目 录

一 软件基础知识试题精解

高级程序员级软件基础知识试题覆盖面宽，有一定的理论深度，需要考生全面系统地掌握大纲所规定的内容。数据结构、操作系统、程序语言、软件工程是考核的重点。二叉树、图、存储管理、编译原理、软件设计方法等知识点考查的次数比较多，应当是重点中的重点。

1990～2000 年软件基础知识试题，按所涉及的知识点统计如下：数据结构基础知识 14 道题，程序语言基础知识 13 道题，操作系统基础知识 17 道题，软件工程 21 道题，数据库基础知 13 道题，多媒体基础知识 5 道题。

1.1 数据结构基础

1.1.1 主要知识点

掌握线情表、多维数组、阵列、栈、树、二叉树，图的定义、存储和操作以及常见的排序和查找算法。重点是二叉树和图以及与其相关的算法。

1.1.1.1 数据结构概述

数据结构是指数据对象及其相互关系和构造方法，一个数据结构 B 形式上可以用一个二元组表示为 B=(A，R)。其中，A 是数据结构中的数据(称为结点)的非空有限集合，R 是定义在 A 上的关系的非空有限集合。数据结构按逻辑关系的不同分为线性结构和非线性结构两大为，共中非线性结构又可分为树形结构和图结构，树形结构又可分为树结构和二叉树结构。

1.1.2 线性表

数据结构中，线性结构习惯称为线性表。线性表是最简单也是最常用的一咱数据结构，它是由相同类型的结点组成的有限序列。一个由 n 个结点 e0，e1，…en-1 组成的线性表记为(e0，e1，…en-1)。线性表的结点个数称为线性表的长度，长度为 0 的线性表称为空的线性表，简称空表。对于非空线性表，e0 是线性表的第一个结点，en-1 是线性表的最后一个结点。线性表的结点构成一个序列，对序列中两个相邻结点 ei-1 和 ei，称前者是后者的前驱结点，后者是前者的后继结点。线性表最重要的性质是线性表中结点的相对位置是确定的。线性表的结点也称为表元，或称记录，要求线性表的结点是同一类型的任何数据。线性表的结点可由若干个成分组成，其中能唯一标识表元的成分称为了，简称键。线性表包含的表元个数可以动态增加或减少，可以在任何位置插入或删除表元。

线性表常用的运算可分成 4 类：查找运算、插入运算、删除运算和其他运算。每类又包括若干种运算。如查找运算就有两种，一种是查找线性表中某个结点的值，另一种是在线性表中查找具有给定键值的记录。

有多种存储方式能将线性表存储在计算机内，其中最常用的是顺序存储和链接存储。

(1)线性表的顺序存储

线性表的顺序存储是最简单的存储方式。程序通常使用一个足够在的数组，从数组的第一个元素开始，将线性表的结点依次存储在数组中。用数组元素的顺序存储来体现线性表中

结点的先后次序关系。其最大优点是能直接访问线性表中的任意一个结点。

（2）线性表的链接存储

用单链表存储线性表是线性表的链接方存储方式。从链表的第一个表元开始，将线性表的结点依次存储在链表的各表元中。链表的每个表元除了要存储线性表结点的信息外，还要有一个成分用来存储其后继结点的指针。

1.1.1.3 栈和队列

栈是只允许在同一端进行插入和删除运算的线性表。允许插入和删除的那一端称为栈顶，另 一端为栈底。若有栈 $S=(s_0，s1，…，s_{n-1})$，则 s0 为栈底结点，s_{n-1} 为栈顶结点。习惯称插入栈的结点为进栈，删除栈的结点为出栈。因为最后进栈的结点必定最先出栈，所以栈具有后进先出的的特性。可以用顺序存储线性表来表示栈，栈也可以用链表实现，用链表实现的栈称为链接栈。

队列是只允许在一端进入插入，另一端进行删除运算的线性表。允许删除运算的一端称为队首，允许插入运算的端为队尾。习惯称插入队列结点为进队，删除队列结点为出队。若有队列 $Q=(q0，q1，…，q_{n-1})q0$ 为队首结点，q_{n-1} 为队尾结点。因最先进入队列的结点将最先出 队，所以队列具有先进先出的特性。可以用顺序存储线性表来表示队列，也可以用链表实现，用链表实现的栈称为链接队列。

1.1.1.4 数组和字符串

数组是最常用的数据结构之一，一般用来顺序存储线性表。数组由固定个数的元素组成，全部元素的类型相同，元素依次顺序存储。每个元素对应一个下标，数组元素按数组名和元素的下标引用，数组元素的下标个数称为数组的维数。在 C 语言中，约定数组的第 1 个元素的下标为 0，其余依次类推，由 n 个元素组成的数组，其最后一个元素的下标为 n-1。数组元素可以是任何类型的，当元素本身又是数组时，就构成多维数组。

多维数组是一维数组的推广，多维数组中最常用的是二维数组。多维数组的所有元素并未排在一个线性序列里，要顺序存储多维数组就需要按一定次序把所有的数组元素排在一个线性序列里，常用的排列次序有行优先顺序和列优先顺序两种。对于多维数组，C 语言按行优先顺序存放。

字符串是非数值处理应用中重要的处理对象。字符串是由某字符集上的字符所组成的任何有限字符序列。不包含任何字符的字符串称为空字符串。字符串所包含的有效字符数称为字符串长度。一个字符串中任意连续的子序列称为该字符串的子串。

1.1.1.5 树

树和二叉树是非线性数据结构，用它们能很好地描述有分支和层次特性的数据集合。习惯称树和二叉树数据结构为树形结构。在许多算法中，都经常使用树形结构描述问题的求解过程或表示求解的对策等。由于二叉树有确定的分支个数，又可以为空，有良好的递归性质，与一般树相比较，特别适宜于程序设计，因此也常将树转换成二叉树进行处理。

(1)树的基本概念

树是一种多分支多层次数据结构，由一组结点组成。由于它呈现出与自然界的树类似的结构形式，所以称它为树。树是由一个或多个结点组成的有限集 T，它满足以下两个条件：

①有一个特定的结点，称为根结点；

②其余的结点分成(m＞m=0)个互不相交的有限集 T0，T1，…，Tm-1。其中每个集合又都是一棵树，称 T0，T1，…，Tm-1 为根结点的子树。

树的定义是递归的，即一棵树由子树构成，子树又由更小的子树构成。一棵树至少有一个结点。一个结点的子树数目，称为结点的度。树中各结点的度的最大值称为树的度。

(2)树的存储结构

树是非线性的结构，不能简单地用结点的线性表来表示。树有多种实用的存储结构，最常用是标准存储结构和带逆存储结构。在树的标准存储结构中，树中的结点内容可分成两部分：结点的数据和指向子结点的指针数组。当程序需从结点返回到其父结点时，需要在树的结点中存有其父结点的位置信息，这种存储形式就是带逆存储结构。

(3)树的遍历

在应用树结构时，常要求按某种次序获得树中全部结点的信息，这可通过树的遍历操作来实现。常用的树的遍存方法有：

·树的前序遍历。首先访问根结点，然后从左到右按前序遍历根结点的各棵子树。

·树的后序遍历。首先从左到右按后序遍历根结点的各棵子树，然后访问根结点。

·树的层次遍历。首先访问处于 0 层上的根结点，然后从左到有依次访问处于 1 层、2 层上的结点等，即自上而下从左到右逐层访问树各层上的结点。

·访问树中的所有叶子结点。

1.1.1.6 二叉树

(1)二叉树的基本概念

与一般的树结构相比较，二叉树在结构上更规范、更有确定性，二叉树的每个结点有两棵子二叉树，分别简称为左子树和右子树。因二叉树可以为空，所以二叉树中的结点可能没有子结点。二叉树的定义为：二叉树是一个有限的结点集合，该集合或者为空，或者由一个根结点及其两棵互不相交的、左右二叉子树所组成。

二叉树与树不同，首先二叉树可以为空，空的二叉树没有结点。另外，在二叉树中，结点的子树是有序的，分左、右两棵子二叉树。一般情况下，二叉树常采用类似树的标准存储形式来存储。

(2)二叉树的遍历

树的所有遍历方法都适用于二叉树，常用的二叉树遍历方法有 3 种：

·前序遍历。访问根结点→按前序遍历根结点的左子树→按前序遍历根结点的右子树

·中序遍历。按中序遍历根结点的左子树→访问根结点→按中序遍历根结点的右子树

·后序遍历。按后序遍历根结点的左子树→按后序遍历根结点的右子树→访问根结点

注意以上 3 种遍历方法都是递定义的。

1.1.1.7 二叉查找树

查找树便于链式存储，还能实现快速查找。作为一种特殊的二叉树，它或者为空，或者满足下列条件：

·若该树根结点的左子树非空，其左子树所有结点的键值都小于该树根结点的键值。

·若该树根结点的右子树非空，其左子树所有结点的键值都大于该树根结点的键值。

·该树的根结点的左子树和右子树均为查找树。

根据以上定义，如果进行中序遍历，即可得到一个从小到大的结点序列。

1.1.1.8 图

(1)图的基本概念

图是比较树形结构更复杂的一种数据结构。在线性结构中，除首结点没有前驱结点，末结点没有后继结点之外，一个结点只有一个前驱结点和一个后继结点。在树形结构中，除根结点没有前驱结点外，一个结点虽只有一个前驱结点，但可以有多个后继结点。但在图结构中，一个结点的前驱结点和后继结点的个数是任意。

在图中，数据结构中的结点被称为顶点，结点之间的关系被称为边。一个图 G 由非空有限的顶点集合 V 和有限的边的集合 E 组成，记为 G=(V，E)。

图分有向图和无向图。图中代表边的结点的偶对如果是有序的，则称此图为有向图。在有向图中用(V1，V2)表示一条有向边，V1 称为边的起点，V2 称为边的终点。(V1，V2)和(V2，V1)代表不同的边。图中代表一条边的结点的偶对如果是无序的，则称此图为无向图。在无向图中，(V1，V2)和(V2，V1)这两个偶对表示同一条边。

如果限定任何一条边或弧的两个顶点都不相同，一个有 n 个顶点的无向图至多有 n(n-1)/2 条边，这样的无向图称为无向完全图。一个有向图至多有 n(n-1)条弧，这样的有向图称为有向完全图。如果给图的每条边赋予一个实数作为边的权，则该图称为带权图，或称为网。

(2)图的存储结构

最常用的图的存储结构是邻接矩阵和邻接表。

图的邻接矩阵是反映顶点间邻接关系的矩阵，设 G(V，E)是具有 n(n≥1)个顶点的图，G 的邻接矩阵 M 是一个 n 行 n 列的矩阵，若(i，j)或(i，j) E，则 M［i］［j］=1；否则 M［i］［j］=0。由邻接矩阵的定义可知，无向图的邻接矩阵是对称的，有向图的邻接矩阵不定的对称。

