**2018年上半年 上午考试答案解析：**

1 D 对阶是指将两个进行运算的浮点数阶码对齐的操作。对阶的目的是为使两个浮点数的尾数能够进行加减运算。首先求出两浮点数阶码的差，即n，将小阶码加上n，使之与大阶码相等，同时将小阶码对应的浮点数的尾数右移相应的位数n，以保证该浮点数的值不变。

2—3B C

阻塞：一般是指线程阻塞，线程在运行的过程中因为某些原因而发生阻塞，阻塞状态的线程的特点是：该线程放弃CPU的使用，暂停运行，只有等到导致阻塞的原因消除之后才回复运行。或者是被其他的线程中断，该线程也会退出阻塞状态，同时抛出InterruptedException。

中断：处理机处理程序运行中出现的紧急事件的整个过程.程序运行过程中，系统外部、系统内部或者现行程序本身若出现紧急事件，处理机立即中止现行程序的运行，自动转入相应的处理程序(中断服务程序)，待处理完后，再返回原来的程序运行，这整个过程称为程序中断;当处理机接受中断时，只需暂停一个或几个周期而不执行处理程序的中断，称为简单中断.中断又可分为屏蔽中断和非屏蔽中断两类。

把一个方法与其所在的类/对象关联起来叫做方法的绑定。绑定分为静态绑定（前期绑定）和动态绑定（后期绑定）。静态绑定（前期绑定）是指在程序运行前就已经知道方法是属于那个类的，在编译的时候就可以连接到类的中，定位到这个方法。动态绑定（后期绑定）是指在程序运行过程中，根据具体的实例对象才能具体确定是哪个方法。

静态绑定发生于数据结构和数据结构间，程序执行之前。静态绑定发生于编译期，因此不能利用任何运行期的信息。它针对函数调用与函数的主体，或变量与内存中的区块。动态绑定则针对运行期产生的访问请求，只用到运行期的可用信息。在面向对象的代码中，动态绑定意味着决定哪个方法被调用或哪个属性被访问，将基于这个类本身而不基于访问范围。

中断保存现场：指的是进入中断服务程序或子程序后，由于寄存器有限，主程序和中断服务程序或子程序中用到相同的寄存器，所以为防止冲突，在中断服务程序前或在子程序前用进栈指令保护那些可能受到冲突的寄存器，然后在返回前恢复。

4-5 D B 汉明码也利用了奇偶位校验的概念，通过在数据位后面增加一些比特，可以验证数据的有效性。利用一个以上的校验位，汉明码不仅可以验证数据是否有效，还能在数据出错的情况下指明错误位置。2P≥P+D+1，其中P代表汉明码的个数，D代表数据位的个数。D=32，所以P=6；

奇偶校验是一种添加一个奇偶位用来指示之前的数据中包含有奇数还是偶数个1的检验方式。如果在传输的过程中，有奇数个位发生了改变，那么这个错误将被检测出来(注意奇偶位本身也可能改变)。一般来说，如果数据中包含有奇数个1的话，则将奇偶位设定为1;反之，如果数据中有偶数个1的话，则将奇偶位设定为0。换句话说，原始数据和奇偶位组成的新数据中，将总共包含偶数个1. 奇偶校验并不总是有效，如果数据中有偶数个位发生变化，则奇偶位仍将是正确的，因此不能检测出错误。而且，即使奇偶校验检测出了错误，它也不能指出哪一位出现了错误，从而难以进行更正。数据必须整体丢弃并且重新传输。在一个噪音较大的媒介中，成功传输数据可能需要很长时间甚至不可能完成。虽然奇偶校验的效果不佳，但是由于他只需要一位额外的空间开销，因此这是开销最小的检测方式。并且，如果知道了发生错误的位，奇偶校验还可以恢复数据。 如果一条信息中包含更多用于纠错的位，且通过妥善安排这些纠错位使得不同的出错位产生不同的错误结果，那么我们就可以找出出错位了。在一个7位的信息中，单个数据位出错有7种可能，因此3个错误控制位就足以确定是否出错及哪一位出错了。

6 C 当流水线达到稳定时，经过最长流水段操作时间后，会出来一个新的产品。用总产品数除以对应时间。就是吞吐率。

7 D AUX接口（Auxiliary）是指音频输入接口，可以输出包括mp3在内的电子声频设备的音频（一般的耳机插孔），可通过车上的音响来输出这些设备内的音乐。

Telnet协议是TCP/IP协议族中的一员，是Internet远程登陆服务的标准协议和主要方式。它为用户提供了在本地计算机上完成远程主机工作的能力。在终端使用者的电脑上使用telnet程序，用它连接到服务器。终端使用者可以在telnet程序中输入命令，这些命令会在服务器上运行，就像直接在服务器的控制台上输入一样。可以在本地就能控制服务器。要开始一个telnet会话，必须输入用户名和密码来登录服务器。Telnet是常用的远程控制Web服务器的方法。

SSH 为 Secure Shell 的缩写，由 IETF 的网络小组（Network Working Group）所制定；SSH 为建立在应用层基础上的安全协议。SSH 是目前较可靠，专为远程登录会话和其他网络服务提供安全性的协议。利用 SSH 协议可以有效防止远程管理过程中的信息泄露问题。SSH最初是UNIX系统上的一个程序，后来又迅速扩展到其他操作平台。SSH在正确使用时可弥补网络中的漏洞。SSH客户端适用于多种平台。几乎所有UNIX平台—包括HP-UX、Linux、AIX、Solaris、Digital UNIX、Irix，以及其他平台，都可运行SSH。

8-9 B A 数字签名保证信息传输的完整性、发送者的身份认证、防止交易中的抵赖发生。

数字签名技术是将摘要信息用发送者的私钥加密，与原文一起传送给接收者。接收者只有用发送者的公钥才能解密被加密的摘要信息，然后用HASH函数对收到的原文产生一个摘要信息，与解密的摘要信息对比。如果相同，则说明收到的信息是完整的，在传输过程中没有被修改，否则说明信息被修改过，因此数字签名能够验证信息的完整性。

数字签名是个加密的过程，数字签名验证是个解密的过程。

10 D 内务内控管理，主要是为了管理内部网络，防止越权访问，以及内部泄露信息。

11 A

重放攻击(Replay Attacks)又称重播攻击、回放攻击或新鲜性攻击(Freshness Attacks)，是指攻击者发送一个目的主机已接收过的包，来达到欺骗系统的目的，主要用于身份认证过程，破坏认证的正确性。

它是一种攻击类型，这种攻击会不断恶意或欺诈性地重复一个有效的数据传输，重放攻击可以由发起者，也可以由拦截并重发该数据的敌方进行。攻击者利用网络监听或者其他方式盗取认证凭据，之后再把它重新发给认证服务器。从这个解释上理解，加密可以有效防止会话劫持，但是却防止不了重放攻击。重放攻击任何网络通讯过程中都可能发生。重放攻击是计算机世界黑客常用的攻击方式之一，它的书面定义对不了解密码学的人来说比较抽象。

拒绝服务(英文名称denial of service;DoS)是指通过向服务器发送大量垃圾信息或干扰信息的方式，导致服务器无法向正常用户提供服务的现象。

利用域名解析服务器不验证请求源的弱点，攻击者伪装成攻击目标域名向全世界数以百万计的域名解析服务器发送查询请求，域名服务器返回的数据要远大于请求的数据，导致目标遭受了放大数十倍的DDoS攻击。被利用的域名服务器因此每天会收到大量的恶意请求，它也不断的遭受较小规模的DDoS攻击。

数据截获，就是通过一个网络设备或软件，窃取通信双方的交流信息。

数据流分析，就是对网络中的流量信息等进行检测。

12 C 国务院于1991年6月4日发布了《计算机软件保护条例》。该条例指出:计算机软件是指计算机程序及有关文档。受保护的软件必须由开发者独立开发，即必须具备原创性，同时，必须是已固定在某种有形物体上而非存在于开发者的头脑中。新条例自2002年1月1日起施行。1991年6月4日国务院发布的《计算机软件保护条例》同时废止。

