，算法的输出由四个 32 位分组组成，将这四个 32 位分组级联后将生成一个 128 位散列值。

使用 Web 方式收发电子邮件时，以下描述错误的是（11）。

（11）A.无须设置简单邮件传输协议

B.可以不设置帐号密码登录

C.邮件可以插入多个附件

D.未发送邮件可以保存到草稿箱

【答案】B

【解析】

使用 WEB 方式收发电子邮件是必须设置账号密码登录。

有可能无限期拥有的知识产权是（12）。

（12）A.著作权　　B.专利权　　C.商标权　　D.集成电路布图设计权

【答案】C

【解析】

客体类型	权力类型	保护期限
公民作品	署名权、修改权、保护作品完整权	没有限制
	发表权、使用权和获得报酬权	作者终生及其死亡后的50年（第50年的12月31日）
单位作品	发表权、使用权和获得报酬权	50年（首次发表后的第50年的12月31日），若其间未发表，不保护。
公民软件产品	署名权、修改权	没有限制
	发表权、复制权、发行权、出租权、信息网络传播权、翻译权、使用许可权、获得报酬权、转让权	作者终生及死后50年（第50年12月31日）。合作开发，以最后死亡作者为准。
单位软件产品	发表权、复制权、发行权、出租权、信息网络传播权、翻译权、使用许可权、获得报酬权、转让权	50年（首次发表后的第50年的12月31日），若其间未发表，不保护
注册商标		有效期10年（若注册人死亡或倒闭1年后，未转移则可注销，期满后6个月内必须续注）
发明专利权		保护期为20年（从申请日开始）
实用新型和外观设计专利权		保护期为10年（从申请日开始）
商业秘密		不确定，公开后公众可用

其中商标权可以通过续注延长拥有期限，而著作权、专利权和设计权的保护期限都是有限期的。

（13）是构成我国保护计算机软件著作权的两个基本法律文件。

（13）A.《软件法》和《计算机软件保护条例》

B.《中华人民共和国著作权法》和《计算机软件保护条例》

C.《软件法》和《中华人民共和国著作权法》

D.《中华人民共和国版权法》和《计算机软件保护条例》

【答案】B

【解析】

我国保护计算机软件著作权的两个基本法律文件是《中华人民共和国著作权法》和《计算机软件保护条例》。

某软件程序员接受一个公司（软件著作权人）委托开发完成一个软件，三个月后又接受另一公司委托开发功能类似的软件，此程序员仅将受第一个公司委托开发的软件略作修改即提交给第二家公司，此种行为（14）。

（14）A.属于开发者的特权　　B.属于正常使用著作权

C.不构成侵权　　D.构成侵权

【答案】D

【解析】

本题已注明第一个公司为软件著作权人，因此该程序员的行为对原公司构成侵权。

结构化分析的输出不包括（15）。

（15）A.数据流图　　B.数据字典　　C.加工逻辑　　D.结构图

【答案】D

【解析】

结构化方法的分析结果由以下几部分组成：一套分层的数据流图、一本数据词典、一组小说明（也称加工逻辑说明）、补充材料。

因此结构图不属于结构化分析的输出。

某航空公司拟开发一个机票预订系统， 旅客预订机票时使用信用卡付款。付款通过信用卡公司的信用卡管理系统提供的接口实现。若采用数据流图建立需求模型，则信用卡管理系统是（16）。

（16）A.外部实体　　B.加工　　C.数据流　　D.数据存储

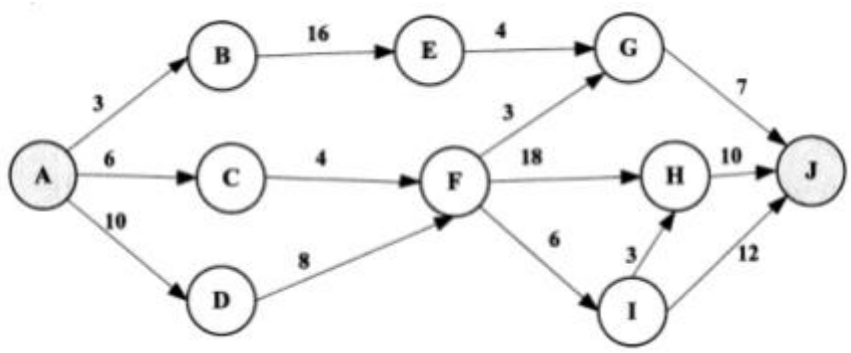
【答案】A

【解析】

数据流图中的基本图形元素包括数据流、加工、数据存储和外部实体。其中，数据流、加工和数据存储用于构建软件系统内部的数据处理模型，而外部实体表示存在于系统之外的

对象，用来帮助用户理解系统数据的来源和去向。外部实体包括：人/物、外部系统、组织机构等。

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（17）天。活动 FG 的松驰时间为（18）天。



（17）A.20　　B.37　　C.38　　D.46

（18）A.9　　B.10　　C.18　　D.26

【答案】D　C

【解析】

以下叙述中，（19）不是一个风险。

（19）A.由另一个小组开发的子系统可能推迟交付，导致系统不能按时交付客户

B.客户不清楚想要开发什么样的软件，因此开发小组开发原型帮助其确定需求

C.开发团队可能没有正确理解客户的需求

D.开发团队核心成员可能在系统开发过程中离职

【答案】B

【解析】本题考查的是风险的概念。

一般认为风险保护两个特性：不确定性和损失。不确定性是指风险可能发生也可能不发生；损失是指如果风险发生，就会产生恶性后果。

本题选项“客户不清楚想要开发什么样的软件”是已经发生的事件，没有不确定性，因此不是一个风险。

对布尔表达式进行短路求值是指：无须对表达式中所有操作数或运算符进行计算就可确

定表达式的值。对于表达式"a or （（c< d） and b） "，（20）时可进行短路计算。

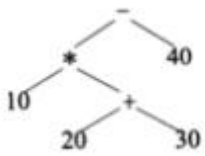
（20）A. d 为 true　　B. a 为 true　　C. b 为 true　　D. c 为 true

【答案】B

【解析】

根据本题题干“a or ((c<d) and b)”，最后计算的是 or，对于或运算，只要有一个为真则结果为真，不需要进行后面的计算，因此当 a 为 true 时，可进行短路计算，直接得到后面的结果。

下面二叉树表示的简单算术表达式为（21）。



（21）A.10*20+30-40　　B.10*（20+30-40）

C.10*（20+30）-40　　D.10*20+（30-40）

【答案】C

【解析】

在程序运行过程中，（22）时涉及整型数据转换为浮点型数据的操作。

（22）A.将浮点型变量赋值给整型变量　　B.将整型常量赋值给整型变量

C.将整型变量与浮点型变量相加　　D.将浮点型常量与浮点型变量相加

【答案】C

【解析】

某计算机系统中互斥资源 R 的可用数为 8，系统中有 3 个进程 P1、P2 和 P3 竞争 R，且每个进程都需要 i 个 R，该系统可能会发生死锁的最小 i 值为（23）。

（23）A. 1　　B. 2　　C. 3　　D. 4

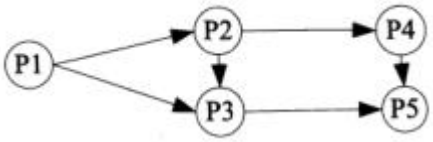
【答案】D

【解析】

本题对于 R 资源可用数为 8，分配到 3 个进程中，为了让最后的 i 值最小，所以每个进

程尽量平均分配，可以得到 3 、3、2 的分配情况，此时如果假设 i 的取值为 3，则必定不会形成死锁。当 i>3 时系统会形成死锁，此时取整，即最小 i 值为 4。

进程 P1、P2、P3、P4 和 P5 的前趋图如下所示：



若用 PV 操作控制这 5 个进程的同步与互斥的程序如下，那么程序中的空①和空②处应分别为（24）；空③和空④处应分别为（25）；空⑤和空⑥处应分别为（26）。

```
begin
  S1,S2,S3, S4, S5, S6: semaphore;      //定义信号量
  S1:=0; S2:=0; S3:=0; S4:=0; S5:=0; S6:=0;
  Cobegin
    process P1    process P2    process P3    process P4    process P5
      Begin         Begin         Begin         Begin         Begin
        P1 执行;      ②;            P(S2);        P(S4);        ⑥;
        V(S1)         P2 执行;      ③;            P4 执行;      P5 执行;
        ①;            V(S3);        P3 执行;      ⑤;            end;
      end;            V(S4);        ④;          end;
    Coend;          end;          end;
end.
```

（24）A. V（S1）和 P（S2）　　B. P（S1）和 V（S2）

C. V（S1）和 V（S2）　　D. V（S2）和 P（S1）

（25）A.V（S3）和 V（S5）　　B.P（S3）和 V（S5）

C.V（S3）和 P（S5）　　D.P（S3）和 P（S5）

（26）A.P（S6）和 P（S5）V（S6）　　B.V（S5）和 V（S5）V（S6）

C.V（S6）和 P（S5）P（S6）　　D.P（S6）和 P（S5）P（S6）

【答案】D　B　C

【解析】本题考查的是利用 PV 操作控制进程的并发执行。

先理清楚前趋图中的逻辑关系：P1 没有前驱，P2 的前驱是 P1，P3 的前驱是 P1、P2，P4 的前驱是 P2，P5 的前驱是 P3、P4。

前驱就是指只有在前驱进程完成后，该进程才能开始执行。由图可知，这里进程之间有 6 条有向弧，分别表示为 P1->P2，P1-P3，P2->P3，P2->P4，P3->P5，P4->P5，各个进程间的

逻辑关系，那么我们需要设定 6 个信号量(S1、S2、S3、S4、S5、S6)，利用 PV 操作来控制这些过程。

对于第一个空，P1 执行完成之后，需要通知 P2、P3 可以开始，此处需要 V(S1)、V(S2)操作分别唤醒 P2、P3 进程，已有 V(S1)，此处需要填写 V(S2)。