图也可用邻接表来存储，为图的每个顶点建立一个链表。且第 i 个链表中的结点表示与顶点 i 相关联的一条边，或由顶点 i 出发的一条弧。有 n 个顶点的图，需要这 n 个链表的头指针按顺序线性表方方式存储。在无向图的邻接表中，对应某结点的链表的结点个数就是该顶点的度。

(3)图的遍历

图的遍历是指从图中的某个顶点出发，沿着图中的边或弧访问图中的每个顶点，并且每个顶点只被访问一次。图的遍历通常采用深度优先搜索或广度优先搜索方法。

图的深度优先搜索类似于树的前序遍历。从图中某个结点出发，访问此结点，然后依次从此结点未被访问的邻接点出发进行深度优先搜索，直至图中所有该结点有路径相通的结点均被 访问到。若此时图中尚有结点未被访问，则另选图中一个未被访问的结点作起始点，重复上述过程，直至图中所有结点都被访问到为止。

广度优先搜索类似于树的层次遍历。从某个结点出发，访问此结点，然后依次访问与此结点邻接的、未被访问过的结点，然后再分别从这些为出发进行广度优先周游，直至图中所有被访问过的结点的相邻结点都被访问到。若此时图中有结点尚未被访问，则另选图中一个未曾访问过的结点作为起点，重复上述过程，直至图中所有结点都被访问到为止。

(4)最小代价生成树

设 G=(V,E)是一个连通的无向图，若 G1 是包含 G 中所有顶点的一个无回路的连通子，则称 G1 为 G 的一棵生成树。含有 n 个顶点的连通图的生成树有 n 个顶点和 n-1 条边。对一个带权的图(网)，在一棵生成树中，各条边的权值之和称为这棵生成树的代价。一个图可以有许多不同的生成树，其中代价最小的生成树称为最小代价生成树。

(5)求最短路径

求最短路径就是从图中某个顶点到其他顶点的最短路径。基本思想是把图中所有结点分成两组，第一组包括已确定最短路径的结点，第二组包括尚未确定最短径的结点，按最短路径长度递增的顺序逐个把第二组的结点加到第一组中去，直至从某结点出发可以到达的所有结点都包括到第一组中。在这个过程中，总保护从该结点到第一组各结点的最短路径长度都不大于从该结点到第二组的任何结点的最短路径长度。

另外，拓扑排序和计算关键路径都是有向图的重要运算。

1.1.1.9 排序与查找

对于有 n 个结点的线性表(e_0，e_1，…e_{n-1})，按照结点中某些数据项的值递增或递减的次序，重新排列线性表结点的过程，称为排序。排序时参照的数据项称为排序码，通常选择结点的键值作为排序码。在排序过程中，线性表的全部结点都在内存、并在内存中调整它们在线性表中的存储顺序，称为内排序。在排序过程中，线性表只有部分结点被调入内存、并借助内存调整结点在外存中的存放顺序的排序方法称为外排序。常用的排序方法有：选择排序、直接插入排序、冒泡排序、希尔排序、堆排序、快速排序、合并排序和外排序，其中外排序是对大文件的排序。

查找就是在按某种数据结构存储的数据集全中，找出满足指定条件的结点的过程。按查找的条件分类，有按结点的关键码查找、按关键码以外的其他数据项查找和按关键码以外的其他数据项的组合查找等。按查找数据在内存还是在外存分为内存查找和外存查找。按查找的目的分类，如果查找只是为了确定指定条件的结点存在与否，称为静态查找。如果查找是为确定结点的插入位置或为了删除找到的结点，称为动态查找。

散列表又称杂凑表，是一种非常实用的查找技术。由于查找码与结点在数据结构中的位置之间不存在确定的关系，查找只能通过查找码与结点的关键码的反复比较来实现。对于通过比较来缩小查找范围来说，唯一能提高查找效率的办法是通过一次比较能大幅减小查找范围。

最理想的情况是直接利用查找码的信息，一次或几次存取便能存取所查结点。为此必须在结点的存储位置和它的关键码间建立一个确定的关系，从而让查找码直接利用这个关系确定结点的位置。

1.1.3 试题解析

数据结构是每年必考的知识点，与初级程序员级考试和程序员级考试相比，高级程序员级考试涉及的内容具有一定的广度和深度。从历年试题统计(见表 2-1)来看，考查的内容主要是树、二叉树、图和排序算法，特别是排序算法的平均比较比较次数，考查的次数相当多，复习是应作为重点，并应熟练掌握相关的算法。

表 2-1　历年高级程序员级数据结构基础试题统计

试题	考查知识点
1990 年试题 4	查找算法的平均查找长度（平均比较次数）
1990 年试题 6	正则二叉树
1991 年试题 3	树和二叉树
1992 年试题 6	常用排序算法
1993 年试题 6	有向图及其遍历
1994 年试题 15	常用排序算法的平均比较次数
1995 年试题 3	树和二叉树
1996 年试题 1	二叉树及其遍历
1996 年试题 14	图和树的判读
1997 年试题 4	线性表、二叉树、树、图的结构及其遍历方法
1998 年试题 3	常用排序算法
1999 年试题 1	常用排序算法
1999 年试题 2	图的概念及图的存储结构
2000 年试题 1	树和二叉树

试题 1(2000 年试题 1)

从供选择的答案中，选出应填入下面叙述中 { }内的最确切的解答，把相应编号写在答卷的对应栏内。

二叉树的前序、中序和后序遍历法最适合采用{ A }来实现。

查找树中，由根结点到所有其他结点的路径长度的总和称为{ B }，而使上述路径长度总和达到最小的树称为{ C }，它一定是{ D }。

在关于树的几个叙述中，只有{ E }是正确的。

供选择的答案

A： ①递归程序 ②迭代程序 ③队列操作 ④栈操作

B： ①路径和 ②内部路径长度 ③总深度 ④深度和

C： ①B-树 ②B+-树 ③丰满树 ④穿线树

D： ①B-树 ②平衡树 ③非平衡树 ④穿线树

E：①用指针方式存储有 n 个结点的二叉树，至少要有 n+1 个指针

②m 阶 B-树中，每个非叶子结点的后件个数≥［m/2］

③m 阶 B-树中，具有 k 个后件的结点，必含有 k-1 个键值

④平衡树一定是丰满树

【解析】

本题综合考查树形数据结构知识。部分内容与 1991 年试题 3 相似。

由于二叉树的前序、中序和后序遍历方法都是递归定义，所以最适合采用递归程序来实现。查找树中，由根结点到所有其他结点的路径长度总和称为内部路径长度。具有最小内部路径长度的树是丰满树，对丰满树的查找树进行插入或者删除操作后，会产生一棵非丰满树。平衡二叉树是指其上任一结点的左右子树的高度(或者结点个数)保持一定比例的树，即平衡树上任一结点的左、右子树仍然保持平衡。平衡树的查找效率和丰满树相近，但是在插入或者删除结点时，更能动态地平衡树的特点。由丰满树同平衡树定义可知，丰满树一定是平稳树，但是平衡树不一定是丰满树。

m 阶 B-树是一种平衡的 m 叉树，具有如下的性质：

(1)每个结点的后件个数小于等于 m；

(2)除了根结点的各结点之外，每个结点的后件个数大于等于(m/2)；

(3)具有 k 个后件的非叶结点含有 k-1 个键值；

(4)所有叶结点在同一层上，而且不附有信息。

【答案】：A：① B：② C：③ D：② E：③

试题 2(1999 年试题 1)

从供选择的答案中，选出应填入下面叙述中{ }内的最确切的解答，把相应编号写在答卷的对应栏内。

结定结点的关键字序列(F，B，J，G，E，A，I，D，C，H)。对它按字母的字典顺序进行排列，采用不同方法，其最终结果相同，但中间结果是不同的。

Shell 排序的第一趟扫描(步长为 5)

结果是结果应为{ A }。

冒泡排序(大数下沉)的第一趟起泡的效果是{ B }。

快速排序的第一趟结果是{ C }。

二路归并排序的第一趟结局是{ D }。

若以层次序列来建立对应的完全二叉树后，采用筛选法建堆，其第一趟建的堆是{ E }。

供选择的答案

A：①(B，F，G，J，A，D，I，E，H，C)

②(B，F，G，J，A，E，D，I，C，H)

③(A，B，D，C，E，F，I，J，G，H)

④(C，B，D，A，E，F，I，G，J，H)

B：①(A，B，D，C，F，E，I，J，H，G)

②(A，B，D，C，E，F，I，H，G，J)

③(B，F，G，E，A，I，D，C，H，J)

④(B，F，G，J，A，E，D，I，C，H)

C：①(C，B，D，A，F，E，I，J，G，H)

②(C，B，D，A，E，F，I，G，J，H)

③(B，A，D，E，F，G，I，J，H，C)

④(B，C，D，A，E，F，I，J，G，H)

D：①(B，F，G，J，A，E，D，I，C，H)

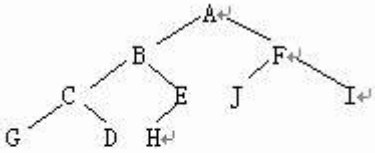
②(B，A，D，E，F，G，I，J，H，C)

③(A，B，D，C，E，F，I，J，G，H)

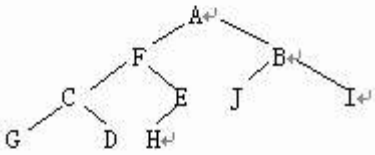
④(A，B，D，C，F，E，J，I，H，G)

E：

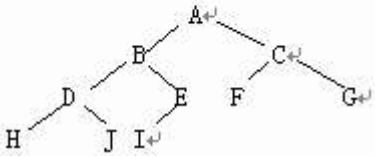
①



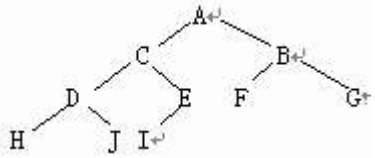
②



③



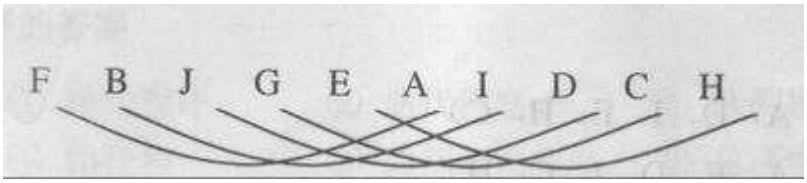
④



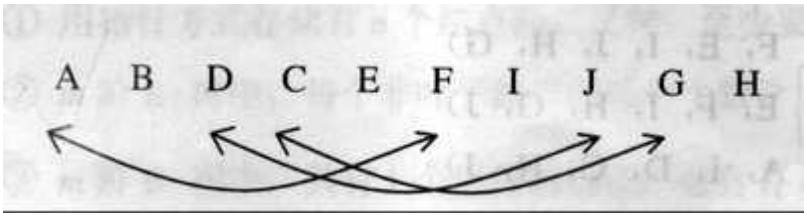
【解析】

解答本题的关键是要掌握几种主要的排序算法。排序是数据处理中经常使用的一种重要运算。插入排序、选择排序、交换排序、基数排序和归并排序是几种常用的排序方法，每种方法中还包括更具体的方法。