软件开发者的开发者身份权保护期不受限制。软件著作权的其他权利保护期为25年，截止于软件首次发表后第25年的12月31日，保护期满前，软件著作权人可以向软件登记机关申请续展25年，但保护期最长不超过50年。因继承或单位分立、合并等法律行为使著作权人主体发生合法变更时，不改变相应软件著作权的保护期。因依法签订使用权或使用权许可合同而转让有关权利时，转让活动的发生不改变有关软件著作权的保护期。当拥有软件著作权的单位终止或拥有软件著作权的公民死亡而无合法继承者时，除开发者身份权外，有关软件的其他各项权利在保护期满之前进入公有领域。

计算机软件所有人应向软件登记机构办理软件著作权登记。软件登记机构发放的登记证明文件，是软件著作权有效或者登记申请文件中所述事实确定的初步证明。

凡已办登记的软件，在软件权利发生转让活动时，受让方应当在转让合同正式签订后3个月内向软件登记管理机构备案，否则不能对抗第三者的侵权活动。中国藉的软件著作权人将其在中国境内开发的软件权利向外国人许可或转让时，应当报请国务院有关主管部门批准并向软件登记管理机构备案。

软件著作权人，是指依法享有软件著作权的自然人、法人或者其他组织。软件著作权自软件开发完成之日起产生。除法律另有规定外，软件著作权属于软件开发者，即实际组织开发、直接进行开发，并对开发完成的软件承担责任的法人或者其他组织;或者依靠自己具有的条件独立完成软件开发，并对软件承担责任的自然人。如无相反证据，在软件上署名的自然人、法人或者其他组织为开发者。

委托开发、合作开发软件著作权的归属及行使原则与一般作品著作权归属及行使原则一样，但职务计算机软件的著作权归属有一定的特殊性。自然人在法人或者其他组织中任职期间所开发的软件有下列情形之一的，该软件著作权由该法人或者其他组织享有，该法人或者其他组织可以对开发软件的自然人进行奖励:

(1)针对本职工作中明确指定的开发目标所开发的软件;

(2)开发的软件是从事本职工作活动所预见的结果或者自然的结果;

(3)主要使用了法人或者其他组织的资金、专用设备、未公开的专门信息等物质技术条件所开发并由法人或者其他组织承担责任的软件。

用户购买了具有版权的软件，仅仅有了使用权，想想正版的windows系统，只能安装一次。

13 B 在12中有解释

14 C 在12中有解释

15 A 数据字典有以下4类条目：数据流，数据项，数据存储和基本加工。

16 B 解析：if(金额小于5000元){}

slse if(欠款时间大于60天){}

slse if(欠款时间小于60天){}

17-18 D C 关键路径为A-B-D-G-I-K-L 从A到H共需要12天(只有一条路径)，从A到L最长为22天，在不影响H到L的前提之下，EH可以推迟22-12-6=5天。同理IJ的松弛时间为3天。

19 C ＣＯＣＯＭＯⅡ中规模表示为源代码千行数（ＫＳＬＯＣ）。常用的方法有工作分解结构、类比评估技术、Ｐａｒｋｓｏｎ法则、专家判定技术、功能点分析法等。其中功能点分析法　是基于数学理论、适用于项目的各个阶段，是ＣＯＣＯＭＯ　Ｉ１提倡的一种方法。有三种不同的规模估算选择：对象点，功能点和代码行。

20AProlog(ProgramminginLogic的缩写)是一种逻辑编程语言。Python,Java/C++都是一种面向对象的解释型计算机程序设计语言，Lisp是一种函数式编程语言。

21C在编译方式下，机器上运行的是与源程序等价的目标程序，源程序和编译程序不在参与目标程序的执行过程。在解释方式下，解释程序和源程序(或其某种等价表示)要参与到程序的运行过程当中，运行程序的控制权在解释程序中。

22B后缀表达式就是二叉树的后序遍历。左树→右树→中树的顺序。

23-24CD P操作是S自减一，S≥0时，继续执行；V是S自加一，S>0，继续执行。

25A 目录文件是长度固定的记录式文件。大多数操作系统如UNIX，DOS采用多级目录机构 ，称为树型目录结构。 从根目录出发到任一非叶结点或树页结点都有且只有一条路径。系统为用户提供一个目前使用的工作目录，称为当前目录。 目录分解法：将目录项分为：名号目录项，基本目录项。 目录文件也分为名号目录文件和基本目录文件。 文件存取控制通过文件的共享，保护和保密三方面体现。 文件的共享是一个文件可以允许多个用户共同使用。空闲块也就是用户没有用到的部分。用户程序和用户数据对系统的影响相对来说比较小。

26D 这个题比较好理解，硬件设备和用户程序之间通过系统交互，而设备驱动程序就相当于设备的接口。中断程序是硬件处理程序。

27-28CB 这里的全文件名是指在根目录下如何打开这个文档，就是绝对路径加上文档全名。

绝对路径：是从盘符开始的路径；相对路径：是从当前路径开始的路径

29 D 对于增量模型：优点1)第一个可交付版本所需要的成本和时间是较少的，从而可减少开发由增量表示的小系统承担的风险2)由于很快发布第一个版本，因此可以减少用户需求的变更3)允许增量投资，即在项目开始时可以仅对一个或两个增量投资

缺点：1)如果没有对用户的变更要求进行规划，那么产生的初始增量可能会造成后来增量的不稳定2)如果需求不像早期思考的那样稳定和完整，那么一些增量就可能需要重新开发，重新发布3)由于进度和配置的复杂性，可能会增大管理成本，超出组织的能力。

30A 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)：其共性目标集中于已管理的过程的制度化。

CL3(已定义级的)：其共性目标集中于已定义的过程制度化。

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。

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

31B 辅助软件维护过程中的活动的软件称为“软件维护工具”，它辅助维护人员对软件代码及其文档进行各种维护活动。软件维护工具主要有：1、版本控制工具；2、文档分析工具；3、开发信息库工具；、4、逆向工程工具；5、再工程工具；6、配置管理支持工具。

32C 概要设计文档主要包括：软件体系总体结构设计；数据结构及数据库设计。

33D耦合度，是对模块间关联程度的度量。耦合的强弱取决于模块间接口的复杂性、调用模块的方式以及通过界面传送数据的多少。模块间的耦合度是指模块之间的依赖关系，包括控制关系、调用关系、数据传递关系。模块间联系越多，其耦合性越强，同时表明其独立性越差(降低耦合性，可以提高其独立性)。软件设计中通常用耦合度和内聚度作为衡量模块独立程度的标准。划分模块的一个准则就是高内聚低耦合。

34-35AB覆盖测试就是要使测试用例可以走完所有的之路，也就是所有的语句都要执行一遍。在本题中，至少需要两个测试用例就可以啦；计算环形复杂度：

（1）流图中的区域数等于环形复杂度。

（2）流图G的环形复杂度V(G)=E-N+2，其中，E是流图中边的条数，N是结点数。

（3）流图G的环形复杂度V(G)=P+1，其中，P是流图中判定结点的数目。

一般采用方法（2）。边的条数指流程线，结点数为各种框的总和

36C 归纳法：是指从测试所暴露的问题出发，收集所有正确或不正确的数据，分析他们之间的关系，提出假想的错误原因，用这些数据来证明或反驳，从而查出错误所在。

试探法：调试人员分析错误的症状，猜测问题所在的位置，利用在程序中设置输出语句，分析寄存器，存储器的内容等手段获得错误的线索，一步步地试探和分析错误的所在。这种方法效率低，适合结构比较简单的程序。

回溯法：调试人员从发现错误的位置开始，人工沿着程序的控制流程往回跟踪代码，直到找出错误根源为止。这种方法适合于小型程序，对于大规模程序，由于其需要回溯的路径太多而不可操作。

对分查找法：这种方法主要用于缩小错误范围，如果已经知道程序中的变量在若干位置的正确取值，可以在这些位置上给这些变量以正确值，观察程序运行的输出结果，如果没有发现问题，则说明赋予变量一个正确值开始到输出结果之间程序没有错误，问题可能在除此之外的程序中，否则错误就在所观察的这部分程序中，对含有错误的程序段再使用这种方法，直接把故障范围缩小到比较容易诊断为止。