对于第二个空，P2 执行之前，需要检查 P1 进程是否完成，因此需要通过 P(S1)操作来判定，P1 是否完成。

对于第三个空，在 P3 执行之前，需要检查 P1、P2 进程是否完成，因此需要通过 P(S2)、P(S3)操作来判定 P1、P2 是否完成，已有 P(S2)，此处填写 P(S3)。

对于第四空，P3 执行完成后，需要通知 P5 进程可以开始，此处需要通过 V(S5)操作唤醒 P5 进程；

对于第五空，P4 进程完成后，需要通知 P5 进程可以开始，此处需要通过 V(S6)操作唤醒 P5 进程；

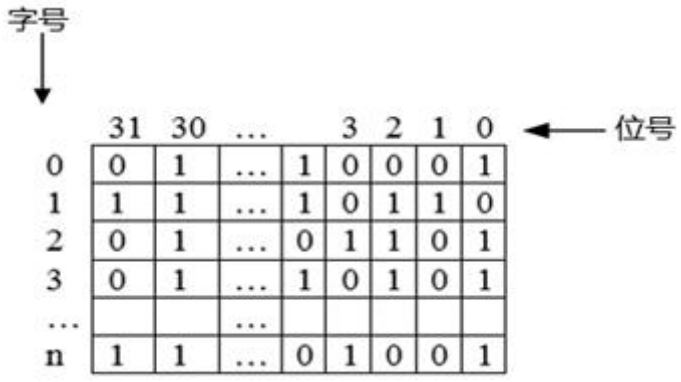
对于第六空，P5 进程开始之前，需要检查 P3、P4 进程是否已完成，因此需要 P(S5)、P(S6)操作来判断 P3、P4 是否完成。

某文件管理系统在磁盘上建立了位示图（bitmap），记录磁盘的使用情况。若磁盘上物理块的编号依次为：0、1、2、….；系统中的字长为 32 位，位示图中字的编号依次为：0、1、2、..，每个字中的一个二进制位对应文件存储器上的一个物理块，取值 0 和 1 分别表示物理块是空闲或占用。假设操作系统将 2053 号物理块分配给某文件，那么该物理块的使用情况在位示图中编号为（27）的字中描述。

（27）A.32　　B.33　　C.64　　D.65

【答案】C

【解析】



2053 号物理块是第 2054 块物理块，每一个字可以表示 32 个物理块的存储情况，2054/32=64.xxx，因此，此时应该排在第 65 个字，从 0 号开始编号，则为第 64 号字。

某操作系统文件管理采用索引节点法。每个文件的索引节点有 8 个地址项，每个地址项大小为 4 字节，其中 5 个地址项为直接地址索引，2 个地址项是一级间接地址索引，1 个地址项是二级间接地址索引，磁盘索引块和磁盘数据块大小均为 1KB。若要访问文件的逻辑块号分别为 1 和 518，则系统应分别采用（28）。

（28）A.直接地址索引和一级间接地址索引

B.直接地址索引和二级间接地址索引

C.一级间接地址索引和一级间接地址索引

D.一级间接地址索引和二级间接地址索引

【答案】B

【解析】

每个物理块大小为 1KB，每个地址项大小为 4B，因此每个物理块可以对应地址项个数为：1KB/4B=256。

直接索引即索引直接指向物理块，可以表示逻辑块号范围：0~4 号

一级索引即索引节点指向的物理块用来存放地址项，可以表示 256 个地址项，即 256 个物理块，可以表示逻辑地址块号范围：5~258，259~515 号

二级索引即索引节点指向的物理块，存放的是一级索引的地址块地址，一共有 256 个地址块用来存放一级索引，每个块可以存放 256 个地址项，共有 2562=65536 个地址项，因此可以表示的逻辑块号范围：516~66052 号

某企业拟开发一个企业信息管理系统，系统功能与多个部门的业务相关。现希望该系统能够尽快投入使用，系统功能可以在使用过程中不断改善。则最适宜采用的软件过程模型为（29）。

（29）A.瀑布模型　　B.原型模型　　C.演化（迭代）模型　　D.螺旋模型

【答案】C

【解析】

本题要求尽量投入使用，并可以再使用过程中不断完善，对于原型模型和演化（迭代）模型，演化模型更合适，原型模型更适用于需求不明确时用以获取需求。

能力成熟度模型集成（CMMI）是若干过程模型的综合和改进。连续式模型和阶段式模型是 CMMI 提供的两种表示方法，而连续式模型包括 6 个过程域能力等级，其中（30）使用量化（统计学）手段改变和优化过程域，以应对客户要求的改变和持续改进计划中的过程域的功效。

（30）A.CL2（已管理的）　　B.CL3（已定义级的）

C.CL4（定量管理的）　　D.CL5（优化的）

【答案】D

【解析】

CL0（未完成的）：过程域未执行或未得到 CL1 中定义的所有目标。

CL1（已执行的）：其共性目标是过程将可标识的输入工作产品转换成可标识的输出工作产品，以实现支持过程域的特定目标。

CL2（已管理的）：其共性目标是集中于已管理的过程的制度化。根据组织级政策规定过程的运作将使用哪个过程，项目遵循已文档化的计划和过程描述，所有正在工作的人都有权使用足够的资源，所有工作任务和工作产品都被监控、控制、和评审。

CL3（已定义级的）：其共性目标集中于已定义的过程的制度化。过程是按照组织的裁剪指南从组织的标准过程中裁剪得到的，还必须收集过程资产和过程的度量，并用于将来对过程的改进。

CL4（定量管理的）：其共性目标集中于可定量管理的过程的制度化。使用测量和质量保证来控制和改进过程域，建立和使用关于质量和过程执行的质量目标作为管理准则。

CL5（优化的）：使用量化（统计学）手段改变和优化过程域，以满足客户的改变和持续改进计划中的过程域的功效。

在 ISO/IEC 9126 软件质量模型中，可靠性质量特性是指在规定的一段时间内和规定的条件下，软件维持在其性能水平有关的能力，其质量子特性不包括（31）。

（31）A.安全性　　B.成熟性　　C.容错性　　D.易恢复性

【答案】A

【解析】

以下关于模块化设计的叙述中，不正确的是（32）。

（32）A.尽量考虑高内聚、低耦合，保持模块的相对独立性

B.模块的控制范围在其作用范围内

C.模块的规模适中

D.模块的宽度、深度、扇入和扇出适中

【答案】B

【解析】

模块化设计要求高内聚、低耦合。

在结构化设计中，系统由多个逻辑上相对独立的模块组成，在模块划分时需要遵循如下原则：

（1）模块的大小要适中。系统分解时需要考虑模块的规模，过大的模块可能导致系统分解不充分，其内部可能包括不同类型的功能，需要进一步划分，尽量使得各个模块的功能单一；过小的模块将导致系统的复杂度增加，模块之间的调用过于频繁，反而降低了模块的独立性。一般来说，一个模块的大小使其实现代码在 1～2 页纸之内，或者其实现代码行数在 50～200 行之间，这种规模的模块易于实现和维护。

（2）模块的扇入和扇出要合理。一个模块的扇出是指该模块直接调用的下级模块的个数；扇出大表示模块的复杂度高，需要控制和协调过多的下级模块。扇出过大一般是因为缺乏中间层次，应该适当增加中间层次的控制模块；扇出太小时可以把下级模块进一步分解成若干个子功能模块，或者合并到它的上级模块中去。一个模块的扇入是指直接调用该模块的上级模块的个数；扇入大表示模块的复用程度高。设计良好的软件结构通常顶层扇出比较大，中间扇出较少，底层模块则有大扇入。一般来说，系统的平均扇入和扇出系数为 3 或 4，不应该超过 7，否则会增大出错的概率。

（3）深度和宽度适当。深度表示软件结构中模块的层数，如果层数过多，则应考虑是否有些模块设计过于简单，看能否适当合并。宽度是软件结构中同一个层次上的模块总数的最大值，一般说来，宽度越大系统越复杂，对宽度影响最大的因素是模块的扇出。在系统设计时，需要权衡系统的深度和宽度，尽量降低系统的复杂性，减少实施过程的难度，提高开发和维护的效率。

模块的扇入指模块直接上级模块的个数。模块的直属下级模块个数即为模块的扇出。

某企业管理信息系统中，采购子系统根据材料价格、数量等信息计算采购的金额，并给财务子系统传递采购金额、收款方和采购日期等信息，则这两个子系统之间的耦合类型为（33）