插入排序的基本思想是：每一步将一个待排序的记录按其关键码值的大小插入到前面已排序的文件中的适当位置上，直到全部插完为止。Shell 排序是插入排序的一种，按增量将待排序记录分组，先取增量 i＜n，把全部记录分成 i 个组，所有距离为 i 的倍数的记录放在一组中，各组内采用插入顺序。然后取 j＜i 作为增量，重复上述分组和排序工作，直至增量为 1，即所有记录放在一个组中排序为止。题中步长 i=5，先将文件分为 5 组，如下所示：



然后在组内用插入法排序，进行第 1 趟扫描，结果如下，只有用箭头连接的组排序位置发生变化。



交换排序的基本思想是两两比较待排序记录的关键码值，并交换不满足顺序要求的那些偶

对，直到全部满足为止。冒泡排序和快速排序又是交换排序的两种具体方法。冒泡排序将待排序的记录两两比较，若为逆序则交换。在升序排序过程中，关键码大的记录下沉，关键码小的记录上浮，故称冒泡。将序列按照此方法不到尾处理一遍后，当前处理范围内最大的记录被交换到最后，如此往复，最后得到一个有序的记录序列。问题 B 比较简单，顺次将关键字序列两两比较，第 1 趟处理结果显然选③。

快速排序是对冒泡排序的一种改进，方法是：在待排序序列中确定一个记录，以它为基准，用交换的方法将所有的记录分成两部分，即关键码值比它小的一部分和比它大的一部分。再分别对两个部分实施上述过程，一直重复到排序完成。问题 C 解答比较简单。取第一个记录 F 为基准，将比 F 大的移到后面，比 F 小的移到前面。一般采用从两端往中间加入的移动方式，第 1 趟排序的过程：

F，B，J，G，E，A，I，D，C，H

C，B，J，G，E，A，I，D，F，H

C，B，F，G，E，A，I，D，J，H

C，B，D，G，E，A，I，F，J，H

C，B，D，F，E，A，I，G，J，H

C，B，D，A，E，F，I，G，J，H--第一趟结束

归并排序要求待排序列已经部分排序，形成了若干子序列。归并排序就是将已排序的子序列进行合并，得到完全排序的序列。合并时比较各子序列的第 1 个记录，便可找出第 2 个记录。如此继续，一遍扫描即可得出结果。具体的方式分为二路归并和多路归并。采用二路归并对有 n 个记录的序列进行排序时，应先将序列分成 n/2 个子序列，使每个子序列包含 2 个(或 1 个)记录，在此子序列内进行排序。然后再将记录序列分成 n/4 个子序列(即从前往后两两合并子序列)，再在子序列内进行排序。重复上述过程，直到只有一个子序列，则排序完成。理解了这个过程，则问题 D 的解答就会非常简单。先将待排序列(F，B，J，G，E，A，I，D，C，H)两两合并，分成 5 个子序列，在各子序列内排序，第一趟结局显然为(B，F，G，J，A，E，D，I，C，H)。

选择排序的基本思想是每次从待排序的记录中选择出码值最小(或最大)的记录，顺序放在已排序的记录序列的最后，直到全部排完为止。堆排序是选择排序的一种，采用堆排序时，首先要按层次序列来建立对应的完全二叉树，然后按筛选法建立立堆。在建堆的过程中，要记住堆的特征，作为堆的二叉树具有这样的特征：树根的关键码值小于树叶的关键码值；堆的调整过程是从最下层开始的，先将二叉树的树根与左子树进行比较，如果树叶小于树根，则交换；然后比较树根与右子，若树叶小于树根，则交换。同时，在交换的过程中如果影响到下层的堆，需要同时调整下层的堆。题中 E 问题的解答用示意图说明。在这里，初始建成的完全二叉树如图 2-1(a)所示。在调整完最下层的子树后，整个子树的结构如图 2-1(b)所示。调整完第 2 层后，整个二叉树的结构如图 2-1(c)所示。最后一次调整的结果就是题中问题 E 的答案②。

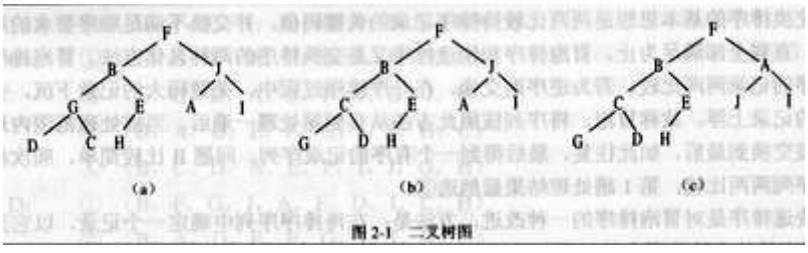


图 2-1　二叉树图

【答案】：A：③ B：③ C：② D：① E：②

试题 3(1999 年试题 2)

从供选择的答案中，选出应填入下面叙述中{ }内的最确切的解答，把相应编号写在答卷的对应栏内。

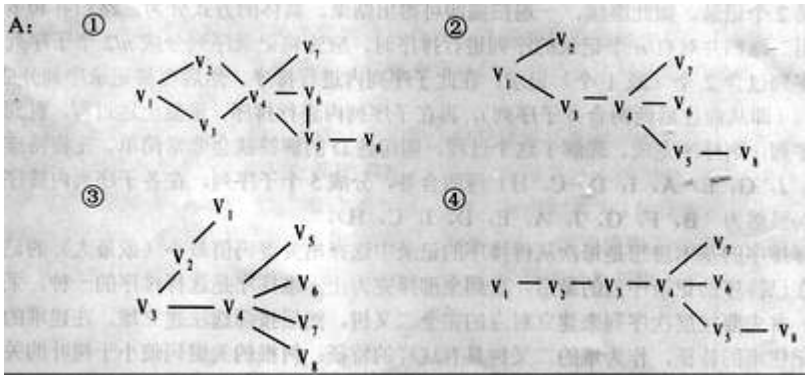
给定数据结构(V，E)，V 为结点的有限集合，V={V1，V2，V3，V4，V5，V6，V7，V8}，E 是 V 上关系的集合。

E={＜V1，V2＞，＜V3，V4＞，＜V5，V8＞，＜V6，＜V1，V3＞，V4＜，V5＞，V6＜，＜V1，V3＞，V4＜，V5＞，＜V2，V4＞，V4＜，V6＞＝它所对应的图表是{ A }，这就是{ B }。

图的存储结构主要有邻接表和{ C }，若用邻接表来存储一个图，则需要保存一个{ D }存储的结点表和若干个{ E }存储的关系(又称边表)。

供选择的答案

A：



B：①树 ②无向图 ③有向图 ④无回路图

C：①转称矩阵 ②邻接矩阵 ③状态矩阵 ④优先矩阵

D：①顺序 ②链接 ③散列 ④分块

【解析】

本题是一道关于图的试题，应该说是比较简单的。

在数据结构(V，E)中，有关系＜V1，V2＞和＜V1，V3＞，这样很快就答案案③和④排除了，然后再看答案①②。在数据结构(V，E)中，还有关系＜V2，V4＞，这在答案②中没有反映出来，所以又排除了答案②。既然确定了答案①就是正确的答案，仔细观察该图形，由于关系＜V1，V2＞、＜V1，V3＞、＜V2，V4＞，＜V3，V4＞以及关系＜V4，V6＞、＜V6，V5＞、＜V4，V5＞的特殊性，我们排除了该图是树、有向图、无回路图的可能性，数据结构(V，E)是图。

图的存储结构主要有邻接表和邻接矩两种。邻接矩阵是表示结点间的相邻关系的矩阵，若 G 是一个具有 n 个结点的图，则 G 的临界阵是如下定义的 n×n 矩阵：

A［ij］=0，表示(Vi，Vj)或者(Vj，Vi)是 G 的边；

A［ij］=1，表示(Vi，Vj)或者(Vj，Vi)不是 G 的边；

用邻接表来存储一个图，需要保存一个顺序存储的结点表和 n 个链接存储的边表。结点表的每个表目对应于图的一个结点，每个表目包括两个字段：一个是结点的数据或指向结点数据的指针，另一个是指向此结点的边表的指针。图的每一个结点都有一个边表，一个结点的边表的每个表目对应于与该结点相关联的一条边。它的每个表目也包括两个字段：一个是与此相边相关联的另一个结点的序号，另一个是指向边表的下一个表目的指针。对于有向图来说，用邻接表保存该图可以保存每个结点的出边表或入边表。

【答案】：A：① B：② C：② D：① E：②

试题 4(1998 年试题 3)

从供选择的答案中，选出应填入下面叙述中{ }内的最确切的答案，把相应编号写在人卷的对应栏内。

在内部排序中，通常要对被排序数据序列进行多趟扫描。各种排序方法有其不同的排序实施过程和(时间)复杂性。

对给定的整数序列(541，132，984，746，518，181，946，314，205，827)进行从小到大的排序时，采用冒泡排序和直接选择排序时，若先选出大元素，则第一趟扫描结果分别是{ A }和{ B }；采用快速排序(以中间元素 518 为基准)的第一趟扫描结果是{ C }。

设被排序数据序列有 n 个元素，冒泡排序和直接选择排序的复杂性是{ D }；快速排序的复杂性是{ E }。

供选择的答案

A～C：①(181，132，314，205，541，518，946，827，746，984)

②(541，132，827，746，518，181，946，314，205，984)

③(205，132，314，181，518，746，946，984，541，827)

④(541，132，984，746，827，181，946，314，205，518)

⑤(132，541，746，518，181，946，314，205，827，984)

⑥(132，541，746，984，181，518，314，946，205，827)

D、E：①$O(n\log_2 n)$ ②$O(n)$ ③$(nccn)$ ④$O(n^2)$

⑤$O(\log_2 n)2$ ⑥$O(n^2\log_2 n)$

【解析】

冒泡排序：冒泡排序的过程很简单。首先将第 1 个数与第 2 个数相比较，若为逆序则交换两数，然后比较每两个数与第 3 个数，依次类推，直到第 n-1 个数与第 n 个数进行过比较为止。上述过程称为一趟冒泡排序，结果是最大的数被称鲐了最后。然后进行第 2 趟，对前面 n-1 个数进行冒泡排序，结果是次大的数被移到了 n-1 的位置上。一般来说，第 i 趟冒泡排序是从第 1 个数到第 n-i+1 的位置上，整个排序过程需进行 $k(1\le k\le n)$趟。

分析冒泡排序的效率，若初始序列为正序，则只进行一次排序。在排序过程中只进行 n-1 次比较，不交换数据。若为逆序，则需进行 n-1 趟排序，需进行 n(n-1)/2 次比较，交换数据的数量组也相同。因此，冒泡排序的复杂性是 $O(n^2)$。