演绎法：根据测试结果，列出所有可能的错误；分析已有的数据，排除不可能和彼此矛盾的原因；对其余的原因，选择可能性最大的，利用已有的数据完善该假设，使假设更具体；用假设来解释所有的原始测试结果，如果能解释这一切，则假设得以证实，也就找出错误，否则，要么是假设不完备或不成立，要么有多个错误同时存在，需要重新分析，提出新的假设知道发现错误为止。

37A对象的状态一般都是指属性，对象的行为一般是指操作方法。

38D 39C 消息传递是指消息在两个不同应用程序或对象之间内容的传递。

类型检查是指对数据以及对象的类型进行检查，主要用于排错。

静态绑定是在编译时绑定；动态绑定是在运行过程中绑定。

40D 类型转换是指把一种数据类型转换成另外一种数据类型。

映射：类似于函数，是指一种一对一的关系。

多态：同一消息可以调用多种不同种类的对象的方法，

41C 42A 43D

类图：就是举了一个例子，有父类，有子类。组件图：各个组件之间的依赖关系。通信图：有路径有序号，部署图：面向对象的物理建模，是静态部署。本题图中有路径（带箭头）有序号

通信图的好处就是方便观察对象之间的消息流及其顺序。

GPS Location很明显表示位置的含义。

44D 45A 46B 47A

观察者：类对外扩展开放，对修改关闭。不同的Web应用，修改是肯定有的，故不可以；

访问者：它和观察者一样都是行为设计模式，它主要表示一个作用于某对象结构中的各个元素的操作。

策略：多种不同的方案切换，来解决问题。

桥接：继承树的拆分。抽象与实现相分离，各自独立变化。

48A a\*表示a可以出现0次或多次。

49B 上下文无关文法是程序设计语言所使用的语法。它的特点是同样的字符串在不同的语境下，意思不变。满足上下文无关文法的语言便于计算机识别和处理。我们已经介绍过，语言是语句的集合，而语句是通过产生式定义的。上下文无关文法要求产生式的左部有且仅有一个非终结符。

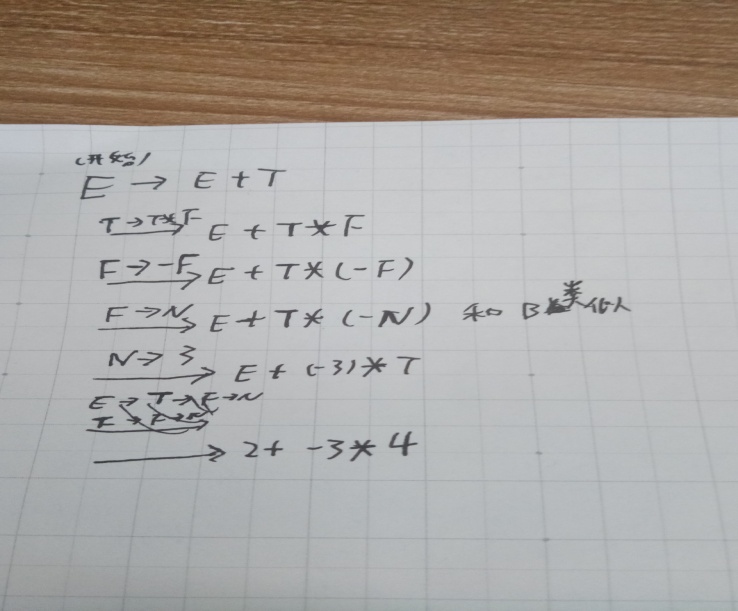
“|”表示“或”，所以

产生式E→T|E+T是产生集合：E→T E→E+T

产生式T→F|T\*F是产生集合：T→F T→T\*F

产生式F→-F|N是产生集合： F→-F F→N

同理还有N→0|1|2|3|4|5|6|7|8|9的



50C 语法制导翻译：基于属性文法的处理过程，对单词符号串进行语法分析，构造语法分析树，然后根据需要构造属性依赖图，遍历语法树并在语法树的各结点处按语义规则进行计算。

51B 伪传递规则：若X→Y在R上成立，且WY→Z，则XW→Z。

52A 53C π是投影运算，选出某个表中的若干列。这里是选出2 4 6 7列；

σ是选择运算，选出满足某些条件的行。这里是选出第二行中的元素小于第七行的；

▷◁是自然连接。等值连接去掉重复的行后就是啦。这里去掉一个B一个C就可以啦。

在（52）中填最后要选出的列。B两者都有所以要标明不可省。在（53）中添加条件。

54A 55D

在（54）处声明此function函数最终要返回的数据的数据类型。

在（55）处声明一个变量，用来存放数据。

56B Data Extraction：：数据抽取；

联机分析处理OLAP是一种软件技术，它使分析人员能够迅速、一致、交互地从各个方面观察信息，以达到深入理解数据的目的。它具有FASMI(Fast Analysis of Shared Multidimensional Information)，即共享多维信息的快速分析的特征。其中F是快速性(Fast)，指系统能在数秒内对用户的多数分析要求做出反应；A是可分析性(Analysis)，指用户无需编程就可以定义新的专门计算，将其作为分析的一部 分，并以用户所希望的方式给出报告；M是多维性(Multi—dimensional)，指提供对数据分析的多维视图和分析；I是信息性(Information)，指能及时获得信息，并且管理大容量信息。

On-Line Transaction Processing联机事务处理过程(OLTP)也称为面向交易的处理过程，其基本特征是前台接收的用户数据可以立即传送到计算中心进行处理，并在很短的时间内给出处理结果，是对用户操作快速响应的方式之一。

ETL，是英文 Extract-Transform-Load 的缩写，用来描述将数据从来源端经过抽取（extract）、交互转换（transform）、加载（load）至目的端的过程。ETL一词较常用在数据仓库，但其对象并不限于数据仓库。

ETL是构建数据仓库的重要一环，用户从数据源抽取出所需的数据，经过数据清洗,最终按照预先定义好的数据仓库模型，将数据加载到数据仓库中去。

57A 循环链表是一种形式的链式存贮结构。它的特点是表中最后一个结点的指针域指向头结点，整个链表形成一个环。

58C 举几个例子就可以啦

59D 顺序存储二叉树就是先把二叉树填满成完全二叉树。在中序遍历，这时最开始的二叉树中元素的位置就是存储位置啦。

60B 散列表（Hash table，也叫哈希表），是根据关键码值(Key value)而直接进行访问的数据结构。也就是说，它通过把关键码值映射到表中一个位置来访问记录，以加快查找的速度。这个映射函数叫做散列函数，存放记录的数组叫做散列表。给定表M，存在函数f(key)，对任意给定的关键字值key，代入函数后若能得到包含该关键字的记录在表中的地址，则称表M为哈希(Hash）表，函数f(key)为哈希(Hash) 函数。

61A 无向图就是指图中的每一条边均是无方向的。

领接矩阵：逻辑结构分为两部分：V和E集合。因此，用一个一维数组存放图中所有顶点数据；用一个二维数组存放顶点间关系（边或弧）的数据，这个二维数组称为邻接矩阵。邻接矩阵又分为有向图邻接矩阵和无向图邻接矩阵。

深度优先遍历，是图论中的经典算法。其利用深度优先搜索算法可以产生目标图的相应拓扑排序表，利用拓扑排序表可以方便的解决很多相关的图论问题，如最大路径问题等等。

62A 63C 64C 65B

**分治法**（归并排序，最大子和段问题）基本思想：将一个难以直接解决的大问题分解成一些规模较小的相同问题，以便各个击破，分而治之。

如规模为n的问题可分解成k个子问题，1＜k≤n，这些子问题互相独立且与原问题相同。分治法产生的子问题往往是原问题的较小规模。

步骤：（1）分解：将原问题分解成一系列子问题。

（2）求解：递归地求解各个子问题。若子问题足够小，则直接求解。

（3）合并：将子问题的解合并成原问题的解。

**动态规划法**（0-1背包问题，最长公共子序列问题；寻找最优解）基本思想：将带求解问题分解成若干个子问题，先求解子问题，然后从这些子问题的解得到原问题的解。

经分解得到的子问题往往不是独立的，在过程当中，可以用一个表来记录所有已解决的子问题的答案，不管该子问题以后是否被用到，只要它被计算过，就将其结果填入表中。

步骤：（1）找出最优解的性质，并刻画其结构特征；

（2）递归地定义最优解的值；

（3）以自底向上的方式计算出最优值；

（4）根据计算最优值时得到的信息，构造一个最优解。

**贪心算法**（活动选择，背包问题）：当前每一步都是最优的，是当前最好的选择，但不一定是最优解。

**回溯法**（0-1背包，n皇后问题）：

66B中继器只是简单的信号放大，属于物理层设备。交换机一般属于数据链路层，但然现在也有三层交换机是网络层，不过不特殊说明一般都是指数据链路层。路由器属于网络层。网桥和交换机类似。