耦合。

（33）A.数据　　B.标记　　C.控制　　D.外部

【答案】A

【解析】

直接耦合：两个模块之间没有直接关系，它们之间的联系完全是通过主模块的控制和调用来实现的。

数据耦合：一个模块访问另一个模块时，彼此之间是通过简单数据参数 (不是控制参数、公共数据结构或外部变量) 来交换输入、输出信息的。

标记耦合 ：一组模块通过参数表传递记录信息，就是标记耦合。这个记录是某一数据结构的子结构，而不是简单变量。其实传递的是这个数据结构的地址；

控制耦合：如果一个模块通过传送开关、标志、名字等控制信息，明显地控制选择另一模块的功能，就是控制耦合。

外部耦合：一组模块都访问同一全局简单变量而不是同一全局数据结构，而且不是通过参数表传递该全局变量的信息，则称之为外部耦合。

公共耦合：若一组模块都访问同一个公共数据环境，则它们之间的耦合就称为公共耦合。公共的数据环境可以是全局数据结构、共享的通信区、内存的公共覆盖区等。

内容耦合：如果发生下列情形，两个模块之间就发生了内容耦合

(1)一个模块直接访问另一个模块的内部数据；

(2)一个模块不通过正常入口转到另一模块内部；

(3)两个模块有一部分程序代码重叠(只可能出现在汇编语言中)；

(4)一个模块有多个入口。

本题属于数据耦合，采购子系统模块给财务子系统模块传递数据。

对以下的程序伪代码（用缩进表示程序块）进行路径覆盖测试，至少需要（34）个测试用例。采用 McCabe 度量法计算其环路复杂度为（35）。

（34）A.2　　B.4　　C.6　　D.8

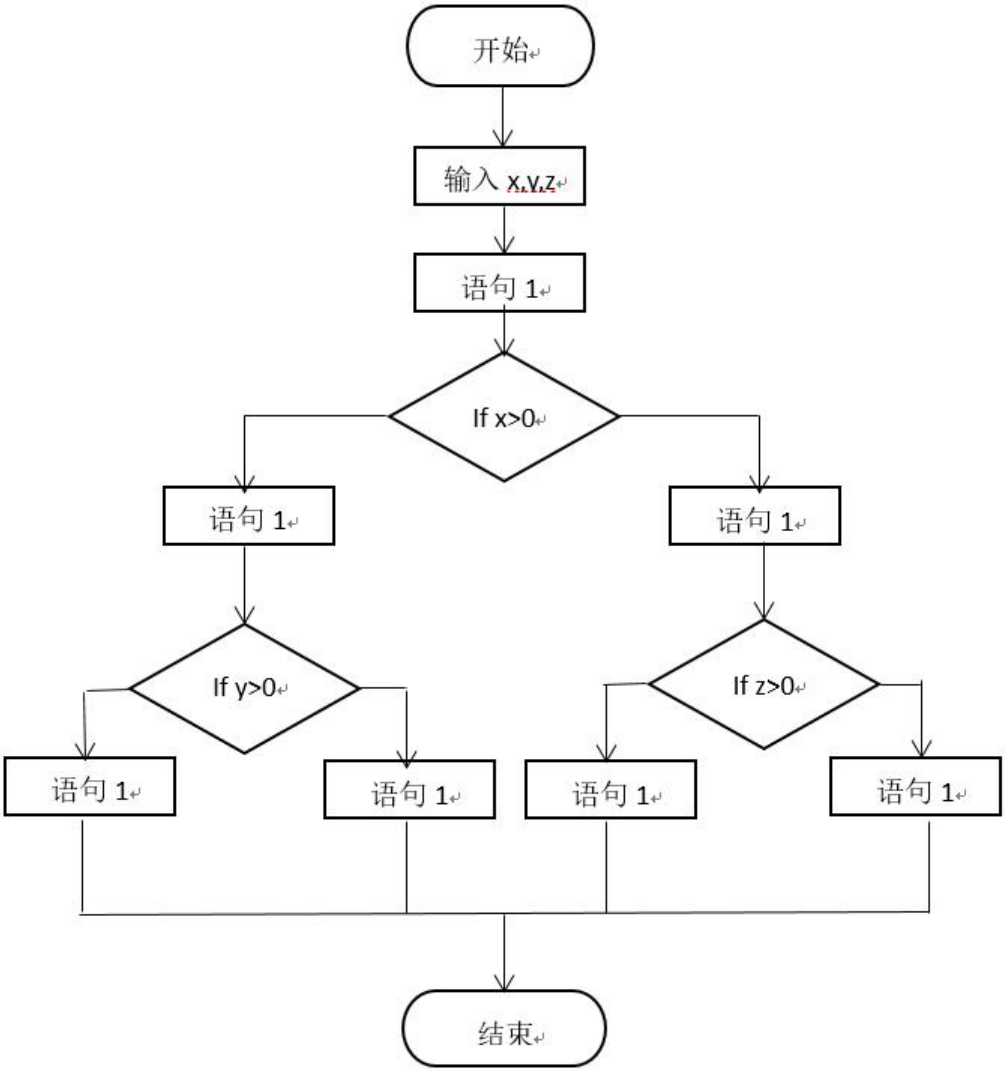
（35）A.2　　B.3　　C.4　　D.5

【答案】B　C

【解析】

对于本题，用例（x,y,z）分别为（1，1，0）（1，-1，0）（-1，0，1）（-1，0，-1），这 4 个测试用例可以走完所有可能路径。因为在伪代码中，我们可以看到，当 x>0 时，只需要对 Y 分别取大于 0 和不大于 0 的值即可，z 不参与比较；当 x 不大于 0 时，只需要对 z 分别取大于 0 和不大于 0 的值即可，y 不参与比较，只需要 4 个用例即可。

对于第二空，转换为结点图如下：



根据 V(G)=m-n+2，其中 m 是有向图的弧，为 15，n 为有向图的节点数，为 13，15-13+2=4，即环路复杂的为 4。

某商场的销售系统所使用的信用卡公司信息系统的数据格式发生了更改，因此对该销售系统进行的修改属于（36）维护。

（36）A.改正性　　B.适应性　　C.改善性　　D.预防性

【答案】B

【解析】

在系统运行过程中，软件需要维护的原因是多样的，根据维护的原因不同，可以将软件维护分为以下四种：

（1）改正性维护。为了识别和纠正软件错误、改正软件性能上的缺陷、排除实施中的误使用，应当进行的诊断和改正错误的过程就称为改正性维护。

（2）适应性维护。在使用过程中，外部环境（新的硬、软件配置）、数据环境（数据库、数据格式、数据输入/输出方式、数据存储介质）可能发生变化。为使软件适应这种变化，而去修改软件的过程就称为适应性维护。

（3）完善性维护。在软件的使用过程中，用户往往会对软件提出新的功能与性能要求。为了满足这些要求，需要修改或再开发软件，以扩充软件功能、增强软件性能、改进加工效率、提高软件的可维护性。这种情况下进行的维护活动称为完善性维护。

（4）预防性维护。这是指预先提高软件的可维护性、可靠性等，为以后进一步改进软件打下良好基础。

本题对该销售系统的修改是为了应对数据格式的变化而做出的修改。

在面向对象方法中，继承用于（37）。

（37）A.在已存在的类的基础上创建新类　　B.在已存在的类中添加新的方法

C.在已存在的类中添加新的属性　　D.在已存在的状态中添加新的状态

【答案】A

【解析】

本题考查的是继承的定义：继承是类之间的一种关系，在定义和实现一个类的时候，可以在一个已经存在的类的基础上进行。

（38）多态是指操作（方法）具有相同的名称、且在不同的上下文中所代表的含义不同

（38）A.参数　　B.包含　　C.过载　　D.强制

【答案】C

【解析】

参数多态：应用广泛、最纯的多态。

包含多态：同样的操作可用于一个类型及其子类型。包含多态一般需要进行运行时的类型检查。

强制多态：编译程序通过语义操作，把操作对象的类型强行加以变换，以符合函数或操作符的要求。

过载多态：同一个名（操作符、函数名）在不同的上下文中有不同的类型。

在某销售系统中，客户采用扫描二维码进行支付。若采用面向对象方法开发该销售系统，

则客户类属于（39）类， 二维码类属于（40）类。

（39）A.接口　　B.实体　　C.控制　　D.状态

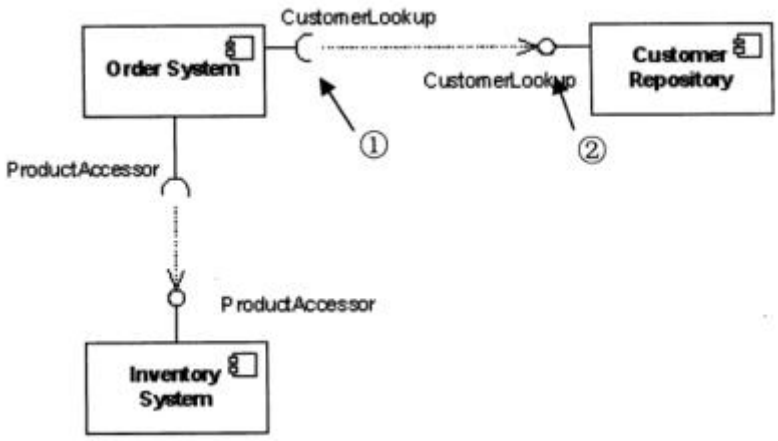
（40）A.接口　　B.实体　　C.控制　　D.状态

【答案】B　A

【解析】

类可以分为三种：实体类、接口类（边界类）和控制类。实体类的对象表示现实世界中真实的实体，如人、物等。接口类（边界类）的对象为用户提供一种与系统合作交互的方式，分为人和系统两大类，其中人的接口可以是显示屏、窗口、Web 窗体、对话框、菜单、列表框、其他显示控制、条形码、二维码或者用户与系统交互的其他方法。系统接口涉及到把数据发送到其他系统，或者从其他系统接收数据。控制类的对象用来控制活动流，充当协调者。

下图所示 UML 图为（41），用于展示（42）。①和②分别表示（43）。



（41）A.类图　　B.组件图　　C.通信图　　D.部署图

（42）A.一组对象、接口、协作和它们之间的关系

B.收发消息的对象的结构组织

C.组件之间的组织和依赖

D.面向对象系统的物理模型

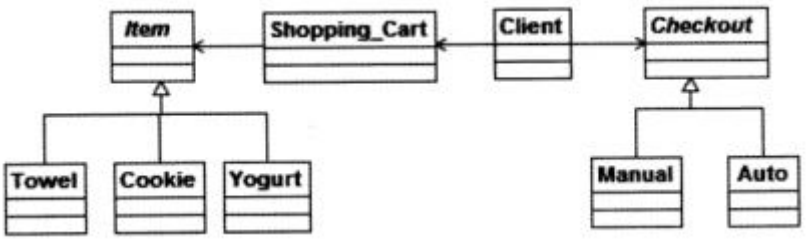
（43）A.供接口和供接口　　B.需接口和需接口

C.供接口和需接口　　D.需接口和供接口

【答案】B　C　C

【解析】

假设现在要创建一个简单的超市销售系统，顾客将毛巾、饼干、酸奶等物品（Item）加入购物车（Shopping_Cart），在收银台（Checkout）人工（Manual）或自动（Auto）地将购物车中每个物品的价格汇总到总价格后结帐。这一业务需求的类图（方法略）设计如下图所示，采用了（44）模式。其中（45）定义以一个 Checkout 对象为参数的 accept 操作，由子类实现此 accept 操作。此模式为（46），适用于（47）。