快速排序是对冒泡排序的一种改进，其基本思想是通过一趟排序将待排序的数据分成两部分，其中一部分的关键字均比另一部分的关键字小，然后再对这两部分分别进行快速排序，最后达到整个序列有序。快速排序的复杂是 $O(n\log_2 n)$。

直接选择排序，又称简单选择排序，其基本思想是每一趟在 n-i+1(i=1，2，…，n-1)个数据中选择最小的数据作为有序序列中的第 i 个数据。一趟直接选择排序的基本操作为通过 n-i 次关键字的比较，从 n-i+1 个数据中选出关键字最小的数据，并和第 i 个数据交换。直接选择排序过程中，所需交换数据的次数较少，最小值为"0"，最大值为 3(n-1)"。然而，无论数据的初始次序如何，它所需进行的关键字的比较次数相同，均为 n(n-1)/2，因此，直接选择排序的复杂性是 $O(n^2)$。

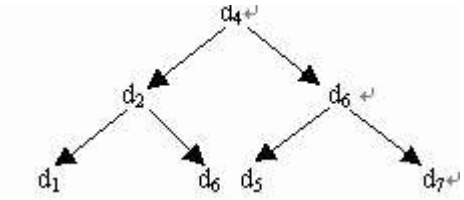
对于题中给定的整数序列(541，132，984，746，518，181，946，314，205，827)进行从小到大排序，若先选出较大的元素，则对于冒泡排序，第 1 趟操作为 541←→132，984←→746，984←→518，984←→181，984←→946，984←→314，984←→205，984←→827，其结果得到的序列为(132，541，746，518，181，946，314，205，827，984)；对于直接选择排序，第 1 趟操作为 984←→827，其结果得到的序列为(541，132，827，746，518，181，946，314，205，984)。

采用快速排序(以中间元素 518 为基准)的第 1 趟扫描结果是(205，132，314，181，518，746

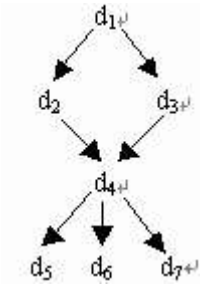
，946，984，827)。
【答案】：A：⑤ B：② C：③ D：④ E：①

试题 5(1997 年试题 4)

从供选择的答案中，选出应填入下面的叙述中{ }内的最确切的解答，把相应编号写在答卷的对应栏内。
设数据结构(D，R)由数据结点集合 D={di|1≤i≤7}及其上的关系 R 组成。
1.当 R={＜d_{i-1}，d|d_{i-1}，di∈D，2≤i≤7}，这个数据结构对应于 A。
2.当 R={＜d4，d2＞，d1＞，＜d2，d3＞，＜d4，d6＞，＜d6，d5＞，＜d6，d7＞}，这个结构的图形是 B，用 C 遍历法可以得到 A 的数据结构。
3.当 R={＜d1，d2＞，＜d1，d3＞，＜d2，d4＞，＜d4，d5＞，＜d4，d5＞，＜d4，d6＞，＜d4，d7＞}，这个结构的图形是 D。用 E 遍历法可以得到 A 的数据结构。
供选择的答案
A、B、D：①二叉树 ②队列 ③二叉排列序树
④线性表 ⑤无向图 ⑥有向无回路
C、E：①前序 ②中序 ③后序
④深度优先 ⑤广度优先
【解析】
该题要求考生熟练掌握数据结构中常用的线性表、二叉树、树、图的结构及其遍历方法。数据结构(D，R)中，D 由 7 个数据组成，R 决定这 7 个数据的排列方式。
(1)R={＜d_{i-1}，d|d_{i-1}，di∈D，2≤i≤7 }
其对应的数据结构可以表示为 d1→d2→d3→d4→d5→d6→d7，它对应一个线性表。
(2)R={＜d4，d2＞，＜d2，d1＞，＜d2，d3＞，＜d4，d6＞，＜d6，d5＞，＜d6，d7＞}
其对应的数据结构可以表示为：



这棵二叉树利用中序遍历法得到序列为 d1d2d3d4d5d6d7。



(3)R={＜d1，d2＞，＜d1，d3＞，＜d2，d4＞，＜d3，d4＞，＜d4，d5＞，＜d4，d6＞，＜d4，d7＞}，对应的数据结构为一个有向无回路图，利用广度优先遍历法得到的序列为 d1d2d3d4d5d6d7，正好对应(1)中的线性表结构。
【答案】：A④ B① C② D⑥ E⑤

试题 6(1996 年试题 1)

从供选择的答案中，选出应填入下面叙述中{ }内的最确切的解答，把相应编号写在答卷的

对应栏内。

一棵二叉排序树可顺序存放在一组物理上相邻的存储区中，每个结点及左、右针依次分别放在该存储区的 3 个连续单元中。现对一棵结点按字母的字典顺序构成的二叉排序树从根结点 P 开始顺序放在一个存储区中，结果如图 2-2 所示。其中 Li 为第 i 个结点的左指针，R 为第 i 个结点的右指针，则 L2 应为 A，L4 应为 B，R1 应为 C。该二叉排序树的前序遍历序列为 D，后序遍历序列为 E。

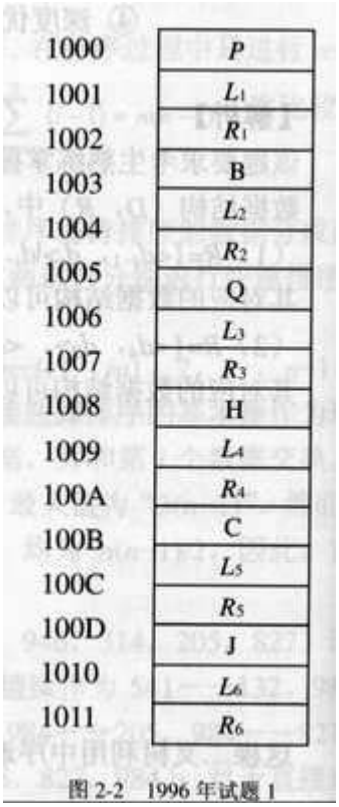


图 2-2　1996 年试题 1

供选择的答案：

A～C：①1003 ②1004

③100A… ④1009

⑤1006 ⑥1000

⑦100C ⑧100F

⑨Null

D、E：①PBQHCJ ②PBHCJQ

③BCHJPQ ④CJHBQP ⑤BHCJQP

【解析】

二叉树或者为空，或者由一个根结点加上左子树和右子树(互不相交的两棵二叉树)构成，因此，若依次遍历根、左子树、右子树，就有 6 种遍历方法，即 DLR、DRL、LDR、RDL、LRD 和 RL

D。限定先左后右的顺序，则有常用的前序遍历、中序遍历和后序遍历 3 种情况。

基于二叉树的递归定义，可得遍历二叉树的递归算法定义：

(1)先序遍历(DLR)算法：若二叉树为空，则进行的是空操作，否则访问根结点：

前序遍历左子树；

前序遍历右子树；

(2)中序遍历(LDR)算法：若二叉树为空，则进行的是空操作，否则

中序遍历左子树；
访问根结点；
中序遍历右子树。
(3)后序遍历(LRD)算法：若二叉树为空，则进行的是空操作，否则
后序遍历左子树；
后序遍历右子树；
访问根结点。
二叉排序树有如下特点：每个结点的左子树中所有结点的值都小于该结点的值，而右子树中所有结点的值都大于该结点的值，可知由根结点 P 始，按结点字母的字典顺序构成的二叉树如图 2-3 所示。

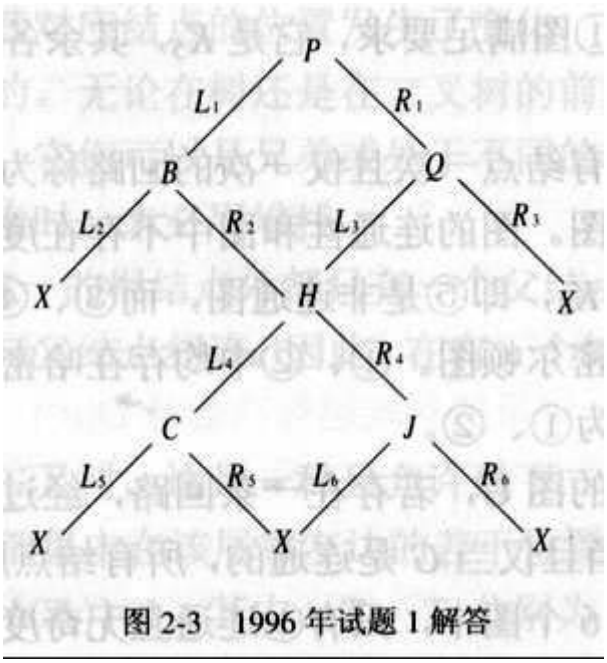
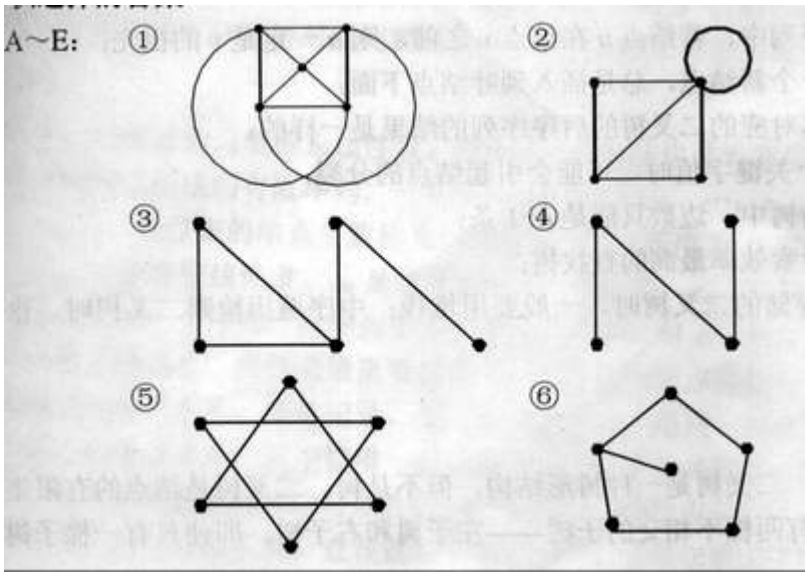


图 2-3　1996 年试题 1 解答

由图 2-3 可得出：L2 指向 Null；L4 指向 C，即 100C；R1 指向 Q，即 1006；该二叉树排序权前序遍历序列为 PBHCJQ，后序遍历序列为 2CJHBQP。
【答案】A：⑨ B：⑦ C：⑤ D：② E：④

试题 7（1996 年试题 14）

从供选择的答案中，选出应填入下面叙述中 内的最确切的解答，把相应编号写在答卷的对应栏内。
下列图中，A 是非简单图，B 是完全图，C 和 D 都是哈密尔顿图，其中 C 又是欧拉图，E 是树。
供选择的答案：