67B 68C

http://www.123.com.cn/a/b.html

各部分意义如下：

http或者https 超文本传输协议（HTTP，HyperText Transfer Protocol)是互联网上应用最为广泛的一种网络协议。所有的WWW文件都必须遵守这个标准。

WWW是环球信息网的缩写,这里可以理解位子域名，也可以换成任意想用的名字

domain.com.cn是网站域名

a是网站的子目录

b.html是网页

69C mattrib为mtools工具指令，模拟MS-DOS的attrib指令，可变更MS-DOS文件的属性；

Modify 意思是“更改（内容），“或者“写入”。当更改了一个文件的内容的时候，此文件的modify的时间记录会被更新。用ls -l看到的文件时间是最近一次modify的时间。modify的行为是三个行为中最有影响力的行为，它发生以后，会使文件的access记录与change记录也同时得到更新。对于目录也是如此。

Linux/Unix 的文件中利用 chmod 可以藉以控制文件如何被他人所调用。

Change 改变（状态或属性）。对一个文件或者目录作mv、chown、chgrp操作后，它的Change时间记录被更新，change时间会受到modify行为的影响。用ls -lc看到的文件时间是最近一次change的时间。

70C 一、主机向本地域名服务器的查询一般都是采用递归查询。所谓递归查询就是：如果主机所询问的本地域名服务器不知道被查询的域名的IP地址，那么本地域名服务器就以DNS客户的身份，向其它根域名服务器继续发出查询请求报文(即替主机继续查询)，而不是让主机自己进行下一步查询。因此，递归查询返回的查询结果或者是所要查询的IP地址，或者是报错，表示无法查询到所需的IP地址。

二、本地域名服务器向根域名服务器的查询的迭代查询。迭代查询的特点：当根域名服务器收到本地域名服务器发出的迭代查询请求报文时，要么给出所要查询的IP地址，要么告诉本地服务器：“你下一步应当向哪一个域名服务器进行查询”。然后让本地服务器进行后续的查询。根域名服务器通常是把自己知道的顶级域名服务器的IP地址告诉本地域名服务器，让本地域名服务器再向顶级域名服务器查询。顶级域名服务器在收到本地域名服务器的查询请求后，要么给出所要查询的IP地址，要么告诉本地服务器下一步应当向哪一个权限域名服务器进行查询。最后，知道了所要解析的IP地址或报错，然后把这个结果返回给发起查询的主机。

71-75DCACC

●创建项目进展的清晰地图是重要的第一步。它可以让你识别风险，澄清目标，并确定项目是否有意义。唯一比发布计划更重要的是不要太认真。

发布计划正在为您的Web项目创建一个游戏计划（71）您认为您希望自己的网站是什么。该计划是向公众，合作伙伴或内部发布的网站的内容，设计元素和功能的指导。它也（72）项目需要多长时间，花多少钱。这个计划不是一个功能性的（73），它详细定义了项目或者产生了一个可以带到银行的预算。

基本上你使用一个发布计划对项目（74）进行初步的健全性检查并且有价值。发布计划是有用的路线图，但不要将它们视为州际公路系统的指南。相反，把它们想象成早期探险家所使用的（75） - 半条传言和猜测，半条希望和期望。

有一个项目的目标地图是一个好主意。

**2018年上半年 下午考试参考答案及解析**

【参考答案】

【问题1】E1：客户； E2：医生； E3：主管

【问题2】D1：通用信息表；D2：预约表；D3：医生列表；D4：出诊时间表；D5：药品库

【问题 3】

起点 终点 名称

P2 D4 删除解聘医生的出诊时间

P2 D2 删除解聘医生的预约数据

D5 P4 查询相关药品库存信息

P4 D2 更新预约表

D3 P5 医生 或 医生信息

【问题 4】

“预约处理”可以分解为子加工：安排出诊时间、就诊预约。

在建模数据流图时应保持父图(图 1-1)与子图(图 1-2)的平衡，即父图中某加工的输入输出数据流必须与它的子图的输入输出数据流在数量和名字上相同，但如果父图的一个输入(或输出)数据流对应于子图中几个输入(或输出)数据流，而子图中组成这些数据流的数据项全体正好是父图中的这一个数据流，那么它们仍然算是平衡的。

**【试题解析】**

本题考察面向结构化软件开发方法学中需求分析阶段使用的数据流图(DFD 图)。作答时，一定要现把题目中“功能部分”大致阅读一遍，画出关键功能性动词，以及以表或库结尾的名词，然后从图中查找题目中的对应句，做题。

【问题 1】需要填写外部实体，外部实体为不属于软件本身但是又与当前软件有交互关系的外部的人、软件、硬件、组织结构、数据库系统等，在作答时需要细心地对每一个阅读到的外部实体(一般为名词)高度重视；

【问题 2】考察数据存储文件，这需要对阅读到的“...文件”或“...表”等能够存储数据的媒介词汇高度重视。

【问题 3】不仅仅通过阅读文字描述来作答，同时也要使用父图与子图的数据守恒原则进行作答。本题的文字描述中，共阐述了(1)～(5)共 5个系统必须要实现的功能模块，这些功能(动词描述)对应数据流图中的加工 P1～P5。由于需要找出图 1-2 中缺少的数据流，故建议边阅读文字描述，边对照图 1-2 以便找出其中缺少的数据流。

从描述"(1)通用信息查询(对应加工 P1)。客户提交通用信息查询请求(可知实体‘客户’有数据流‘通用信息查询请求’流向加工‘通用信息查询’)，查询通用信息表(可知存储文件‘通用信息表’有数据流‘通用信息’流向加工‘通用信息查询’)，返回查询结果(可知加工‘通用信息查询’有数据流‘查询结果’流向实体‘客户’)。"，并对照图 1-2中 P1 加工和 E1 实体处可知 E1 为实体"客户"，D1 为“通用信息表”。此处通过对比，未发现缺少的数据流！

从描述“(2)医生聘用(对应加工 P2)。医生提出应聘/辞职申请(可知实体‘医生’有数据流‘应聘/辞职请求’流向加工 P2)，交由主管进行聘用/解聘审批(可知加工 P2 有数据流“聘用申请”和“解聘申请”流向实体‘主管’，以便主管审批)，更新医生表(可知加工 P2 有数据流流向存储文件‘医生表’，即 D3)，并给医生反馈聘用/解聘结果(可知加工 P2有对应的数据流‘聘用反馈’和‘解聘反馈’流向实体‘医生’);删除解聘医生的出诊安排(可知应该有一条数据流由 P2流向类似‘出诊安排表’的存储文件，此处缺少此数据流，打个标记，待确定‘出诊安排表’类似的文件后再补上)”。阅读到此，可确定 E2 为“医生”，E3 为“主管”，D3 为“医生列表”，且找到一条缺少的数据流。