（44）A.观察者（Observer）　　B.访问者（Visitor）

　　C.策略（Strategy）　　D.桥接器（Bridge）

（45）A.Item　　B.Shopping_Cart　　C.Checkout　　D.Manual 和 Auto

（46）A.创建型对象模式　　B.结构型对象模式

　　C.行为型类模式　　D.行为型对象模式

（47）A.必须保存一个对象在某一个时刻的（部分）状态

　　B.想在不明确指定接收者的情况下向多个对象中的一个提交一个请求

　　C.需要对一个对象结构中的对象进行很多不同的并且不相关的操作

　　D.在不同的时刻指定、排列和执行请求

【答案】B　A　D　C

【解析】本题为访问者模式。

对于观察者模式是一个被观察者和多个观察者对象，与图示不符；桥接模式是结构型模式，存在部分与整体的联系，与本题不符；策略模式是对于不同算法的封装和切换，但是调用策略的对象只有一个，与本题不符。一个对象结构包含很多类对象（Item），而系统要求这些对象实施一些依赖于某具体类（Checkout）的操作时，可以使用访问者模式。

在以阶段划分的编译器中，（48）阶段的主要作用是分析程序中的句子结构是否正确。

（48）A.词法分析　　B.语法分析　　C.语义分析　　D.代码生成

【答案】B

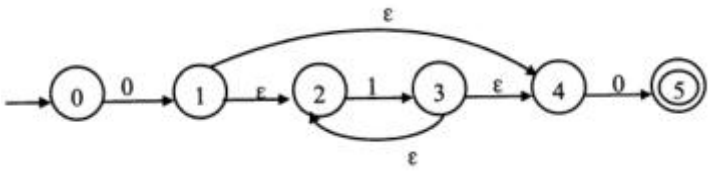
【解析】

词法分析：从左到右逐个扫描源程序中的字符，识别其中如关键字(或称保留字)、标识符、常数、运算符以及分隔符（标点符号和括号）等。

语法分析：根据语法规则将单词符号分解成各类语法单位，并分析源程序是否存在语法上的错误。包括：语言结构出错、if…end if 不匹配，缺少分号、括号不匹配、表达式缺少操作数等。本题属于语法分析阶段的作用。

语义分析：进行类型分析和检查，主要检测源程序是否存在静态语义错误。包括：运算符和运算类型不符合，如取余时用浮点数

下图所示为一个不确定有限自动机（NFA）的状态转换图。该 NFA 可识别字符串（49）。



（49）A.0110　　B.0101　　C.1100　　D.1010

【答案】A

【解析】

本题因为是不确定的有限自动机，中间内容由多种可能，但由图可以看到，从初态 0 开始，首字符只能为 0，到终态结束之前，尾字符也只能为 0。

函数 f 和 g 的定义如下图所示。执行函数 f 时若采用引用（call by reference）方式调用函数 g（a），则函数 f 的返回值为（50）。

```
f()
int a = 5, c;
c = g(a);
return a+c;
```

```
g(形式参数 x)
int m = 2;
m = x * m ;   x = m - 1;
return x+m;
```

（50）A.14　　B.18　　C.24　　D.28

【答案】D

【解析】本题采用引用调用，会改变实参的值。

对于实参 a，传递给 g(a)之后，在 g(a)函数，表现为形参 x。

根据 g(x)代码：m=5*2=10，x=10-1=9，返回值 x+m=19；

返回 f()代码，此时 a（即 g(x)中的 x）的值已经改变，为 9；c 等于 g(a)的返回值，也就是 19。

最终可得 f()的返回值 a+c=28。

数据库系统中的视图、存储文件和基本表分别对应数据库系统结构中的（51）。

（51）A.模式、内模式和外模式　　B.外模式、模式和内模式

C.模式、外模式和内模式　　D.外模式、内模式和模式

【答案】D

【解析】

本题考察的是数据库体系结构：三层模式。对于题干给出的视图、存储文件、基本表分别对应：视图-外模式，存储文件-内模式，基本表-模式。

在分布式数据库中，（52）是指用户或应用程序不需要知道逻辑上访问的表具体如何分块存储。

（52）A.逻辑透明　　B.位置透明　　C.分片透明　　D.复制透明

【答案】C

【解析】本题考查的是分布式数据库相关知识。

分片透明：是指用户不必关系数据是如何分片的，它们对数据的操作在全局关系上进行，即关系如何分片对用户是透明的，因此，当分片改变时应用程序可以不变。分片透明性是最高层次的透明性，如果用户能在全局关系一级操作，则数据如何分布，如何存储等细节自不必关系，其应用程序的编写与集中式数据库相同。

复制透明：用户不用关心数据库在网络中各个节点的复制情况，被复制的数据的更新都由系统自动完成。在分布式数据库系统中，可以把一个场地的数据复制到其他场地存放，应用程序可以使用复制到本地的数据在本地完成分布式操作，避免通过网络传输数据，提高了系统的运行和查询效率。但是对于复制数据的更新操作，就要涉及到对所有复制数据的更新。

位置透明：是指用户不必知道所操作的数据放在何处，即数据分配到哪个或哪些站点存储对用户是透明的

局部映像透明性（逻辑透明）是最低层次的透明性，该透明性提供数据到局部数据库的

映像，即用户不必关系局部 DBMS 支持哪种数据模型、使用哪种数据操纵语言，数据模型和操纵语言的转换是由系统完成的。因此，局部映像透明性对异构型和同构异质的分布式数据库系统是非常重要的。

本题提到不需要了解具体如何分块存储，如果描述为不需要了解物理存储或存储位置，则为位置透明，而涉及到如果分块存储，应该为分片透明。对于分布式数据库，分片是一种大局性的划分，而物理上的存储位置则更为底层，所以对于如何分块存储，强调更多的是分片而不是物理位置。

设有关系模式 R（A1，A2，A3，A4，A5，A6），函数依赖集 F={A1→A3，A1 A2→A4，A5 A6→A1，A3 A5→A6，A2 A5→A6}。关系模式 R 的一个主键是（53）， 从函数依赖集 F 可以推出关系模式 R（54）。

（53）A.A1A4　　B.A2A5　　C.A3A4　　D.A4A5

（54）A.不存在传递依赖，故 R 为 1NF

B.不存在传递依赖，故 R 为 2NF

C.存在传递依赖，故 R 为 3NF

D.每个非主属性完全函数依赖于主键，故 R 为 2NF

【答案】B　D

【解析】本题看起来逻辑很复杂，但解题相对比较简单。

根据函数依赖集，可以简单分析，在本题中唯一入度为 0 的属性为 A2，因此，A2 一定属于候选键集合，在选项中只有 B 选项符合要求。

第二空，根据第一空可知 R 的一个主键为 A2A5，由函数依赖集 F 可知，存在 A2A5àA6，A5A6àA1，A1àA3，这里存在传递函数依赖，故 A、B 选项均不正确，C 选项本身不正确，存在非主属性对候选键的传递函数依赖，是不满足 3NF 的。因此本题选择 D 选项。

也可将完整的依赖图示绘制出来判断本题 A2A5 为候选键，并且每个非主属性完全函数依赖于主键。

给定关系 R（A, B,C,D）和 S（C,D,E），若关系 R 与 S 进行自然连接运算，则运算后的元组属性列数为（55）；关系代数表达式 $\pi_{1,4}(\sigma_{2=5}(R \bowtie S))$ 与（56）等价。

（55）A.4　　B.5　　C.6　　D.7

（56）A.πA,D（σC=D（R×S））　　B.πR.A,R.D（σR.B=S.C（R×S））

C.πA,R.D（σR.C=S.D（R×S））　　D.πR.A,R.D（σR.B=S.E（R×S））

【答案】B　D

【解析】本题考查的是数据库中关系代数的相关知识内容。

对于第一空，关系 R 与 S 进行自然连接后，属性列数为二者之后并减去其中的重复列，本题 R 和 S 都存在 C、D 属性，因此自然连接后属性列数为 4+3-2=5。

栈的特点是后进先出，若用单链表作为栈的存储结构，并用头指针作为栈顶指针，则（57）。

（57）A.入栈和出栈操作都不需要遍历链表

B.入栈和出栈操作都需要遍历链表

C.入栈操作需要遍历链表而出栈操作不需要

D.入栈操作不需要遍历链表而出栈操作需要

【答案】A

【解析】

本题用单链表作为栈的存储结构，因为栈的操作是先进后出，因此无论是入栈还是出栈，都只对栈顶元素操作，而在单链表中用头指针作为栈顶指针，此时无论是出栈还是入栈，都只需要对头指针指向的栈顶指针操作即可，不需要遍历链表。

已知某二叉树的先序遍历序列为 A B C D E F、中序遍历序列为 B A D C F E，则可以确定该二叉树（58）。

（58）A.是单支树（即非叶子结点都只有一个孩子）

B.高度为 4（即结点分布在 4 层上）

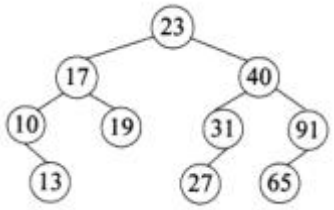
C.根结点的左子树为空

D.根结点的右子树为空

【答案】B

【解析】

可以构造出下图所示二叉排序树（二叉检索树、二叉查找树）的关键码序列是（59）。



（59）A.10 13 17 19 23 27 31 40 65 91　　B.23 40 91 17 19 10 31 65 27 13

C.23 19 40 27 17 13 10 91 65 31　　D.27 31 40 65 91 13 10 17 23 19

【答案】B

【解析】

二叉排序树的构造过程：

若查找二叉树为空树，则以新结点为查找二叉树；

将要插入结点键值与插入后父结点键值比较，就能确定新结点是父结点的左子结点，还是右子结点，直到将序列中的所有元素（关键码）全部插入。

根据排序二叉树的构造过程，可知 A 选项的根节点为 10，D 选项的根节点为 27，因此可以排除。对于 C 选项，构造根节点的子结点，可知 19 为其左孩子结点，与图不符。本题只有 B 选项可以构造出图示的排序二叉树。