非简单图：既无平先边也无环的无向图称为简单无向图，有平先边或有环的图是非简单图。在 6 个图中只有②有环，其余的 5 个图都既无平先边也无环，因而 A 的答案应为②。

完全图：每个顶点度数都等于(n-1)的 n 阶无向简单图称为 n 阶无向完全图，并记作 kn。在给定的 6 个图中，只有①图满足要求，它是 k5，其余各图均不不完全图，所以 B 的答案为①。

哈密尔顿图：通过图中所有结点一次且仅一次的回路称为该图的哈密尔顿回路，具有哈密尔顿回路的图称为哈密顿图。图的连通性和图中不存在度数为 1 的顶点都是哈密尔顿图的必要条件。⑤图是两分离的 K3，即⑤是非连能图，而③、④、⑥已均有度数为 1 的顶点，所以③、④、⑤、⑥都不是哈密尔顿图。①、②中均存在哈密尔顿回路，因而①、②均为哈密尔顿图。所以 C、D 的答案为①、②。

欧拉图：给定无孤立结点的图 G，若存在一条回路，经过图中每边一次且仅一次，该条路称为欧拉回路。无向图 G，当且仅当 G 是连通的，所有结点度数全为偶数，且有欧拉回路，该无向图称为无向欧拉图。在 6 个图中，只有①连通且无奇度顶点，因而①为欧拉图，其余各图中，⑤无奇度顶点，但它不连通，所以不是欧拉图。而②、③、④、⑥中都有奇度顶点，因而它们也不是欧拉图，所以 C 的答案为①。

树：连通且无回路的无向图称为无向树。6 个图中只有④满足要求，其余 5 个图中均有回路，因而都不是树，E 的答案为④。

【答案】A：② B：① C：① D：② E：④

试题 8(1995 年试题 3)

从下列有关树的叙述中，选出 5 条正确叙述，并按编号从小到大的次序写在答卷的 A～E 栏内。

①一棵二叉树的层次遍历方法只有前序法和后序法两种；

②在哈夫曼树中，外部结点的个数比内部结点个数多 1；

③完全二叉树一定是平衡二叉树；

④在二叉树的前序序列中，若结点 u 在结点 v 之前，则 u 一定是 v 的祖先；

⑤在查找树中插入一个新结点，总是插入到叶结点下面；

⑥树的后序序列和其对应的二叉树的后序序列的结果是一样的；

⑦对 B 树删除某一个关键字值时，可能会引起结点的分裂；

⑧在含有 n 个结点的树中，边数只能是 n-1 条；

⑨最佳查找树就是检索效率最高的查找树；
⑩中序遍历二叉链存储的二叉树时，一般要用堆栈；中序遍历检索二叉树时，也必须使用堆栈。
【解析】
由二叉树的定义可知，二叉树是一种树形结构，但不是树。二叉树是结点的有限集，可以是空集。任一结点至多有两棵不相交的子树--左子树和右子树。即使只有一棵子树也必须区分清它是左子树还是右子树。
二叉树与树有着密切的关系：任一棵树都可用一棵二叉树唯一地表示。当然树中结点之间的父子、兄弟的关系在它所对应的二叉树中会有变化。树和二叉树的层次遍历方法有前序法、后序法、中序法三种。
由于树转换为二叉树时，其对应结点的位置发生了变化，因此树的后序序列和其对应的二叉树的后序序列一般是不同的。无论在树还是在二叉树的前序序列中，排序在前的结点未必是排序在后的结点父结点，它们可以是兄弟或处于不同的子树中。二叉树用中序遍历方法存储时，一般用堆栈，而检索时，未必用堆栈。
在树中，每个结点，除了唯一的根结点外都只有一个父结点，而根结点是没有父结点的。由于每个结点有且仅有一条边与父结点相连，因此，在有 n 个结点的树中，其边数只能是 n-1 条。
完全二叉树是一种特殊的二叉树，这是一种只允许最下二层结点的度数小于 2(其他均为 2)，并且最下面一层的结点都集中在该层靠左边的若干位置上。在二叉树中定义非空二叉树 T 的高度 h(T)=max{h(T1),h(T2)}+1。其中，T1、T2 分别为 T 的左子树和右子树。若一棵二叉树中任一结点的左子树高度与右子树高度之差不超过 1，则该称二叉树为平衡二叉树。根据完全二叉树与平衡树的定义，可知完全二叉树一定是平衡二叉树。
查找树也是一种二叉树，若二叉树 T 的结点按中序遍历，结点 u 的左子树的所有结点的键值都小于结点 u 的键值，而且结点 u 的右子树的所有结点的键值都大于结点 u 的键值。查找树有 3 种操作：查找、插入和删除。当向查找树插入一个新结点时，总是插在叶结点的下面。对给定的查找二叉树，在结点的每个空指针上都有附加结点作为该结点的子结点，称这些附加结点为外部结点，而原树的结点称为内部结点。哈夫曼树就是一种扩充二叉树。
在查找二叉树中查找结点，有许多种方法。最大查找时间在 $O(\log_2 n)$和 O(n)之间。由于所查找的结点是随机的，因此将平均查找时间作为代价函数。最佳查找树就是通过构造出各种可能的查找树，计算出每棵查找树的代价，然后挑选其中代价最小的树而获得的，因而检索效率是最高的。
B 树是一种平衡的多叉树，是索引文的有效结构。在 B 树中删除某一键值时，常采用与分裂相反的处理过程--联接，这样做有可能使整棵 B 树减少一层，但不会引起结点的分裂。
【答案】：A：② B：③ C：⑤ D：⑧ E：⑨

试题 9(1994 年试题 15)

从供选择的答案中，选出应填入下面有关排序算法复杂性的叙述中{ }内的正确答案，把编号写在答卷的对应栏内。
对由 n 个记录所组成的有按关键码排序时，下列各常用排序算法的平均比较次数分别是：二路归并排序为 A，冒泡排序 B，快速排序为 C。其中，归并排序和快速排序所需要的辅助存储分别是 D 和 E。
供选择的答案
A-E：①O(1) ②$(n\log_2 n)$ ③O(n) ④$O(n^2)$
⑤O（$n(\log_2 n)2$） ⑥$(n\log_2 n)$

【解析】

排序就是对由 n 个记录所组成的表按某个关键字的值的大小对记录重新排列。这是数据处理中常见的重要运算。排序的方法很多，衡量一个排序方法的优劣，主要看排序算法的时间复杂性和空间复杂性，即平均运算次数及所需要的辅助存储空间的大小。

冒泡排序(bubble sort)：从上到下对每对相邻记录比较关键字大小，使较小关键字的记录上升，比较一遍后最大的关键字的记录将排在最后，再对其余记录重复上述方法，反复执行，直到无记录上升为止，n 个记录的平均运算次数是 O(n)2。

归并排序(merge sort)：是把待排序的文件分成 n 个已排序的子文件，将这些文件合并得到完全排序的文件。n 个记录的平均运算次数是 $O(n\log_2 n)$，所需的辅助存储空间是 O(n)。

快速排序(quick sort)：又称分区交换排序。先取一个记录作为基准，用交换的办法把所有记录按关键字值比基准值大或小分列在基准记录的两边，然后再分别对这两边的记录重复上述步骤，直到排序完成。n 个记录的平均运算次数是 $O(n\log_2 n)$，所需的辅助存储空间是 $O(\log_2 n)$。

【答案】：A：② B：④ C：② D：③ E：⑥

试题 10(1993 年试题 6)

从供选择的答案中，选出应填入下面关于数据结构叙述中{ }内的正确答案，把编号写在答卷的对应栏内。

图 2-4 所示是带权的有向图 G 的邻表表示法。以结点 V1 出发，深度遍力图 G 所得的结点序列为 A；广度遍力图 G 所得的结点序列是 B；G 的一个拓扑序列是 C；从结点到 V1 到结点 V8 的最短路径是 D；从结点 V1 到结点 V8 的关键路径是 E。

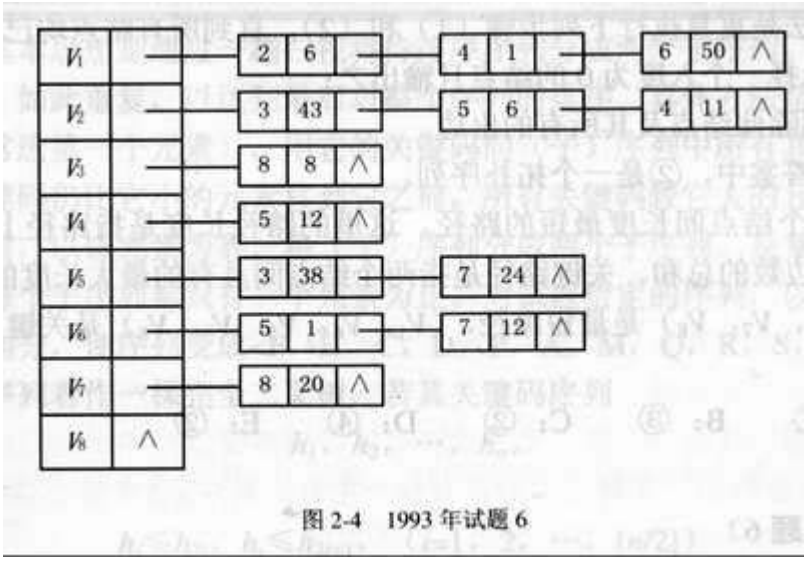


图 2-4　1993 年试题 6

供选择的答案

A～C：①V1，V2，V3，V4，V5，V6，V7，V8

②V，V2，V4，V6，V5，V3，V7，V8

③V1，V2，V4，V6，V3，V5，V7，V8

④V1，V2，V4，V6，V7，V3，V5，V8

⑤V1，V2，V3，V8，V4，V5，V6，V7

⑥V1，V2，V3，V8，V4，V5，V7，V6

⑦V1，V2，V3，V8，V5，V7，V4，V6

D、E①(V1，V2，V4，V5，V3，V8)

②(V1，V6，V5，V3，V8)
③(V1，V6，V7，V8)
④(V1，V2，V5，V7，V8)
【解析】
邻接表表示法可保存一个顺序存储的结点表，是图常用的一种存储方式。结点表的每一个表目对应于图的一个结点，包括结点的数据或指向结点数据的指针，以及指向此结点的边表的指针的两个字段。边表的每个表目对应于与该结点相关联的一条边，结点表的每个表目对应一个链接存储的边表，包括与该结点相关联的一个结点的序号及指向边表下一个表目的指针的两个字段。因此，题中邻接表所对应的如图 2-5 所示。