阅读描述“(3)预约处理(对应加工 P3)。医生安排出诊时间(对应图中 E2 流向 P3 的数据流‘出诊时间’)，存入医生出诊时间表(对应图中 P3 流向 D4 的数据流‘新增出诊时间’，即 D4 为‘出诊时间表’);根据客户提交的预约查询请求 (对应 E1 至 P3 的数据流‘预约查询请求’)，查询在职医生(对应 D3 至 P3 的数据流‘在职医生列表’)及其出诊时间(对应 D4 至 P3 的数据流‘所需出诊时间’)等预约所需数据并返回(对应 P3 至 E1 的数据流‘预约所需数据’);创建预约，提交预约请求(对应 E1 至 P3 的数据流‘预约请求’)，在预约表中新增预约记录(对应 P3 至 D2 的数据流‘新预约’，即 D2为‘预约表’)，更新所约医生出诊时间(对应 P3 至 D4 的数据流‘新增出诊时间’)并给医生发送预约通知(对应 P3 至 E2的数据流‘预约通知’);给客户反馈预约结果(对应 P3 至 E1 的数据流‘预约反馈’)。”。到此得出 D2 为预约表，D3 为 “医生表”，D4 为“出诊时间表”。根据本段的分析加上之前标记缺少的数据流，可分析出缺少的数据流有两条，一条由P2 流向 D4，名称为“删除解聘医生的出诊时间”，另外一条由 P2 流向 D2，名称为“删除解聘医生的预约数据”。

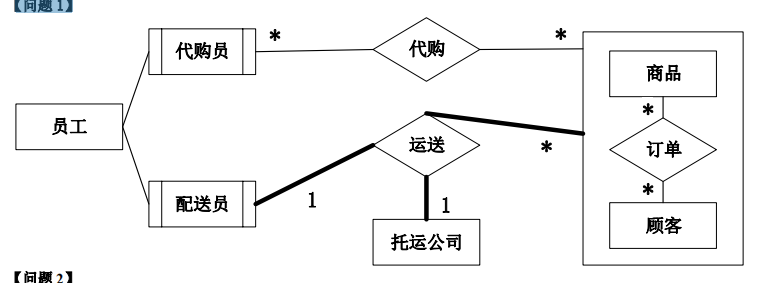
阅读描述“(4)药品管理(对应加工 P4)。医生提交处方(对应 E2 至 P4 的数据流‘处方’)，根据药品名称从药品数据中查询相关药品库存信息(可知存储文件‘药品库’应有数量流‘查询相关药品库存信息’流向 P4，图中未发现)，开出药品，更新对应药品的库存(对应 P4 至 D5 的数据流‘更新的药品库存信息’，知 D5 为‘药品库’)以及预约表中的治疗信息(可分析出 P4 应有数据流‘更新预约表’流向 D2，图中未发现);给医生发送“药品已开出”反馈(对应 P4 至 E2 的数据流‘药品已开出反馈’数据流)。”。到此，可分析出 D5 为“药品库”，且缺少 P4 至 D2 的数据流“更新预约表”和 D5 流向 P4 的数据流“查询相关药品库存信息”。

阅读描述“(5)报表创建(对应加工 P5)。根据主管提交的报表查询请求(报表类型和时间段)(对应 E3 至 P4 的数据流 ‘报表查询请求’)，从预约数据(对应 D2 至 P5 的数据流‘预约数据’)、通用信息(对应 D1 至 P5 的数据流‘通用’)、药品库存数据(对应 D5 至 P5 的数据流‘药品库存数据’)、医生(对应 D3 至 P5 的数据流‘医生信息’，缺少)以及医生出诊时间(对应 D4 至 P5 的数据流‘出诊时间’)中进行查询，生成报表返回给主管(对应 P5 至 E3 的数据流‘报表’)。”。通过本段分析，发现缺少 D3 至 P5 的一条数据流，名称为“医生信息”。通过仔细阅读“(3)预约处理……给客户反馈预约结果”段落，其中“医生安排出诊时间，存入医生出诊时间表;”完全属于医生使用的功能“安排出诊时间”，而“根据客户提交的预约查询请求，查询在职医生及其出诊时间等预约所需数据并返回;创建预约，提交预约请求，在预约表中新增预约记录，更新所约医生出诊时间并给医生发送预约通知;给客户反馈预约结果。”属于客户的“就诊预约”功能，故“预约处理”可以分解为“安排出诊时间”和“就诊预约”两个子加工。在本题的分层数据流图中，图 1-1 是父图，图 1-2 是子图，在建模数据流图时应保持父图与子图的平衡，即父图中某加工的输入输出数据流必须与它的子图的输入输出数据流在数量和名字上相同，但如果父图的一个输入（或输出）数据流对应于子图中几个输入（或输出）数据流，而子图中组成这些数据流的数据项全体正好是父图中的这一个数据流，那么它们仍然算是平衡的。

试题二 参考答案及解析

【参考答案】

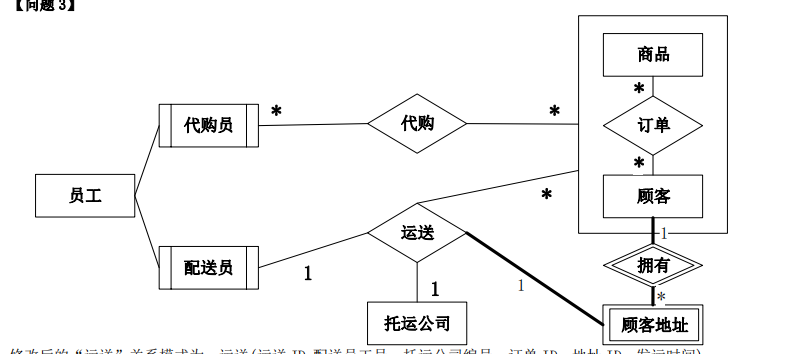
【问题 1】



【问题 2】

(a)运送方式、顾客编号、商品条码 (b)订单 ID

【问题 3】



修改“运送”模式:运送(运送 ID,配送员工号，托运公司编号，订单ID，地址ID，发运时间)。

【试题解析】

下午试题二的解题方法，建议为先阅读问题，勾画出关键字，再去边仔细阅读文字描述，边看图、边看关系模式边作答。

根据文字描述“(5)顾客登录系统之后…顾客需要在订单里指定运送方式：空运或海运…”知关系模式“订单”缺少一属性“运送方式”，又由于将 E-R 图转换为关系模式时，E-R 图中多对多的联系“订单”转换的关系二维表中需要将两端的实体“商品”和“顾客”的主键字段加入对应的二维表，故关系模式“订单”中还缺少“顾客编号”和“商品条码”两个属性，以实现关系模式的参照完整性,故空(b)为“运送方式、顾客编号、商品条码”。

根据文字描述“(6)代购员根据顾客的订单在超市采购对应商品，一份订单所含的多个商品可能由多名代购员从不同超市采购。”知 E-R 图中“订单”与“代购员”之间是多对多的联系，此联系即为“代购”，将联系“代购”转换为关系模式时，为了实现参照完整性，需要将两端的主键字段加入对应的关系模式，故关系模式“代购”中缺少订单的主键“订单ID”，空(a)为“订单 ID”。

阅读文字描述“(7)采购完的商品交由配送员根据顾客订单组合装箱(说明一个订单的所有商品整体打包运送，一个订单一个包一个配送员足够)，然后交给托运公司运送。托运公司按顾客订单核对商品名称和数量(说明一个订单一个包对应一个托运公司)，然后按顾客的地址进行运送。”可知，运送与实体“托运公司”、子实体“配送员”、“订单”之间有关联关系，配送员、托运公司、订单三者之间的关系为“1:1：\*”，补充后的图 2-1 请参加【问题 1】的参考答案。

为方便顾客，允许顾客在系统中保存多组收货地址，即属性“收货地址”变成了实体“顾客”的多值属性，为了避免数据冗余而带来的数据表的存储异常问题，可新增一弱实体“顾客地址”，其对强实体“顾客”有存在依赖关系，弱实体用双边框矩形表示(在 E-R 中的属性为“地址编号，收货地址”)，其与强实体之间的联系用双线菱形表示，同时为了保证每个商品订单运送地址的唯一性，需要在 E-R 图中补充联系“运送”与“收货地址”的连接关系。补充后的图 2-1 参见

【问题 3】的参考答案。通过分析，不难看出，关系模式“运送”应加入一字段“地址编号”，修改后的关系模式为“运送(运送 ID,配送员工号，托运公司编号，订单 ID，地址 ID，发运时间)”。