图 G 的邻接矩阵如下图所示（顶点依次表示为 v0、v1、v2、v3、v4、v5），G 是（60）。对 G 进行广度优先遍历（从 v0 开始），可能的遍历序列为（61）。

$$\begin{bmatrix} \infty & 18 & 17 & \infty & \infty & \infty \\ \infty & \infty & \infty & 20 & 16 & \infty \\ \infty & 19 & \infty & 23 & \infty & \infty \\ \infty & \infty & \infty & \infty & \infty & 15 \\ \infty & \infty & \infty & \infty & \infty & 12 \\ \infty & \infty & \infty & \infty & \infty & \infty \end{bmatrix}$$

（60）A.无向图　　B.有向图　　C.完全图　　D.强连通图

（61）A.v0、v1、v2、v3、v4、v5　　B.v0、v2、v4、 v5、v1、v3

C.v0、v1、v3、v5、v2、v4　　D.v0、v2、v4、v3、v5、v1

【答案】B　A

【解析】

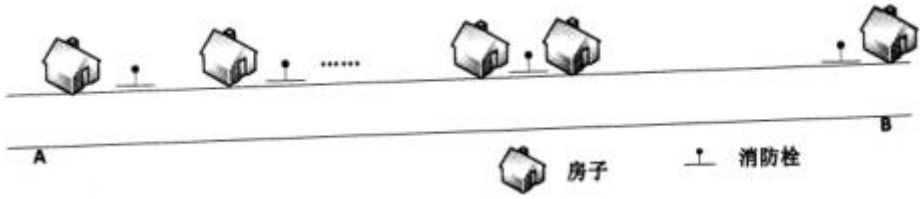
由邻接矩阵可知，对于结点 V0 和 V1 之间，只存在弧 V0àV1，而没有弧 V1àV0，因此图 G 不属于无向图，也不属于完全图。

强连通图：在有向图 G 中如果对于每一对顶点 Vi，Vj，从顶点 Vi 到顶点 Vj 和从顶点 Vj 到顶点 Vi 都存在路径，则称图为强连通图。本题不满足该条件。

因此本题第一空应该选择 B 选项有向图。

对于第二空，图的广度遍历过程：从图中的某个顶点 V 触发，在访问了 V 之后一次访问 V 的各个未被访问的邻接点，然后分别从这些邻接点出发，依次访问它们的邻接点，并使“先被访问的顶点的邻接点”先于“后被访问的顶点的邻接点”被访问，直到图中所有已被访问的顶点的邻接点都被访问到。本题从 V0 出发，一次访问其邻接点 V1、V2，只有 A 选项符合条件。

在一条笔直公路的一边有许多房子，现要安装消防栓，每个消防栓的覆盖范围远大于房子的面积，如下图所示。现求解能覆盖所有房子的最少消防栓数和安装方案（问题求解过程中，可将房子和消防栓均视为直线上的点）。



该问题求解算法的基本思路为：从左端的第一栋房子开始，在其右侧 m 米处安装一个消防栓，去掉被该消防栓覆盖的所有房子。在剩余的房子中重复上述操作，直到所有房子被覆盖。算法采用的设计策略为（62）；对应的时间复杂度为（63）。

假设公路起点 A 的坐标为 0，消防栓的覆盖范围（半径）为 20 米，10 栋房子的坐标为（10，20,，30，35，60，80，160，210，260，300），单位为米。根据上述算法，共需要安装（64）个消防栓。以下关于该求解算法的叙述中，正确的是（65）。

（62）A.分治　　B.动态规划　　C.贪心　　D.回溯

（63）A. $\Theta(\lg n)$　　B. $\Theta(n)$　　C. $\Theta(n\lg n)$　　D. $\Theta(n^2)$

（64）A.4　　B.5　　C.6　　D.7

（65）A.肯定可以求得问题的一个最优解　　B.可以求得问题的所有最优解

C.对有些实例，可能得不到最优解　　D.只能得到近似最优解

【答案】C　B　B　C

【解析】

（一）对于第一空，本题使用的是贪心法。

1、分治法特征：对于一个规模为 n 的问题，若该问题可以容易地解决（比如说规模 n 较小）则直接解决；否则将其分解为 k 个规模较小的子问题，这些子问题互相独立且与原问题形式相同，递归地解这些子问题，然后将各子问题的解合并得到原问题的解。

2、动态规划法：在求解问题中，对于每一步决策，列出各种可能的局部解，再依据某种判定条件，舍弃那些肯定不能得到最优解的局部解，在每一步都经过筛选，以每一步都是最优解来保证全局是最优解。本题情景没有列出所有的可能解进行筛选，因此，本题不属于动态规划法。

3、回溯法：回溯法是一种选优搜索法，按选优条件向前搜索，以达到目标。但当搜索到某一步时，发现原先选择并不优或达不到目标，就退回一步重新选择。这种走不通就退回再走的技术就是回溯法。本题情景没有探索和回退的过程，因此，本题不属于回溯法。

4、贪心法：总是做出在当前来说是最好的选择，而并不从整体上加以考虑，它所做的每步选择只是当前步骤的局部最优选择，但从整体来说不一定是最优的选择。由于它不必为了寻找最优解而穷尽所有可能解，因此其耗费时间少，一般可以快速得到满意的解，但得不到最优解。

5、舍弃已被覆盖的房子，可以将问题的规模逐步缩小，形成规模较小的子问题，而这些问题的求解与原问题的求解过程相同，因此本题属于分治法的算法思想。

（二）对于第二空，时间复杂度。

由于本题的算法过程，是依次与各个房子进行判断，当所有房子都被比较之后，则问题结束，因此时间复杂度与房子的个数相关，本问题的时间复杂度应该趋于现象，为 O(n)。

（三）对于第三空，关于对应序列（10，20，30，35，60，80，160，210，260，300）

1、第一轮放置：在第一座房子 x=10 的右侧 20 米处安装一个消防栓，可以覆盖 10，20，30，35 这 4 栋房子；

2、第二轮放置：去掉前 4 栋房子，在第 5 栋房子 x=60 的右侧 20 米处安装一个消防栓，可以覆盖 60、80 这 2 栋房子；

3、第三轮放置：去掉前面已覆盖的房子，在第 7 栋房子 x=160 的右侧 20 米处安装一个消防栓，只可以覆盖 160 这一栋房子；

4、第四轮放置：去掉前面已覆盖的房子，在第 8 栋房子 x=210 的右侧 20 米处安装一个消防栓，可以覆盖 210 这一栋房子；

5、第五轮放置：去掉前面已覆盖的房子，在第 9 栋房子 x=260 的右侧 20 米处安装一个消防栓，可以覆盖 260、300 这 2 栋房子；

6、房子全部覆盖完毕，因此共需安装 5 个消防栓。

（四）对于第四空，对于得到一个最优解是动态规划的特点，可以得到问题所有的最优解，是回溯法的特征。

使用 ADSL 接入 Internet，用户端需要安装（66）协议。

（66）A.PPP　　B.SLIP　　C.PPTP　　D.PPPoE

【答案】D

【解析】

ADSL Modem 上网拨号方式有 3 种，即专线方式（静态 IP）、PPPoA 和 PPPoE。

PPPoE（英语：Point-to-Point Protocol Over Ethernet），以太网上的点对点协议，是将点对点协议（PPP）封装在以太网（Ethernet）框架中的一种网络隧道协议。

PPTP（Point to Point Tunneling Protocol），即点对点隧道协议。该协议是在 PPP 协议的基础上开发的一种新的增强型安全协议，支持多协议虚拟专用网（VPN），可以通过密码验证协议（PAP）、可扩展认证协议（EAP）等方法增强安全性。可以使远程用户通过拨入 ISP、通过直接连接 Internet 或其他网络安全地访问企业网。

SLIP（Serial Line Internet Protocol，串行线路网际协议），该协议是 Windows 远程访问的一种旧工业标准，主要在 Unix 远程访问服务器中使用，现今仍然用于连接某些 ISP。

PPP（点到点协议）是为在同等单元之间传输数据包这样的简单链路设计的链路层协议。这种链路提供全双工操作，并按照顺序传递数据包。设计目的主要是用来通过拨号或专线方式建立点对点连接发送数据，使其成为各种主机、网桥和路由器之间简单连接的一种共通的解决方案。

下列命令中，不能用于诊断 DNS 故障的是（67）。

（67）A.netstat　　B.nslookup　　C.ping　　D.tracert

【答案】A

【解析】

netstat 是控制台命令，是一个监控 TCP/IP 网络的非常有用的工具，它可以显示路由表、实际的网络连接以及每一个网络接口设备的状态信息。netstat 用于显示与 IP、TCP、UDP 和 ICMP 协议相关的统计数据，一般用于检验本机各端口的网络连接情况。

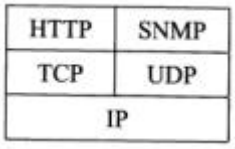
nslookup 是一个监测网络中 DNS 服务器是否能正确实现域名解析的命令行工具。

PING 命令常用于测试连通性，在此过程中可看出是直接 ping 的目标地址。

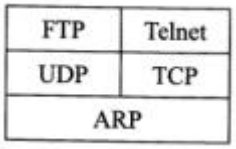
nslookup、ping、tracert 都可以加上一个主机域名作为其命令参数来诊断 DNS 故障，nslookup 还可以看到本地 DNS 服务器地址。Arp 命令是与 arp 记录有关，与 DNS 无关联。

以下关于 TCP/IP 协议和层次对应关系的表示中，正确的是（68）。

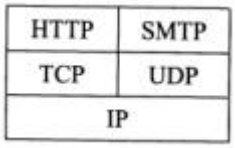
（68）A.