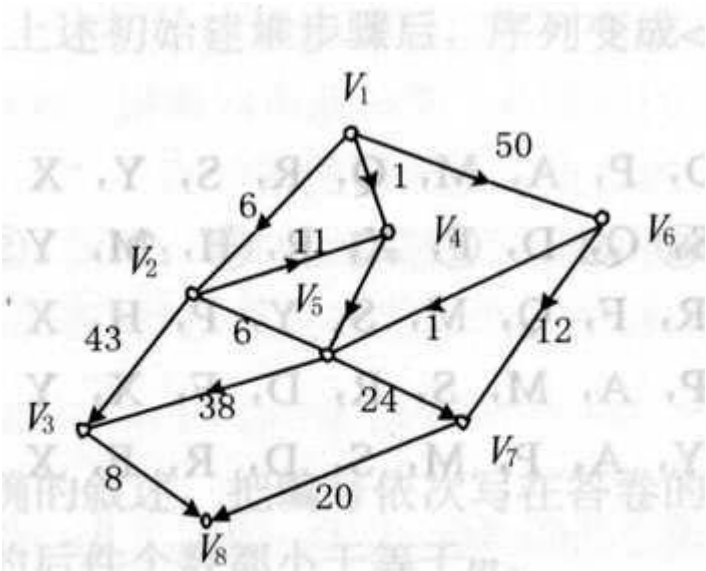


图 2-5　1993 年试题 6

图的深度优先遍历是从图中某个结点 V1 了发，访问此结点，然后依次从 V1 的未被访问的相邻
结点出发进行深度优先遍历，直至图中所有和 V1 有路径相通的结点都被访问到。此时图中若
尚有未被访问的结点，则另选图中一个未被访问到的结点作起始点。重复上述过程，直至图中所有结点都被访问到为止。因此，邻接表图的深度优先遍历是 V1，V2，V3，V8，V5，V7，V4，V6。
广度优先遍历是先访问结点 V1，然后访问 V1 连接到的所有未被访问的结点 V2，V3，…，Vt，再依次访问 V2，V3，…，Vt 连接到的所有未被访问的结点。如此进行下去，直到访问遍所有结点，因此邻接表图的广度优先遍历是 V1，V2，V4，V6，V3，V5，V7，V8。
如果有向图的某个结点序列满足如下条件：若从结点 Vi 到 Vj 有一条路径，则在序列中结点 Vi 必定在 Vj 之前，则称该序列是一个拓扑序列。任何无环有向图的结点都可以排在一个拓扑序列中。
拓扑排序的方法是重复执行下列步骤(1)和(2)，直到所有结点均已被输出。
(1)从图中选择一个入度为 0 的结点且输出之；
(2)从图中删除此结点及其所有的出边。
在可供选择的答案中，②是一个拓扑序列。
最短路径是两个结点间长度最短的路径。这里的路径长度是指路径上的边所带权的总和，而

不是路径上边数的总和。关键路径是指两个结点间具有的最大长度的路径。通过计算可知，(V1，V2，V5，V7，V8)是最短路径，(V1，V6，V5，V3，V8)是关键路径。
【答案】A：⑦ B：③ C：② D：④ E：②

试题 11(1992 年试题 6)

从供选择的答案中选出应填入{ }内的正确答案，把编号写在答卷的对应栏内。
在内排序的过程中，通常需要对待排序的关键码集合进行多遍扫描。采用不同排序方法，会产生不同的排序中间结果。设要将序列＜Q，H，C，Y，P，A，M，S，R，F，X＞中的关键码按字母的升序重新排列，则 A 是冒泡排序一趟扫描的结果，B 是初始步长为 4 的希尔(Shell)排序一趟扫描的结果，C 是两路归并(合并)排序一趟扫描的结果，D 是以第一个元素为分界元素的快速排序一趟扫描的结果，E 是堆排序初始建堆的结果。
供选择的答案
A～E：①F，H，C，D，P，A，M，Q，R，S，Y，X
②P，A，C，S，Q，D，F，X，R，H，M，Y
③A，D，C，R，F，Q，M，S，Y，P，H，X
④H，C，Q，P，A，M，S，R，D，F，X，Y
⑤H，Q，C，Y，A，P，M，S，D，R，F，X
【解析】
冒泡排序将待排序的记录顺次两两比较，如为逆序则进行交换，将待排序序列依此法从头至尾处理一遍称作一趟冒泡，一趟冒泡的效果是将关键码值最大的元素交换到了最后位置。对于试题给定的序列，经一趟冒泡后变成序列＜H，C，Q，P，A，M，S，R，D，F，X，Y＞。
希尔排序也是一种插入排序，其特点是把待排序序列每隔某个"步长"的元素组成一个子序列，分别对每个子序列进行直接插入排序，排序后的结果称为一趟扫视。每趟减少相隔步长，直至步长为 1，对整个序列作一次扫视。对于试题给定的序列，取初始步为 4，希尔排序一趟扫视后，将序列变成为＜P，A，C，S，Q，D，F，X，R，H，M，Y＞。
两路归并排序是一种最简单的归并排序，它将含 n 个元素的待排序序列看作 n 个已排序的子序列。首先将每两个子序列归并，得到 n/2 个已排序的较大的子序列；再对这些子序列归并，如此反复，直到最后归并成一个序列，排序即告完成。由上述两路归并的方法可知，对于试题所给的序列，经两路归并排序的一趟扫描后，序列变成＜H，Q，C，Y，A，P，M，S，D，R，F，X＞。
快速排序的基本思想是通过一趟扫视将待排序序列分成两个子序列，然后分别对这两个子序列进行排序，如此重复，以达到最后对整个序列的排序。具体方法是：任选待排序序列中的某元素(通常选第一个元素)，用它的关键码同(子)序列中所有其余元素的关键码相比较，将所有关键码仍比它小的元素移到它之前，所有关键码较它大的元素都移到它之后。经这样一趟扫视后，以该元素为界，将(子)序列分成两个子序列。反复对上述子序列作同样的操作，直至每个子序列都只有一个元素为止。对试题给定的序列，以它的第一个元素为分界元素作一次划分，使序列都只有一个元素为止。对试题给定的序列，以它的第一个元素为分界元素作一次划分，使序列变成＜F，H，C，D，P，A，M，Q，R，S，Y，X＞。
堆将待排序序列看作一棵完全二叉树，若其关键码序列
h1,h2,…，hn.
满足以下性质：
$h_i \le h_{2i}$，$h_i \le h_{2i+1}$，(i=1，2，…，［n/2］)
则称该序列是一个堆。即子树根结点 hi，其关键码值比它的左、右子结点 h2i($2i \le n$)和

$h_{2i+1}(2i+1\leq n)$的关键码值都要小。
堆排序的基本思想如下：由于堆的根结点的关键码值是最小的，取走根结点，让剩下的其余结点重新建成堆，则能从新堆的根结点取走次最小的。如此反复，得到一个递增的排序序列。堆排序的基本问题是如何建堆，建立初始堆的方法是首先将第 1 个至第 n 个结点分别看作是只有一个结点的堆，然后逐步把结点第［n-1］个、第［n/2］-1 个…第 1 个结点，分别加到它们的两棵左、右子树上构成堆，最终变成一个堆。
对于试题给定的序列，经上述初始建堆步骤后，序列变成＜A，D，C，R，F，Q，M，S，Y，P，H，X＞。
【答案】A：④ B：② C：⑤ D：① E：③

试题 12(1991 年试题 3)

从下列叙述中选出 5 条正确的叙述，把编号依次写在答卷的 A～E 栏内。
①m 阶 B-树每一个结点的后件个数都小于等于 m。
②m 阶 B-树每一个结点的后件个数都大于等于［m/2］。
③m 阶 B-树具有 k 个后件的非叶子结点含有 k-1 个键值。
④m 阶 B-树的任何一个结点左右子树的高度都相等。
⑤中序遍历一棵查找树的结点就可得到排好序的结点序列。
⑥用指针的方式存储一棵有 n 个结点的二叉树，最少要 n+1 个指针。
⑦任一查找树的平均查找时间都小于用顺序查找法查找同样结点的线性表的平均查找时间。
⑧平衡树一定是丰满树。
⑨已知树的前序遍历并不能唯一地确定这棵树，因为不知道树的根结点是哪一个。
⑩不使用递归，也可实现二叉树的前序、中序及后序遍历。
【解析】
m 阶 B-树是一种平衡的 m 叉树，具有如下的性质：
(1)每个结点的后件个数小于等于 m；
(2)除了根结点和叶结点之外，每个结点的后件个数大于等于(m/2)；
(3)具有 k 个后件的非叶结点含有 k-1 个键值；
(4)所有叶结点在同一层上，而且不附有信息。
m 阶 B-树是平衡树，其上任一结点的所有子树的高度都是相等的。
具有下列性质的二叉树称为查找树：除叶结点外，每个结点的键值大于其左子树上一切结点的键值，且小于等于其右子树上一切结点的键值。
中序遍历法访问二叉树结点的过程是：递归地访问左子树，然后访问树根，再访问右子树。因此，用中序遍历法遍历一棵二叉查找树可以得到按键值升序排列的结点序列。在用链接表表示法存储 n 个结点的二叉树上，最多有 2n 个指针，其中 n+1 个是空指针，n-1 个指针用于指向后件。因此，最少要 n+1 个指针的说法是不对的。
在给定的查找树上，由根结点到所有其他结点的路径长度总和称为内部路径长度。当一棵二叉树退化为线性链表时，它的内部路径长度最大，等于 n(n-1)/2，所以退化二叉树的查找时间(包括平均查找时间)等于线性表顺序查找时间。答案⑦是错误的。
具有最小内部路径长度的树是丰满树，对丰满树的查找树进行插入或者删除操作后，会产生一棵非丰满树。平衡二叉树是指其上任一结点的左右子树的高度(或者结点个数)保持一定比例的树，即平衡树上任一结点的左、右子树仍然保持平衡。平衡树的查找效率和丰满树相近，但是在插入或者删除结点时容易产生仍保持平衡的树。由丰满树同平衡树定义可知，丰满树一定是平衡树，但是平衡树不一定是丰满树。
用前序遍历法访问一棵树的原则是：递归地访问树根，然后访问左子树，再访问右子树。因

此，前序遍历第一个被访问的结点就是原树的根结点，"不知道树的根结点"的说法是错误的。二叉树的前序、中序和后序遍历法都是递归的。因此，最适合使用递归程序方法实现这 3 种遍历算法。当然不用递归程序方法也能够实现这 3 种遍历算法，但是程序比较复杂，一般需要用栈或队列操作来实现。

【答案】A：① B：③ C：④ D：⑤ E⑩

试题 13(1990 年试题 4)

从供选择的答案中选出应填入下列叙述中的{ }内的正确答案，把编号写在答案的对应栏内。

在查找算法，可用平均查找长度(记为 ASL)来衡量一个查找算法的优劣，其定义为：

$$ASL=\sum_{i=1}^{n}PiCi$$

此处 Pi 为表中第 i 个记录被查找的概率，Ci 为查找第 i 个记录时同关键字比较的次数，n 为表中表有记录数。

以下叙述中均假定每一个记录被查找的概率相等，即 Pi=1/n(i=1，2，…，n)。

当表中的记录连续存储备在一个一维数组中时，可采用顺序查找与折半查找方法(折半查找要求表是按关键字有序排列的)。顺序查找时的 ASL 为 A，折半查找时的 ASL 为 B。记录的关键字有序时，用二叉排序树查找记录，在最坏的情况下，ASL 为 C。当二叉排序树是一棵平衡树时，ASL 为 D。在平衡树上删除一个结点后可以通过旋转使其平衡，最坏的情形下需 E 次旋转。