**试题三 参考答案及解析**

【参考答案】

【问题 1】A1：Central system；A2：Driver；A3：Traffic advice center：A4：Traffic advice center。 (其中 A1 与 A2 可交换，A3 与 A4 可交换)

【问题 2】U1：Underpaid transaction；U2：Record Illegal use；U3：Create transaction；

U4：Record traffic event；U5：Charge card； (其中 U1 与 U2 可交换)

【问题 3】C1：Central system；C2：Toll gantry；C3：Traffic lanes；C4：Radar sensor；

C5：Radio transceiver；C6：Digital Camera （注意：C4、C5、C6 可交换）

**【试题解析】**

本题考查面向对象分析与设计方法，UML 是面向对象分析与设计过程中最常用的建模语言，关于面向对象分析与设计、及 UML 相关知识请参见比特培训相关课程。本题的作答方式与试题一、试题二类似，比特建议先阅读问题，带着问题去阅读文字描述，然后边阅读、边对照 UML 图、边思考、边作答的方式能精准地定位答案。

文字描述“…在高速公路沿线的特定位置上设置一个横跨道路上空的龙门架(Toll gantry),龙门架下包括 6 条车道(Traffic lanes)(可知龙门架是整体类，车道是局部类，它们之间是组合的关联关系),每条车道上安装有雷达传感器(Radar sensor)、无线传输器(Radio transceiver)和数码相机(Digital Camera)等用于不停车收费的设备(可知车道是整体类，雷达传感器、无限传输器、数码相机是局部类，它们之间是组合关系。)，以完成正常行驶速度下的收费工作。”，在对比图 3-2 知 C2 为龙门架“Toll gantry”，C3 为车道“Traffic lanes”，C4 为雷达传感器“Radar sensor”，C5 为无线传输器“Radio transceiver”，C6 为数码相机“Digital Camera”，其中 C4、C5、C6 可交换。对于【问题 2】，虽然本题并未明确要求使用英文类名作答，但由于图 3-2 中已有的类名为英文，故本题的答案建议写英文类名，当然，中文类名也不错。

从文字描述“(2)当汽车通过某条车道时，不停车收费设备识别车载器内的特有编码，判断车型，将收集到的相关信息发送到该路段所属的区域系统(Regional center)中，计算通行费用，创建收费交易(Transaction)…”可看出，车道“Traffic lanes”类(C3)与交易“Transaction”类是有关联关系的，这与图 3-2 中 C3 与类 Transaction 的“1：\*”的关系相符；同时也能看出一个区域系统(Regional center)覆盖多个路段，而一个路段有多条车道(Traffic lanes)，一个车道上方又安装有对应的龙门架，这与图 3-2 中类“Regional center”和类“Toll gantry”的“1：\*”的关联关系相符。

“…计算通行费用，创建收费交易(Transaction)，从驾驶员的专用账户中扣除通行费用。如果驾驶员账户透支,则记录透支账户交易信息。…”中“如果”两字表明系统在执行用例“记录收费交易(Create transaction)”时，用例“记录透支账户交易信息(Underpaid transaction)”有可能执行，也可能不执行。这说明用例“Create transaction”是被扩展用例，而“Underpaid transaction”是扩展用例，再对照图 3-1 可知 U3 为“Create transaction”，U1 为“Underpaid transaction”，注意，在用例的扩展关系中，箭头指向“被扩展”的用例。

通过 “…区域系统再将交易后的账户信息发送到维护驾驶员账户信息的中心系统(Central system)…”可知区域系统(Regional center)与中心系统(Central system)是多对 1 的关联关系，这与图 3-2 中 C1 处对比知 C1 为类“Central system”。

通过以上分析，用例“记录收费交易(Create transaction)”需要从从驾驶员的专用账户中扣除通行费用，而驾驶员账户信息由中心系统(Central system)进行维护，故图 3-1 中与用例 U3“记录收费交易(Create transaction)”相关联的参与者 A1 为“Central system”。

文字描述“(3)车载器中的磁卡可以使用邮局的付款机进行充值。充值信息会传送至中心系统,以更新驾驶员账户的余额。”表明“中心系统(Central system)”与用例“此卡充值(Charge card)”有关联关系，而充值应该是驾驶员(Driver)自行完成，故图 3-1 中 A2 应为参与者“驾驶员(Driver)”，用例 U5 为“此卡充值(Charge card)”。A1 与 A2 的答案可交换。文字描述“(4)当没有安装车载器或者车载器发生故障的车辆通过车道时，车道上的数码相机将对车辆进行拍照，并将车辆照片及拍摄时间发送到区域系统，记录失败的交易信息；并将该交易信息发送到中心系统。”也表明 ETC 系统在记录交易信息时可能会出现失败的状况，故用例“记录失败交易信息(Record Illegal use)”应对应图 3-1 中的 U2。U1 与 U2的答案可交换。

文字描述“(5)区域系统会获取不停车收费设备所记录的交通事件(Traffic events);交通广播电台(Traffic advice center)根据这些交通事件进行路况分析并播报路况。”表明用例“记录交通事件(Record traffic event)”与两个参与者有关联关系，一个是“交通广播电台(Traffic advice center)”，另一个是“区域系统(Traffic advice center)”，再对照图 3-1 可知，U4 为“Record traffic event”，A3 为“Traffic advice center”，A4 为“Traffic advice center”，其中 A3 与 A4 可交换。对于【问题 1】中每个参与者的答案，比特建议写英文名称，当然写中文名称也不错，因为题目并未规定，但是对于【问题 2】中的用例名一定要写英文名称，因为表 3-1 已经明确了用例名为英文名称。

要想正确地解答本题，需要掌握面向对象的分析与设计思想和 UML 建模语言知识，同时还需要有丰富的解题经验。对于本题，有解题经验的朋友会发现，表 3-1 中用例名为英文名称，图 3-2 中已经有名称的类也是英文名称可得出的结论是用例图 3-1 和类图 3-2 中需要填入的内容也应该是英文名称，那么答案就应该缩小到本题文字描述中具有英文名称的关键字部分，这就一下子就缩小了答案的搜索范围呢！！

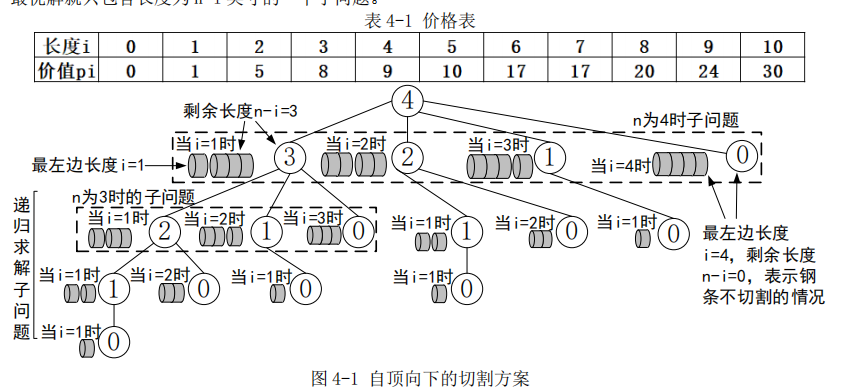
**试题四 【参考答案】**

(1)i<=n (2)i<=j (3)temp<p[i]+r[j-i]?p[i]+r[j-i]:temp (4)r[j]=temp; (5)动态规划 (6)O(2𝑛) (7)O(𝑛2)

【试题解析】

本题要解决钢条如何切割从而获取最大价值的问题，这是一个最优化问题。假设钢条切割价值类似表 4-1 所示，对于长度为 n 英寸的钢条，用rn表示最大收益，则rn=max{r1+rn−1, r2+rn−2,…,rn−1+r1, pn}，其中pn表示不切割钢条的情况，其他式子表示切成两段，分别长 i 英寸和 n-i 英寸，接着求解这两段的最优切割收益。为了求得原问题的解，先求性质完全一样但规模更小的子问题，即完成首次切割后，将切割出的两部分分别求解后再组合子问题的最优解，在所有可能的两段切割方案中选取组合收益最大的构成原问题的最优解，据此分析本题的钢条切割问题有最优子结构性质，这适合使用动态规划算法。