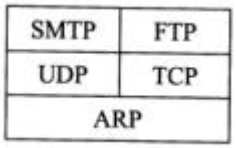
B.



C.



D.



【答案】A

【解析】

把 CSS 样式表与 HTML 网页关联，不正确的方法是（69）。

（69）A.在 HTML 文档的<head>标签内定义 CSS 样式

B.用@import 引入样式表文件

C.在 HTML 文档的<!-- -->标签内定义 CSS 样式

D.用<link>标签链接网上可访问的 CSS 样式表文件

【答案】C

【解析】

使用（70）命令可以释放当前主机自动获取的 IP 地址。

（70）A.ipconfig/all　　B.ipconfig/reload

C.ipconfig/release　　D.ipconfig/reset

【答案】C

【解析】

ipconfig/all 能为 DNS 和 WINS 服务器显示它已配置且所要使用的附加信息（如 IP 地址等），并且显示内置于本地网卡中的物理地址。

ipconfig/release 也只能在向 DHCP 服务器租用其 IP 地址的计算机上起作用。如果你输入 ipconfig /release，那么所有接口的租用 IP 地址便重新交付给 DHCP 服务器。

The project workbook is not so much a separate document as it is a structure imposed on the documents that the project will be producing anyway.

All the documents of the project need to be part of this （ 71 ）. This includes objectives ,external specifications , interface specifications , technical standards , internal specifications and administrative memoranda（备忘录）.

Technical prose is almost immortal. If one examines the genealogy （ 手册） of a customer manual for a piece of hardware or software , one can trace not only the ideas , but also many of the very sentences and paragraphs back to the first （72） proposing the product or explaining the first design. For the technical writer, the paste-pot is as mighty as the pen.

Since this is so, and since tomorrow's product-quality manuals will grow from today’ s memos, it is very important to get the structure of the documentation right. The early design of the project （73） ensures that the documentation structure itself is crafted, not haphazard. Moreover, the establishment of a structure molds later writing into segments that fit into that structure.

The second reason for the project workbook is control of the distribution of （74）. The problem is not to restrict information, but to ensure that relevant information gets to all the people who need it.

The first step is to number all memoranda, so that ordered lists of titles are available and h worker can see if he has what he wants. The organization of the workbook goes well beyond this to establish a tree-structure of memoranda. The （75） allows distribution lists to be maintained by subtree, if that is desirable.

（71）A.structure	B.specification	C.standard	D.objective
（72）A.objective	B.memoranda	C.standard	D.specification
（73）A.title	B.list	C.workbook	D.quality
（74）A.product	B.manual	C.document	D.information
（75）A.list	B.document	C.tree-structure	D.number

【答案】A B C D C

【解析】

项目工作手册不是单独的一篇文档，它是对项目必须产出的一系列文档进行组织的一种结果。

项目的所有文档都必须是该结构的一部分。这包括目标，外部规范说明，接口规范，技术标准，内部规范和管理备忘录（备忘录）。

技术说明几乎是必不可少的。如果某人就硬件和软件的某部分,去查看一系列相关的用户手册。他发现的不仅仅是思路,而且还有能追溯到最早备忘录的许多文字和章节,这些备忘录对产品提出建议或者解释设计。对于技术作者而言,文章的剪裁粘贴与钢笔一样有用。基于上述理由,再加上“未来产品”的质量手册将诞生于“今天产品”的备忘录,所以正确的文档结构非常重要。事先将项目工作手册设计好,能保证文档的结构本身是规范的,而不是杂乱无章的。另外,有了文档结构,后来书写的文字就可以放置在合适的章节中。使用项目手册的第二个原因是控制信息布。控制信息发布并不是为了限制信息,而是确保信息能到达所有需要它的人的手中项目手册的第一步是对所有的备忘录编号,从而每个工作人员可以通过标题列表来检索是否有他所需要的信息。还有一种更好的组织方法,就是使用树状的索引结构。而且如果需要的话,可以使用树结构中的子树来维护发布列表。

试题一

【说明】

某房产中介连锁企业欲开发一个基于 Web 的房屋中介信息系统，以有效管理房源和客户，提高成交率。该系统的主要功能是:

1.房源采集与管理。系统自动采集外部网站的潜在房源信息，保存为潜在房源。由经纪人联系确认的潜在房源变为房源，并添加出售/出租房源的客户。由经纪人或客户登记的出售/出租房源，系统将其保存为房源。房源信息包括基本情况、配套设施、交易类型、委托方式、业主等。经纪人可以对房源进行更新等管理操作。

2.客户管理。求租/求购客户进行注册、更新，推送客户需求给经纪人，或由经纪人对求租/求购客户进行登记、更新。客户信息包括身份证号、姓名、手机号、需求情况、委托方式等。

3.房源推荐。根据客户的需求情况（求购/求租需求情况以及出售/出租房源信息），向已登录的客户推荐房源。

4.交易管理。经纪人对租售客户双方进行交易信息管理，包括订单提交和取消，设置收取中介费比例。财务人员收取中介费之后，表示该订单已完成，系统更新订单状态和房源状态，向客户和经纪人发送交易反馈。