供选择的答案

A～E：①O(1) ②O(lon^2n) ③O(lon^2n)2 ④O($nlog_2n$)

⑤O(n) ⑥O(n^2)

【解析】

顺序查找：比较一次可查到第 1 个元素，比较两次可查到第 2 个元素。一般地，比较 i 次可查到第 i 个元素。

顺序查找的 ASL 为：ASL=（1/n）*(n(n+1)/2) =n(n+1)/2=O(n)

折半查找：比较一次可找到一个元素；比较两次可找到两个元素；一般地，比较 i 次可找到 2i-1 个元素。总共 n 个元素，最多需比较 $log_2(n+1)$次。

折半查找的 ASL 为：ASL

$$ASL=\sum_{i=1}^{\log_2(n+1)}\frac{1}{n}\cdot i\cdot 2^{i-1}$$

而归纳法不难证明：

$$\sum_{i=1}^{k}i\cdot 2^{i-1}=2^k(k-1)+1$$

将此式入上式，折半查找的 ASL 为：

ASL=1/n(n+1)($log_2(n+1)$+1)=1/n(n+1)$log_2(n+1)$-1

=O(log_2n)

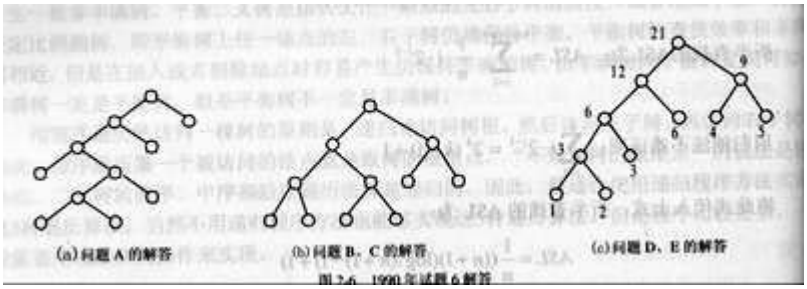
对于二叉排序树的查找，最坏情况是二叉树中没有结点，有两棵子树。在这种情况下，二叉排序树查找相当于顺序查找，即最坏情况下，ASLA 为 ASL=O(n)。

对于平衡的二叉排序树查找，平衡二叉排序树中的每个结点的左右子树高度差不大于 1，n

个结点的平衡树的深度和 $\log_2 n$ 是同数量级。对二叉排序树来说，它的查找时间与树的深度成正比。因此，平衡二叉排序树的 ASL 为 $ASL=O(\log_2 n)$。
在平衡树上删除或插入结点往往会破坏树的平衡。通过旋转又可使其重新成为一棵平衡树。在最坏情况下，从叶子开始一直调整到根为止，需进行的旋转次数与树的深度一致，等于 $O(\log_2 n)$次。
【答案】：A：⑤ B：② C：⑤ D：② E②

试题 14（1990 年试题 6）

从供选择的答案中选出应填入下列叙述中的{ }内的正确答案，把编号写在答卷的对应栏内。
设 T 是正则二叉树，它具有 6 片树叶，那么树 T 的高度最多可以是 A；最小可以是 B；树 T 的内结点数是 C；如果 T 又是哈夫曼(Huffman)最优树，且各片树叶的权分别是 1、2、3、4、5、6，则最优树 T 的非树叶的权之和是 D；权为 1 的树叶的高度是 E。(注：树的根结点高度为 1)
供选择的答案：
A、B、C、E：①7 ②5 ③6 ④4 ⑤3
D：①27 ②30 ③45 ④51 ⑤64
【解析】
这里要注意正则树的概念，所谓正则二叉树，就是每个非叶结点都有两个子结点。根据这个原则，我们用下面的 2-6(a)、2-6(b)、2-6(c)图来解答本题。



【答案】：A：③ B：④ C：④ D：④ E②

1.2 程序语言基础知识

1.2.1 主要知识点

了解程序语言的种类、特点和适用范围，掌握汇编、编译、解释系统的基本原理，以及程序语言的数据结构和控制结构。特别是编译系统形式的基础知识，应当下功夫掌握。

12.1.1 程序语言概述

低级语言又称面向机器语言，它是特定的计算机系统所固有的语言。用机器语言编制出来的,程序可读性很差，程序员难以修改和维护，于是人们考虑改用助记符号来表示机器指令中操作码和操作数，这就是汇编语言。汇编语言仍然是一种和计算机的机器语言十分接近的语言，它的书写格式在很大程度上取决于特定计算机的机器指令。它是一种低级语言，这对于人们抽象思维和交流十分不便。在这个基础上，高级语言就发展起来了。目前已有许多流行较广的高级语言，如 Fortra、Cobol、PascalC、和 C++等。这类语言与人们的自然语言比较接近，大大提高了程序设计的效率，便于人们进行交流。

计算机只能理解和执行机器语言。程序语言要在计算机上运行，必须有一个程序，使机

器能够理解用程序语言书写的用户程序，这就是所谓的语言处理程序。它可以分为两大类：解释程序和翻译程序。

1.2.1.2 程序语言基础知识

程序语言种类繁多，在应用上各有不同的侧重面。

Fortran 是第一个被广泛用于科学计算的高级语言。

Algol 是另一个早期研制出来的高级语言。它有严格的文法规则，用巴科斯范式 BNF 来描述语言的文法。Algol 是一个分程序结构的语言。

Cobal 是一种面向事务处理的高级语言。

Pascal 语言提供的为数不多而又相当紧凑的机制使得这个语言具有相当强的表达能力。

C 是一种通用程序设计语言。C 作为一种较低级的语言，提供了指针和地址操作的功能。C 提供书写结构良好的程序所需的控制结构。C 与 UNIX 操作系统紧密相关，UNIX 操作系统及其上的许多软件都是 C 编写的。

还有几种特殊而又重要的程序语言：

(1)面向对象的 C++

C++是在 C 语言的基础上发展起来的与 C 兼容的语言。主要增加了类功能，成为面向对象的程序设计语言。面向对象程序语言至少包含以下几个重要概念：

①对象是人们要进行研究的任何事物，它包括状态(用数据来描述)和操作(用来改变对象的状态)两方面。面对对象语言把状态和操作封装于对象体之中，并提供一种访问机制，使对象状态的具体表示和操作的具体实现都是隐蔽的。

②类是面向对象语言必需提供的用户定义的数据的类型，它将具有相同状态、操作和访问机制的多个对象抽象成一个对象类。在定义了类以后，属于这种类的一个对象叫作类实例或类对象。

③继承是面向对象语言的另一个必备要素。类与类之间可以组成继承层次，一个类的定义(称为子类)可以定义在另一个已定义类(称为父类)的基础上。子类可以继在父类中的属性和操作，也可以定义自己的属性和操作。

(2)逻辑型语言 Prolog

逻辑型语言是一类以形式逻辑为基础的语言。Prolog 是这类语言的代表。Prolog 建立在关系理论和一阶谓词理论基础上，具有和传统的命令型程序设计完全不同的风格。Prolog 程序由一些称为事实和规则的 Horn 子句组成。具有很强的推理功能，适用于书写自动定理证明、专家系统、自然语言理解等问题的程序。

(3)函数型语言 LISP

函数型程序语言是一类以 γ 演算为基础的语言。LISP 是典型的函数型程序语言。函数是一种对应规则(映射)，它使其定义域中每一个值和值域中唯一的值相对应。

函数型程序设计语言的优点之一是对表达式中出现的任何函数都可以用其他函数来代替，只要这些函数调用产生相同的值。由于用函数程序设计语言书写的程序是利用自变量的值来计算函数的值的，它没有副作用。这些特点有助于程序模块化的实现。

1.2.1.3 程序语言的数据类型

不同程序语言所提供的数据类型不尽相同。数据是程序操作的对象，具有名称、类型、存储类、作用域和生存期等属性，使用时要为它分配内存空间。数据名称由用户可通过标识

符命名：类型说明数据占用内存的大小和存放形式，存储类说明数据在内存中的位置和生存期；作用域说明数据可以使用的范围；生存期说明数据占用内存的时间。

数据从不同角度可分成不同的类别。按数据的作用域大小，可分为全局量和局部量；按生存期可分为自动生存期、静态生存期和动态生成期；按程序运行时数据的值是否能改变可分为常量和变量。

数据按类型可分为 4 种：Void、标量(Scalar)、函数和聚合(Aggregate)。标量又可分为算术、枚举和指针；聚合类型可分为数组、结构体和共用体。

数据按其构造方式又可分为基本类型和派生类型。

1.2.1.4 程序语言的控制结构

程序语言中控制结构为将数据和数据上的运算组合成程序提供了基本框架。可计算问题的程序都可用顺序、选择和循环这 3 种控制结构来描述。

1.2.1.5 汇编程序基础知识

汇编语言是为特定的计算机中计算机系统设计的面向机器的语言。汇编语言中的语句可分成两大类：一类是与机器指令相对应的可执行汇编语句；另一类汇编语句称为汇编控制语句(亦称伪指令)。用汇编语言编写的源程序，要通过汇编程序将它民机器语言程序，才能被计算机理解执行。经过汇编程序的工作，可执行汇编语句被转化成对应的机器指令；而伪指令并不翻译成机器指令，它们的作用是控制汇编程序工作。伪指令主要用来告诉汇编程序做一些除了翻译机器指令外必须做的工作。

汇编程序的功能是将汇编语言所编写的源程序翻译成由机器指令和其他信息组成的目标程序。汇编程序的基本工作包括两项：一是将每一条可执行汇编语句转换成对应的机器指令；二是处理源程序中出现的伪指令。

1.2.1.6 解释程序基础知识

解释程序是一种语言处理程序，它直接执行源程序或源程序的内部形成。它并不产生目标程序，它是它和编译程序的主要区别。

高级语言实现语言处理 4 种方案：

第 1 种，源程序被直接解释执行。

第 2 种，先将源程序翻译成高级中间代码，然后再扫描高级中间代码，对高级中间代码进行解释执行。

第 3 种，也是一种解释程序的实现方案，与第 2 种方案的解释程序不同点在于，首先将源程序转化成和机器代码十分接近的低级中间代码，然后再解释执行这种低纸中间代码。这类系统具有良好的可移植性。

第 4 种，是普通的编译程序。在编译程序方案下，高级语言编写的源程序被最终翻译成机器语言表示的目标程序。这类系统的目标执行效率最高。

一般说来，建立在翻译基础上的系统在执行速度上都优于建立在解释执行基础上的系统。翻译系统的缺点是其复杂性，这使得它的开发和维护费用都大。相反，解释系统比较简单，可称植性较好，适合于以交互方式执行程序，其缺点是执行速度慢。