根据本题题意，将钢条从左边切割下长度为 i 英寸的一段，然后只对剩下的右边长度为 n-i英寸的另一段进行再切割 (递归求解)，对左边的一段则不再进行切割，所以rn=max1≤i≤𝑛(pi + rn−i)，即左边一段的价值加上右边的最大收益，这种情况下，最优解就只包含长度为 n-i 英寸的一个子问题。



如图4-1为长度为4英寸的钢条自顶向下的切割方案树。将长度为4英寸的钢条切成两段的情况有四种：

1.当i=1时，即最左边切割段长度为1英寸，右边段长度为3英寸(n-i=4-1=3)，获取的收益为p1+r4−1，而r4−1的最大收益还需要递归求解;

2.当i=2时，即最左边切割段长度为2英寸，右边段长度为2英寸(n-i=4-2=2)，获取的收益为p2+r4−2，而r4−2的最大收益还需要递归求解;

3.当i=3时，即最左边切割段长度为3英寸，右边段长度为1英寸(n-i=4-3=1)，获取的收益为p3+r4−3，而r4−3的最大收益还需要递归求解;

4.当i=4时，即最左边切割段长度为4英寸，右边段长度为0英寸(n-i=4-4=0)，获取的收益为p4，这种情况表示钢条不切割，获取的收益即p4，当然，这种场景也无须再递归求解。

以上的四种情况需要一个循环来实现，且能看出i的取值范围是1～4，比较4种情况的收益获得最大收益(这与空(1)处的循环体代码对应)。当钢条的长度为n英寸时，则i的取值范围是1～n，这与试题空(1)处的代码对应，故空(1)填写“i<=n”。

从图4-1可看出，算法从最长的长度n开始依次将规模大的问题划分为规模(n-i)较小的子问题，然后自顶向下，依次往深度方向递归求解子问题的解，图中的叶子节点个数即为算法的时间复杂度，长度为 4 英寸的叶子节点有24−1=8，则长度为n英寸的叶子节点个数为2𝑛−1个，故空(6)为“O(2𝑛)”。从图中易看出存在相同子问题的重复求解现象，如根为2的子树就有两棵(表示长度为2英寸的同一子问题求解了两次)，根为1的子树有4棵(表示长度为1英寸的同一子问题求解了4次)，要避免相同子问题的重复求解需要使用动态规划算法。

试题中Bottom\_Up\_Cut\_Rod(int p[], int n)函数使用数组r[LEN]保存了每个子问题的收益值，消除了同一子问题的重复求解过程。此函数的求解过程如下：

1.当j=1 时，求长度为1英寸的钢条切割最优收益值，只有1种切割方案，即不切割,最左边长度为 i=1英寸，剩余的右边段长度为 j-i=1-1=0 英寸，获得的收益为p1+𝑟𝑗−𝑖=p1+𝑟0 =1+0=1，保存到 r[j]即r[1]中，值为 1；

2.当 j=2 时，有两种切割方案(求解规模为 2 的子问题)：

(1)最左边长度为 i=1 英寸，剩余的右边段长度为 j-i=2-1=1 英寸，获得的收益为p1+𝑟𝑗−𝑖 =p1+𝑟1 =1+1=2；

(2)最左边长度为 i=2=j 英寸，剩余的右边段长度为 j-i=2-2=0英寸(不切割)，获得的收益为p2 + 𝑟𝑗−𝑖=p2+𝑟2−2 = 5+0=5。

对比两种方案的收益值，取最大的一个值 5 存入r[j]，即r[2]中。

3.当 j=3 时，有三种切割方案(求解规模为 3 的子问题)：

(1)最左边长度为 i=1 英寸，剩余的右边段长度为 j-i=3-1=2 英寸，获得的收益为p1+r𝑗−𝑖 =p1+r3−1=p1+r2=1+5=6；

(2)最左边长度为 i=2 英寸，剩余的右边段长度为 j-i=3-2=1 英寸，获得的收益为p2+r𝑗−𝑖 =p2+r3−2=p2+r1=5+1=6；

(3)最左边长度为 i=3=j 英寸，剩余的右边段长度为 j-i=3-3=0 英寸(不切割)，获得的收益为p3+r𝑗−𝑖=p3+r3−3=p3+r0=8+0=8；

对比三种方案的收益值，取最大的一个值 8 存入 r[j]，即 r[3]中。

使用如上的算法步骤可依次推导出规模为 n-1 问题的解，最后求出规模为 n 的大问题的最优值，最优值保存在 r[n]中，这种顺序即是自底向上的求解过程。

根据以上的分析，j 的取值范围是 1～n(表示问题的规模)，而 i 表示每个规模为 j 的子问题对应的最左边段长度，i的取值范围为 1～j。以上的步骤需要双重循环来实现，对应空(2)、(3)、(4)处的代码，空(2)为“i<=j”，空(2)处的循环是求规模为 j 的子问题的最优值，需要比较每个切割方案(i=1、i=2、i=3、…、i=j)的收益并将最大的值保存到临时变量 temp 中，故空(3)填“temp<p[i]+r[j-i]?p[i]+r[j-i]:temp”，即实现rn=max1≤i≤𝑛 (pi + rn−i)，规模为j的子问题最优值需要保存到数组元素 r[j]中，故空(4)填写“r[j]=temp”。

Bottom\_Up\_Cut\_Rod(int p[], int n)函数的运行时间主要花费在循环体中的双重循环处，当 j=1 时，i 的范围是 1；当 j=2 时，i 的范围是 1～2；当 j=3 时，i 的取值范围是 1～3；依次类推，当 j=n 时，i 的取值范围是 1～n，故循环的次数为 1+2+3+4+…+n=(1+n)\*n/2，故空(7)填写“O(𝑛2)”。

以上的算法步骤中，由于每个子问题的最优值都保存在数组 r 中，故求规模更大的子问题最优值时，对于需要用到的较小规模子问题的最优值无须再次求解，只需直接从 r 数组中读取即可，如求解规模为 3 且左边段 i=1 时的收益值为p1+r𝑗−𝑖=p1+r3−1=p1+r2，其中r2为规模为 2 的子问题的最优值(在之前的步骤中已经求出并保存起来了)，这里只需要直接读取值参与运算即可，这既是动态规划策略思想，故空(5)填写“动态规划”。

本题对应的完整程序如下：

#include<stdio.h>

#define LEN 100

int Top\_Down\_Cut\_Rod(int p[], int n) {/\*自顶向下\*/

int r = 0;

int i;

if (n == 0) { return 0; }

for (i = 1; i <= n; i++) {// (1)

int tmp = p[i] + Top\_Down\_Cut\_Rod(p, n - i);

r = (r >= tmp) ? r : tmp;

}

return r;

}

int Bottom\_Up\_Cut\_Rod(int p[], int n) {/\*自底向上\*/

int r[LEN] = { 0 };

int temp = 0;

int i, j;

for (j = 1; j <= n; j++) {//(2)

temp = 0;

for (i = 1; i <= j; i++) {

temp = temp < p[i] + r[j - i] ? p[i] + r[j - i] : temp; //(3)

}

r[j] = temp; //(4)

}

return r[n];

}

void main()

{

int p[11] = { 0,1,5,8,9,10,17,17,20,24,30 }; //切割价值表

int r; //保存最优值

//r = Top\_Down\_Cut\_Rod(price, 4);

r = Bottom\_Up\_Cut\_Rod(price, 4); //求长度为 4 英寸的钢条切割最优值

printf("最大切割价值为：%d", r);

getchar();

}

输出结果：最大切割价值为：10

●试题五 参考答案及解析

【参考答案】

（1）virtual void buildPartA()=0; （2）virtual Product\* getResult()=0;（3）product->setPartA