5.信息查询。客户根据自身查询需求查询房屋供需信息。

现采用结构化方法对房屋中介信息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。

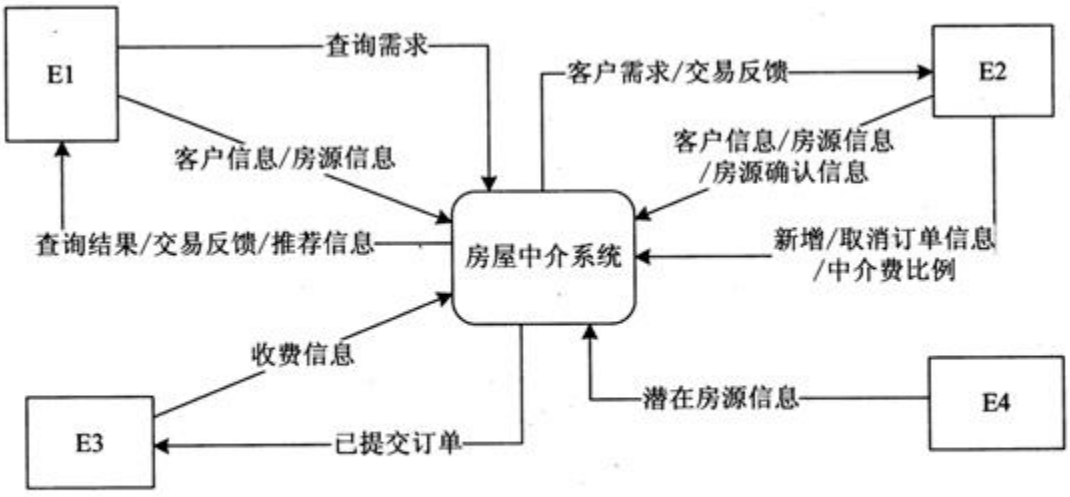


图 1-1 上下文数据流图

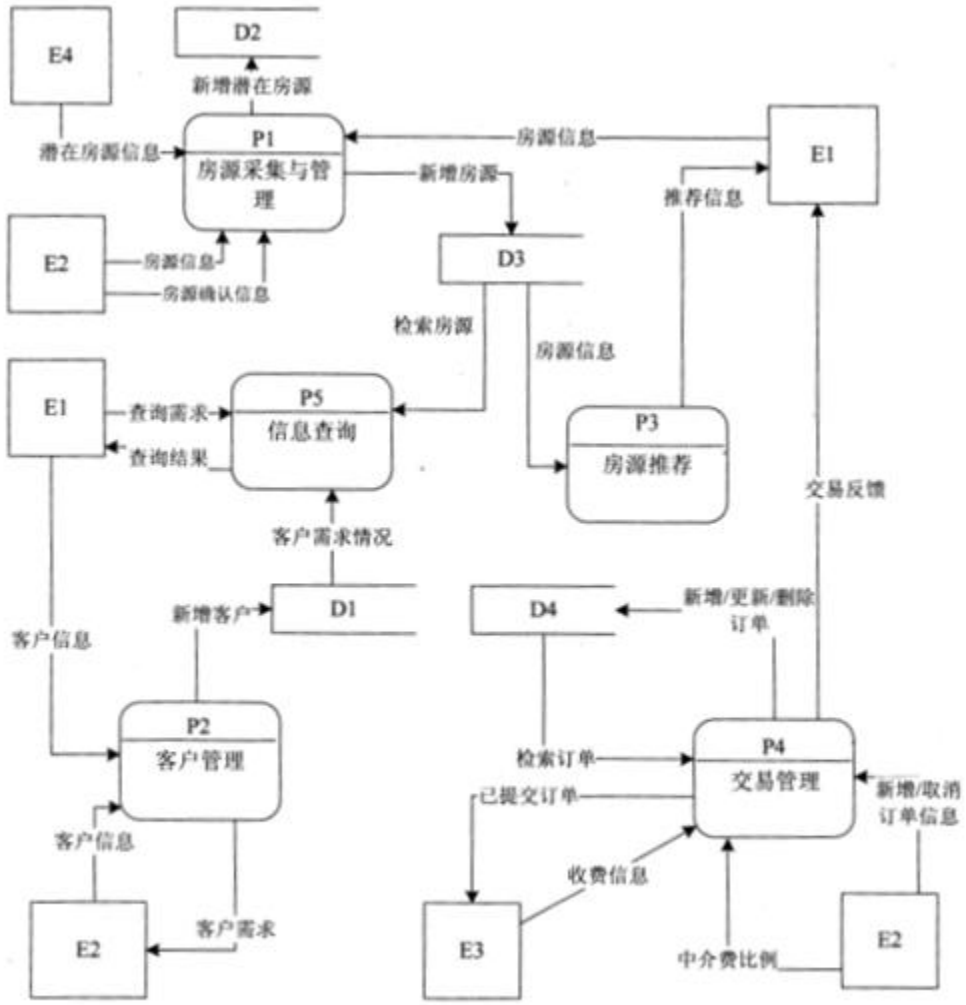


图 1-2 0 层数据流图

【问题 1】（4 分）

使用说明中的词语，给出图 1-1 中的实体 E1-E4 的名称。

【问题 2】（4 分）

使用说明中的词语，给出图 1-2 中的数据存储 D1-D4 的名称。

【问题】（3 分）

根据说明和图中术语，补充图 1-2 中缺失的数据流及其起点和终点。

【问题 4】（4 分）

根据说明中术语，给出图 1-1 中数据流"客户信息"、"房源信息"的组成。

答案：

【问题 1】

E1：客户；E2：经纪人；E3：财务人员；E4：外部网站

【问题 2】

D1：客户记录；D2：潜在房源记录；D3：房源记录；D4：订单记录

【问题 3】

缺失数据流如下：

1、交易反馈：起点-P4 交易管理，终点-E2

2、客户需求：起点-D1，终点-P3 房源推荐

3、房源状态：起点-P4 交易管理，终点-D3

【问题 4】

客户信息：身份证号，姓名，手机号，需求情况，委托方式。

房源信息：基本情况，配套设施，交易类型，委托方式，业主等。

解析：

【问题 1】

题干说明中，自动采集潜在房源信息，并且无反馈信息的为外界网站，即 1-1 中的 E4；

只有经纪人可以确认潜在房源，因此 E2 为经纪人；

系统只向客户推送推荐房源，因此 E1 为客户。

财务管理人员收取中介费用，因此 E3 为财务人员。

【问题 2】

对于新增潜在房源，保存为潜在房源，即 D2 为潜在房源记录；

对于新增房源保存为房源，即 D3 为房源记录；

对于新增客户应该保存为客户信息，因此 D1 为客户记录或客户信息表；

对于订单检索的对象应该为订单记录，即 D4 为订单记录。

试题二

【说明】

某集团公司拥有多个分公司，为了方便集团公司对分公司各项业务活动进行有效管理，集团公司决定构建一个信息系统以满足公司的业务管理需求。

【需求分析】

1.分公司关系需要记录的信息包括分公司编号、名称、经理、联系地址和电话。分公司编号唯一标识分公司信息中的每一个元组。每个分公司只有一名经理，负责该分公司的管理工作。每个分公司设立仅为本分公司服务的多个业务部门，如研发部、财务部、采购部、销售部等。

2.部门关系需要记录的信息包括部门号、部门名称、主管号、电话和分公司编号。部门号唯一标识部门信息中的每一个元组。每个部门只有一名主管，负责部门的管理工作。每个部门有多名员工，每名员工只能隶属于一个部门。

3.员工关系需要记录的信息包括员工号、姓名、隶属部门、岗位、电话和基本工资。其中，员工号唯一标识员工信息中的每一个元组。岗位包括:经理、主管、研发员、业务员等。

【概念模型设计】

根据需求阶段收集的信息，设计的实体联系图和关系模式（不完整）如图 2-1 所示:

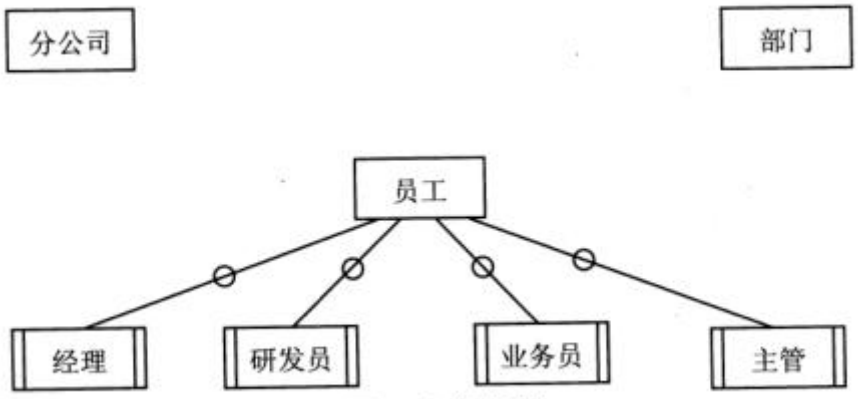


图 2-1 实体联系图

【关系模式设计】

分公司（分公司编号，名称，(a)，联系地址，电话）

部门（部门号，部门名称，(b)，电话）

员工（员工号，姓名（c)，电话，基本工资）

【问题 1】(4 分)

根据问题描述，补充 4 个联系，完善图 2-1 的实体联系图。联系名可用联系 1、联系 2、联系 3 和联系 4 代替，联系的类型为 1:1、1:n 和 m:n （或 1:1、1:*和*:*）。

【问题 2】（5 分）

根据题意，将关系模式中的空 （a）-（c） 补充完整。

【问题 3】（4 分）

给出"部门"和"员工"关系模式的主键和外键。

【问题 4】（2 分）

假设集团公司要求系统能记录部门历任主管的任职时间和任职年限，那么是否需要在数据库设计时增设一个实体?为什么?

答案：

【问题 1】

联系 1：分公司：经理，1：1

联系 2：分公司：部门，1：*

联系 3：部门：主管，1：1

联系 4：部门：员工，1：*

【问题 2】

（a）经理工号

（b）分公司编号，主管号

（c）隶属部门，岗位

【问题 3】

部门

主键：部门号；外键：分公司编号，主管号

员工

主键：员工号；外键：隶属部门

【问题 4】

不需要增加新的实体，对于任职情况，可以将部门与主管的联系单独形成关系模式，联系（部门号，主管工号，任职时间，任职年限），考虑到同一个员工可能会在不同的时间多次担任主管，因此，对于该关系模式，可以将（部门号，主管工号，任职时间）作为组合主键。

试题三

【说明】

社交网络平台 （SNS） 的主要功能之一是建立在线群组，群组中的成员之间可以互相分享或挖掘兴趣和活动。每个群组包含标题、管理员以及成员列表等信息。

社交网络平台的用户可以自行选择加入某个群组。每个群组拥有一个主页，群组内的所有成员都可以查看主页上的内容。如果在群组的主页上发布或更新了信息，群组中的成员会自动接收到发布或更新后的信息。

用户可以加入一个群组也可以退出这个群组。用户退出群组后，不会再接收到该群组发布或更新的任何信息。

现采用面向对象方法对上述需求进行分析与设计，得到如表 3-1 所示的类列表和如图 3-1 所示的类图。

表 3-1　类列表

类名	描述
SNSSubject	群组主页的内容
SNSGroup	社交网络平台中的群组（在主页上发布信息）
SNSObserver	群组主页内容的关注者
SNSUser	社交网络平台用户/群组成员
SNSAdmin	群组的管理员

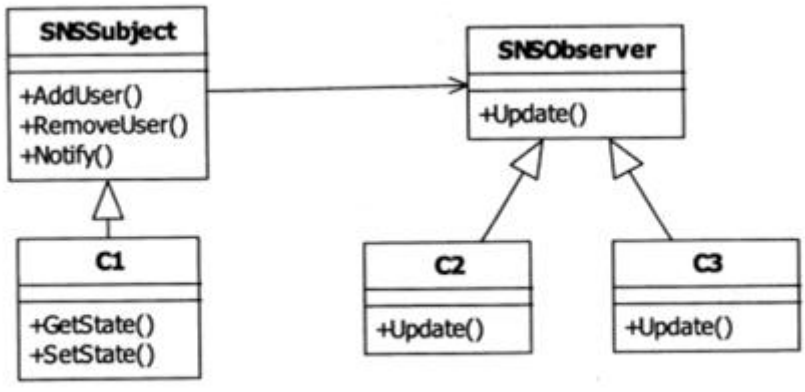


图 3-1　类图

【问题 1】（6 分）

根据说明中的描述，给出图 3-1 中 Cl C3 所对应的类名。

【问题 2】（6 分）

图 3-1 中采用了哪一种设计模式?说明该模式的意图及其适用场合。

【问题 3】（3 分）

现在对上述社交网络平台提出了新的需求:一个群体可以作为另外一个群体中的成员，例如群体 A 加入群体 B 。那么，群体 A 中的所有成员就自动成为群体 B 中的成员。若要实现这个新需求，需要对图 3-1 进行哪些修改? （以文字方式描述）

答案：

【问题 1】

C1：SNSGroup；C2：SNSUser；C3：SNSAdmin。

（其中 C2、C3 可以互换）

【问题 2】

采用的设计模式：观察者模式

意图：当被观察者（群组主页）发生改变时，可以通知所有的观察者（群组主页内容的关注者）随之改变，以达到联动的效果。

使用场合：观察者模式是行为型模式，定义对象间的一种一对多的依赖关系，当一个对象的状态发生改变时，所有依赖于它的对象都得到通知并自动更新。

【问题 3】

新增一个被观察者对象群组 B 的主页，对于观察者，新增一个方法，加入群组 B，加入之后，可以接收被观察者群组 B 的主页变动所发送的通知。

试题四

【说明】

给定一个字符序列 B=$b_1b_2\ldots b_n$，其中$b_i \in \{A, C, G, U\}$。B 上的二级结构是一组字符对集合 S=$\{(b_i,b_j)\}$，其中$i, j \in \{1,2,\ldots,n\}$，并满足以下四个条件：

（1）S 中的每对字符是(A,U),(U,A),(C,G)和(G,C)四种组合之一；

（2）S 中的每对字符之间至少有四个字符将其隔开，即 $i < j-4$；

（3）S 中每一个字符（记为 b_k）的配对存在两种情况：b_k 不参与任何配对；b_k 和字符 b_t 配对，其中 $t < k-4$；

（4）（不交叉原则）若(b_i,b_j)和(b_k,b_l)是 S 中的两个字符对，且 $i < k$，则 $i < k < j < l$ 不成立。

B 的具有最大可能字符对数的二级结构 S 被称为最优配对方案，求解最优配对方案中的字符对数的方法如下：

假设用 C(i,j)表示字符序列 $b_ib_{i+1}\ldots b_j$ 的最优配对方案（即二级结构 S）中的字符对数，则 C(i,j) 可以递归定义为：

$$C(i,j)=\begin{cases}\max\left(C(i,j-1), \max\left(C(i,t-1)+1+C(t+1,j-1)\right)\right) & \text{若}b_t\text{和}b_j\text{匹配且}i<j-4\\ 0 & \text{否则}\end{cases}$$

下面代码是算法的C语言实现，其中

n：字符序列长度

B[]：字符序列

C[][]：最优配对数量数组

【C 代码】

```
#include <stdio.h>
#include <stdlib.h>
#define LEN 100

/*判断两个字符是否配对*/
int isMatch(char a,char b){
    if((a == 'A' && b == 'U') || (a == 'U' && b == 'A'))
        return 1;
    if((a == 'C' && b == 'G') || (a == 'G' && b == 'C'))
        return 1;
    return 0;
}
```

```
/*求最大配对数*/
int RNA_2(char B[LEN], int n){
    int i, j, k, t;
    int max;
    int C[LEN][LEN] = {0};

    for(k = 5; k <= n - 1; k++){
        for(i = 1; i <= n - k; i++){
            j = i + k;
            ____(1)____;
            for(____(2)____; t <= j - 4; t++){
                if(____(3)____ && max < C[i][t - 1] + 1 + C[t + 1][j - 1])
                    max = C[i][t - 1] + 1 + C[t + 1][j - 1];
            }
            C[i][j] = max;
            printf("c[%d][%d] = %d--", i, j, C[i][j]);
        }
    }
    return ____(4)____;
}
```

【问题 1】（8 分）

根据题干说明，填充 C 代码中的空（1）-（4）。

【问题 2】（4 分）

根据题干说明和 C 代码，算法采用的设计策略为（5）

算法的时间复杂度为（6），（用 O 表示）。

【问题 3】（3 分）

给定字符序列 ACCGGUAGU ，根据上述算法求得最大字符对数为（7）

答案：

【问题 1】

（1）max=C[i][j-1]

（2）t=i

（3）isMatch(b[t],b[j])，或 isMatch(B[t],B[j])==1，或与其等价的形式

（4）max 或 C[i-1][j]

【问题 2】

采用的算法策略：动态规划法

时间复杂度：O(n3)

【问题 3】

最大字符对数：2

试题五

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】

某航空公司的会员积分系统将其会员划分为:普卡（Basic）、银卡（Silver）和金卡（Gold）三个等级。非会员（NonMember）可以申请成为普卡会员。会员的等级根据其一年内累积的里程数进行调整。描述会员等级调整的状态图如图5-1所示。现采用状态（State）模式实现上述场景，得到如图5-2所示的类图。

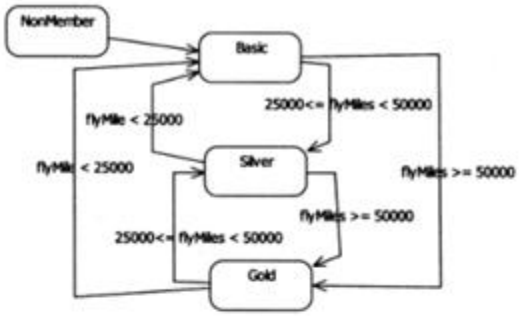


图5-1 会员等级调整状态图

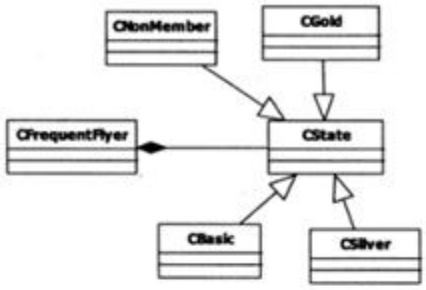


图5-2 状态模式类图

【C++代码】

```
#include <iostream>
using namespace std;
class FrequentFlyer; class CBasic; class CSilver; class CGold; class CNoCustomer; // 提前引用
class CState {
private:   int flyMiles;   // 里程数
public:
        ____(1)____;      // 根据累积里程数调整会员等级
};
class FrequentFlyer {
friend class CBasic;   friend class CSilver;     friend class CGold;
private:
     CState *state;  CState *nocustomer;     CState *basic;   CState *silver;    CState *gold;
     double flyMiles;
public:
     FrequentFlyer(){    flyMiles = 0;   setState(nocustomer);    }
     void setState(CState *state){  this->state = state;  }
     void travel(int miles) {
```

```
        double bonusMiles = state->travel(miles, this);
        flyMiles = flyMiles + bonusMiles;
    }
};
class CNoCustomer : public CState {    // 非会员
public:
    double travel(int miles, FrequentFlyer* context) {    // 不累积里程数
        cout << "Your travel will not account for points\n";    return miles;
    }
};
class CBasic : public CState {    // 普卡会员
public:
    double travel(int miles, FrequentFlyer* context) {
        if(context->flyMiles >= 25000 && context->flyMiles < 50000)
            _____(2)_____;
        if(context->flyMiles < 25000)    _____(3)_____;
        return miles + 0.5*miles;          // 累积里程数
    }
};
class CGold : public CState {    // 金卡会员
public:
    double travel(int miles, FrequentFlyer* context) {
        if(context-> flyMiles   >= 25000 && context-> flyMiles < 50000)
            _____(4)_____;
        if(context-> flyMiles < 25000)    _____(5)_____;
        return miles + 0.5*miles;          // 累积里程数
    }
};
class CSilver : public CState {   // 银卡会员
public:
    double travel(int miles, FrequentFlyer* context) {
        if(context-> flyMiles < 25000)
            context->setState(context->basic);
        if(context-> flyMiles >= 50000)
            context->setState(context->gold);
        return (miles + 0.25*miles);
    }
};
```

答案：

【问题 1】

（1）virtual double travel(int miles,FrequentFlyer* context)=0

（2）context->setState(context->silver)

（3）context->setState(context->gold)

（4）context->setState(context-> silver)

（5）context->setState(context->basic)

试题六

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】

某航空公司的会员积分系统将其会员划分为:普卡（Basic）、银卡(Silver)和金卡（Gold）三个等级。非会员 （NonMember）可以申请成为普卡会员。会员的等级根据其一年内累积的里程数进行调整。描述会员等级调整的状态图如图 6-1 所示 。现采用状态 （State） 模式实现上述场景，得到如图 6-2 所示的类图。

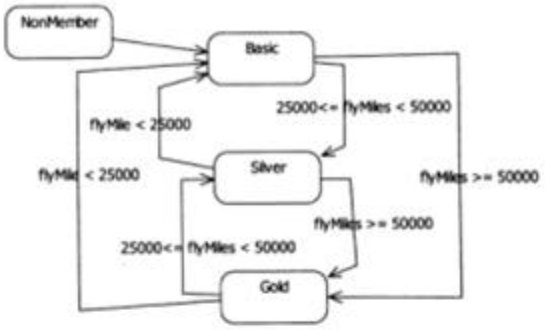


图 6-1 会员等级调整状态图

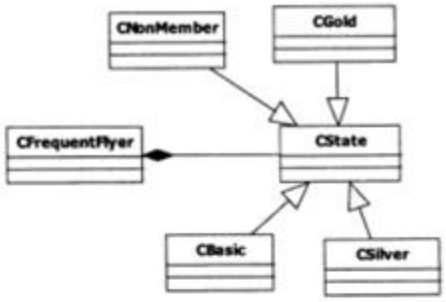


图 6-2 状态模式类图

【Java 代码】

```
import java.util.*;

abstract class CState {
    public int flyMiles;          // 里程数
    public _____(1)_____;  // 根据累积里程数调整会员等级
}
class CNoCustomer extends CState {      // 非会员
    public double travel(int miles, FrequentFlyer context) {
        System.out.println("Your travel will not account for points");
        return miles;              // 不累积里程数
    }
}
class CBasic extends CState {        // 普卡会员
    public double travel(int miles, FrequentFlyer context) {
        if(context.flyMiles >= 25000 && context.flyMiles < 50000)
            _____(2)_____;
        if(context.flyMiles >= 50000)
            _____(3)_____;
        return miles;
```

```
    }
}

class CGold extends CState {              // 金卡会员
    public double travel(int miles, FrequentFlyer context) {
        if(context.flyMiles >= 25000 && context.flyMiles < 50000)
            ____(4)____;
        if(context.flyMiles < 25000)
            ____(5)____;
        return miles + 0.5*miles;          // 累积里程数
    }
}
class CSilver extends CState {            // 银卡会员
    public double travel(int miles, FrequentFlyer context) {
        if(context.flyMiles <= 25000)
            context.setState(new CBasic());
        if(context.flyMiles >= 50000)
            context.setState(new CGold());
        return (miles + 0.25*miles);       // 累积里程数
    }
}

class FrequentFlyer {
    CState state;
    double flyMiles;
    public FrequentFlyer(){
        state = new CNoCustomer();
        flyMiles = 0;
        setState(state);
    }
    public void setState(CState state){      this.state = state;    }
    public void travel(int miles) {
        double bonusMiles = state.travel(miles, this);
        flyMiles = flyMiles + bonusMiles;
    }
}
```

答案：

【问题 1】

（1）abstract double travel(int miles,FrequentFlyer context)

（2）context.setState(new CSilver())

（3）context.setState(new CGold ())

（4）context.setState(new CSilver())

（5）context.setState(new CBasic())