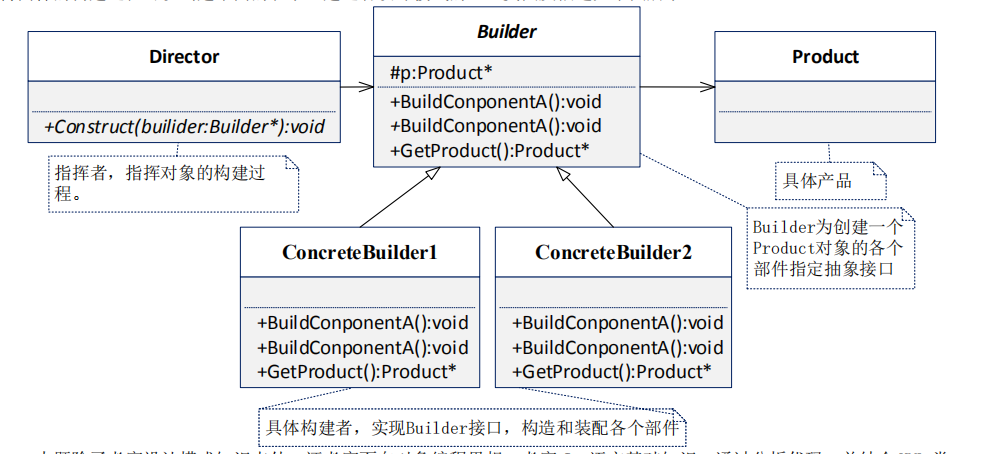
（4）product->setPartB（5）builder->buildPartA() 或 builder->buildPartB()

（注意，其中(1)与(2)可交换）

【试题解析】

本题考察建造者设计模式。建造者模式(Builder)又叫生成器模式，其将一个复杂对象的构建过程与它的表示分离，使

得同样的构建过程可以创建不同的表示。建造者设计模式的 UML 类图及描述如下图所示。



本题除了考察设计模式知识点外，还考察面向对象编程思想，考察 C++语言基础知识。通过分析代码，并结合 UML 类图 5-1 知 Builder 为两个派生类 ConcreteBuilder1 和 ConcreteBuilder2 的共同父类型，派生类中共同的成分(包括数据成员、函数成员，甚至是相同的函数头或函数体中相同的部分代码)都需提取到父类型中去。观察父类型中有一纯虚函数 buildPartB()(派生类中都已经实现，但函数体不一样)，则 buildPartA()在父类型中也应该被定义为纯虚函数，故空(1)填写 “virtual void buildPartA()=0”。同理，派生类中 getResult()也应该在父类型被定义为纯虚函数，空(2)填写“virtual Product\* getResult()=0”。

对于空(3)与空(4)，则使用比特培训课堂邓帅经常提到的“帅哥拥有富婆”的故事轻松搞定。帅哥“ConcreteBuilder1”类为啥要拥有富婆“product”(Product 类对象)作为他的数据成员呢？就是因为帅哥“ConcreteBuilder1”类要实现功能 buildPartA()但是又不想自己写代码实现，而让其拥有的富婆“product”替其实现，故空(3)应该填写“product->setPartA”，犹如你拥有了富婆，你就可花富婆的钱一样，否则帅哥找富婆干么？世间哪有那么多纯洁的爱情？！。不开玩笑，让我们严谨地分析！因为 ConcreteBuilder1 类的成员对象 product 已经实现了构建组件 A 的功能(setPartA 函数)，

故 ConcreteBuilder1 实现 buildPartA()时可调用其成员对象的功能函数 setPartA 来实现，避免同样的功能重复编码现象，这种是通过组合方式实现功能复用的场景。同理，空(4)应该填写“product->setPartB”。

在构建者设计模式中，为了减轻客户端的压力(可理解为本题的 main 函数)，让客户端无须关心对象的构建过程和构建细节，降低客户端和构建者之间的耦合度，需要引入指挥者类 Director，让指挥者类负责指挥对象的统一构建。指挥者类中的 construct()函数主要负责指挥完成对象的构建过程，而指挥者 Director拥有了一个 “builder”。

要实现的功能 construct()应该由其拥有的成员“builder”来实现，因为“builder”已经完成了 partA 和 partB的具体实现，故空(5)应为“builder->buildPartA()”或“builder->buildPartB()”。

●试题六 参考答案及解析

【参考答案】

（1）abstract void buildPartA() 或 void buildPartA()

（2）abstract Product getResult() 或 Product getResult()

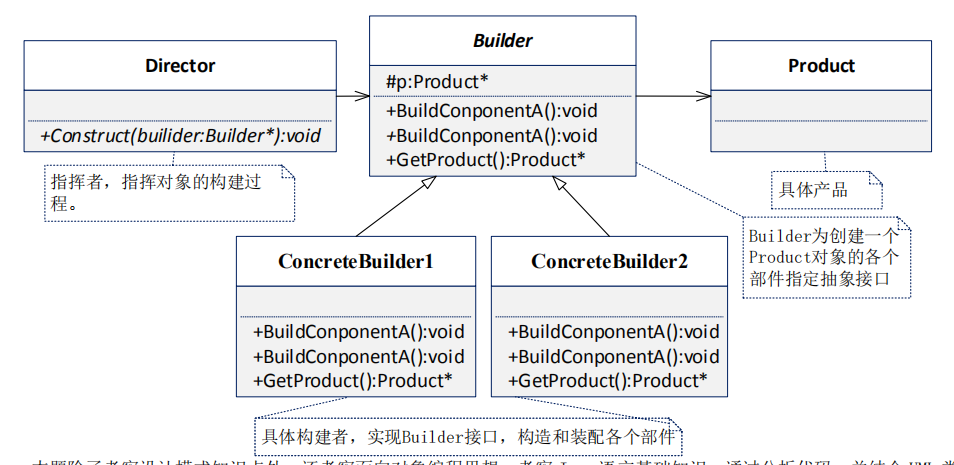
（3）product.setPartA （4）product.setPartB

（5）builder.buildPartA() 或 builder.buildPartB()

（注意，其中(1)与(2)可交换）

【试题解析】

本题考察建造者设计模式。建造者模式(Builder)又叫生成器模式，其将一个复杂对象的构建过程与它的表示分离，使得同样的构建过程可以创建不同的表示。建造者设计模式的 UML 类图及描述如下图所示。



本题除了考察设计模式知识点外，还考察面向对象编程思想，考察 Java 语言基础知识。通过分析代码，并结合 UML 类

图 6-1 知 Builder 为两个派生类 ConcreteBuilder1 和 ConcreteBuilder2 的共同父类型，派生类中共同的成分(包括数据成员、函数成员，甚至是相同的函数头或函数体中相同的部分代码)都需提取到父类型中去。观察父类型中有一抽象函数“public void buildPartB()”(派生类中都已经实现，但函数体不一样)，则 buildPartA()在父类型中也应该被定义为抽象函数，故空(1)填写“abstract void buildPartA()”，由于接口中 public 和 abstract 两关键字都可省略，故也可写成答案 “void buildPartA()”。同理，派生类中 getResult()也应该在父类型被定义为抽象函数，空(2)填写“abstract Product getResult()”或“Product getResult()”。

对于空(3)与空(4)， “ConcreteBuilde1”类为啥要拥有 “product”(Product 类对象)作为他的数据成员呢？就是因为 “ConcreteBuilder1”类要实现功能 buildPartA()但是又不想自己写代码实现，而让其拥有 “product”替其实现，故空(3)应该填写“product.setPartA”，因为 ConcreteBuilder1 类的成员对象 product 已经实现了构建组件 A 的功能(setPartA 函数)，故 ConcreteBuilder1 实现 buildPartA()时可调用其成员对象的功能函数 setPartA 来实现，避免同样的功能重复编码现象，这种是通过组合方式实现功能复用的场景。同理，空(4)应该填写“product.setPartB”。

在构建者设计模式中，为了减轻客户端的压力(可理解为本题的 Test 类)，让客户端无须关心对象的构建过程和构建细节，降低客户端和构建者之间的耦合度，需要引入指挥者类 Director，让指挥者类负责指挥对象的统一构建。指挥者类中的 construct()函数主要负责指挥完成对象的构建过程，而指挥者 Director拥有了一个 “builder”，故要实现的功能 construct()应该由其拥有的成员“builder”来实现，因为“builder”已经完成了 partA 和 partB的具体实现，故空(5)应为“builder.buildPartA()”或“builder.buildPartB()”。