解释系统的结构可分成两部分。第 1 部分包括通常用的词法分析程序以及语法和语义分析程序，它的作用仍是把源翻译成中间代码，中间代码的设计常采用逆波兰表示形式。第 2 部分是解释部分，用来对第 1 部分所产生的中间代码进行解释执行，完成真正的解释工作。

1.2.1.7 编译程序基础知识

编译程序的功能是把某高级语言书写的源程序翻译成与之等价的低级语言(汇编语言或机器语言)的目标程序。其过程很复杂，可分成 6 个阶段；

①词法分析阶段是编译过程的第 1 个阶段。词法分析所依据的是语言的词法规则，即描述单词结构的规则。词法规则可用 3 型文法(正规文法)或正规式来描述，有限自动机能识别正规文法所定义的语言和正规式所表示的集合。

②语法分析阶段。在词法分析的基础上将单词符合序列分解成各类语法单位。语法分析所依据的是语言的语法规则，即描述程序结构的规则。

词法分析和语法分析本质上都是对源程序的结构进行分析。

③语义分析阶段是审查源程序有无语义错误，为代码生成阶段收集类型信息。比如语义分析的一个工作是进行类型审查，审查每个算符是否具有语言规范允许的运算对象。

④中间代码生成阶段。在进行了上述的语法分析和语义分析阶段工作之后，有的编译程序将源程序变成一种内部表示形式，这种内部表示形式叫做中间语言或中间代码。所谓"中间代码"是一种简单、含义明确的记号系统。

语义分析和中间代码生成所依据的是语言的语义规则。一般采用语法指导翻译规则和中间代码生成规则。

⑤代码优化阶段是对前阶段产生的中间代码进行变换或进行改造，目的是使生成的目标代码更为高级，即省时间和省空间。

⑥目标代码生成阶段。是把中间代码变换成特定机器上的绝对指令代码或可重定位的指令代码或汇编指令代码。这是编译的最后阶段，它的工作与硬件系统的结构和指令的含义有关。

编译过程的 6 个阶段的任务，再加上表格管理和出错处理的工作可分别由几个模块或程序完成，它们分别称作词法分析程序、语法分析程序、语义分析程序、中间代码生成程序、代码优化程序、目标代码生成程序、表格管理程序和出错处理程序。

1.2.2 试题分析

从历年试题统计(见表 2-2)来看，程序语言基础试题是每年必考的知识点，考查的重点是形式语言基础、语法和词法分析方法、程序控制结构。复习中还应注意补充有关程序语言最新发展方面的知识。

高级程序员考试试题分类精解

表 2-2　历年高级程序员级程序语言基础试题统计

试题	考查知识点
1991 年试题 7	编译语法分析器
1992 年试题 2	程序语言常识
1993 年试题 7	文法
1994 年试题 2	子程序调用、程序控制结构（递归和循环）
1994 年试题 16	文法
1995 年试题 15	短语结构文法
1996 年试题 5	有限状态自动机（词法分析）
1997 年试题 1	子程序参数调用
1997 年试题 7	程序语言常识
1998 年试题 4	文法语法分析、算符优先文法
1998 年试题 6	子程序参数调用
1999 年试题 3	文法
2000 年试题 3	文法

试题 1 (2000 年试题 3)

从供选择的答案中，选出应填入下面叙述中{ }内的最确切的解答，把相应编号写在答卷的

对应栏内。

假设某程序语言的文法如下：

S→a|b|(T)

T→TdS|S

其中：Vt={a,b,d,(,)},Nn={S,T}，S 是开始符号。

考查该文法，称句型(Sd(T)db)是 S 的一个 A。其中 B 是句柄；C 是素短语；D 是该句型的直接短词；E 是短语。

供选择的答案

A：①最左推导 ②最右推导 ③规范推导 ④推导

B：①S ②b ③(T) ④Sd(T)

C：①S ②b ③(T) ④Sd(T)

D：①S ②S，(T)，b

③S，(T)，TdS，b ④(Sd(T)db)

E：①(Sd(T)db) ②d(T) ③Td ④Sd(T)d

【解析】

解答本题要搞清楚基本概念，下面具体分析各个问题。

先来看问题 A。最左(右)推导：任何一步推导过程 σ→β(其中 σ、β 是句型)都是对 σ 中的最左(最右)非终结符进行替换，这种推导为最左(最右)推导。在形式语言中，最右推导常被称为规范推导。

题中的句型(Sd(T)db)的第一步肯定是由 S→(T)→(TdS)得出的。按照最左推导的规则(Tds)→(TdSdS)→(SdSdS)，最终不可能推出原来的句型。

按照最右推导的规则(Tds)→(Tdb)→(Td(T)db)，最终不可能推出原先的句型。

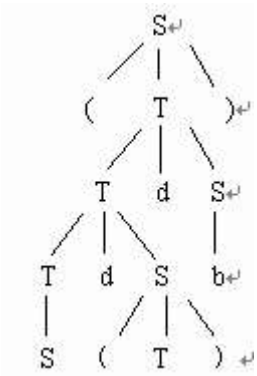
最后可以看出句型(Sd(T)db)是由一般推导推出的，步骤如下：

S→(T)→(Tds)→(Tdb)→(Td(T)db)→(Sd(T)db)

所以正确答案是④。

再来看问题 B～E。来文法，S 是文法的开始符号，αβδ 是文法 G 的一个句型。如果有 S→αAδ，且 A→β，则称 β 是句型 αβδ 相对于非终符 A 的短语。特别是如有 A　β，则称 β 是句型 αβδ 相对于规则 A→β 的直接短语。一个句型的最左直接短语称为该句型的句柄。

本文法推导树如下：



所以，S 是句型相对于规则 T→S 的直接短语，也是最左直接短语(句柄)。(T)是句型相对于规则 S→(T)的直接短语，对于问题 B，选择①是正确的。

素短语是一个短语，它至少包含一个终结符，并除自身外不包含其他的素短语。所以，问题 C 的答案③正确。

d 是句型相对于规则 S→d 的直接短语，则问题 D 的答案②正确。

由推导树可知，无论如何，无法由 S 推导出 d(T)、Td 或 Sd(T)d，所以问题 E 的答案①正确。
【答案】： A：④ B：① C：③ D：② E：①

试题 2（1999 年试题)3

从供选择的答案中，选出应填入下面叙述中{ }内的最确切的解答，把相应编号写在答卷扔对应栏内。
假设某程序语言的文法如下：
S→SaT|T
T→TbR|R
R→PdR|P
P→fSg|e
其中 Vr{a,b,d,e,f,g};Vn={S,T,R,P};S 是开始符号，那么，此文法是 A 文法。这种文法的语法分析通常采用优先短阵。优先短阵给出了该文法中各个终结符之间的优先关系(大于、小于、等于和无关系)。在上述文法中，某些终结符之间的优先关系如下：
b{B}a; f{C}g; a{D}a;d{E}d。
供选择的答案
A：①五则文法 ②算符文法 ③二义文法 ④属性文法
B：①大于 ②小于 ③等于 ④无关系
C：①大于 ②小于 ③等于 ④无关系
D：①大于 ②小于 ③等于 ④无关系
E：①大于 ②小于 ③等于 ④无关系
【解析】
所谓算符文法，可以作如下的描述：如果在一个文法 G 中，不含有形如"U→…AB…"的产生式，其中 A,B∈Vn，则 G 为算符文法。也就是说，如果 G 是一算符文法，那么 G 的任何产生式的右部都不会出现两个非终结符号相邻的情况，而且，对算符文法而言，也不会含有两个非终对符号相邻出现的句型。这种性质意味着，如果把终对符号看作广义运算符，而把非终结符号看作广义运算的对象，则在算符文法的任何句型中，两相邻运算符之间的运算对象至多只有一个，而不会出现其间运算对象个数不确定的情况，这样就使得广义运算总是按照中缀形式出现，对语法分析工作非常有益。
对于给定的文法 G，可以逐个检查 G 的各产生式，查看它们的右部是否含有相邻出现的非终结符号，以确定 G 是否为一算符文法，然后再构造相应的优先矩阵。若此矩阵中无多重定义的元素(即各运算符对之间至多只有一种优先关系)，同则可确认理一算符优先文法。
在算符文法中，一般按照如下规则判断终结符之间的优先关系：
当且仅当 G 中有形如"U→…ab…"或者"U→…aBb…"的产生时，a=b；
当且仅当 G 中有形如"U→…aA…"的产生式，且有或者"$A=^{+}=>b$…"或者"$A=^{+}=>$…aB"时，a＞b；
当且仅当 G 中有形如"U→…Ab…"的产生式，且有或者"$A=^{+}=>a$…"或者"$A=^{+}=>$…aB"时，a＞b。
如果算符文法 G 的任何一对终结符号之间，至多只有 3 种算符优先关系：等于、大于或者小于成立，则称 G 为算符优先文法。
【答案】：A：② B：① C：③ D：① E：②

试题 3（1998 年试题 4)

从供选择的答案中，选出应填入下面叙述中{ }内的最确切的解答，把相应编号写在答卷的

对应栏内。
语法分析方法大体上可分成自上而下和自下而上的两种。自下而上分析法，是从输入符号串开始逐步进行 A，直至 A 成文法的起始符号。自上而下分析法，则是从文法的起始符号开始反复使用产生式进行 B 直至 B 出输入符号串。
符符优先文法是一种自下而上分析方法，其文法的特点是文法产生式中 C。自下而下的分析方法，通常要求文法的产生式 D，如 E 文法就是一种可以自上而下分析的文法。
供选择的答案
A、B：①递归 ②综合 ③回归 ④推导 ⑤分解 ⑥归约
C：①不含两个相邻的非终结符 ②不含两个相邻的终结符
③不含 产生式 ④不含长度为 1 的产生式
D：①不以非终结符开头 ②不以终结符开头
③不含左递归 ④不含右递归
E：①LR(1) ②LL(1) ③SLR(1) ④LALR(1)
【解析】
语法分析的程序以词法分析程序所输出的用内部编码表示的单词序列为输入，其任务是分析源程序的结构，差别它是否是相应程序设计语言的一个合法程序。为了完成这个任务，通常由语法分析程序尝试着为其构造一棵完整的语法树。若尝试成功，则表明输入的符号串在结构上是一个合乎语法的程序，否则，源程序中必然存在的错误。
就产生语法树的方向而言，可大致把它们分为自顶向下和自底向上两大类。所谓自顶向下的分析是对给定的符号串，试图自上而下地为其构造出一棵语法树，或者说从文法的开始符号出发，为其构造一个最左推导。所谓自底向上的分析是给定的符号串，试图自下而上地为其构造出一棵语法树，或者说从给定的符号串本身出发，试图将其归约为文法的开始符号。
算符优先文法属于自下而上的分析法，它利用各个算符间的优先关系和结合规则来进行语法分析，特别适用于分析各种表达式。算符优先文法的任何产生式的右部都还会出现两个非终结符相领的情况，且其任何一对终对符之间至多只有 3 种算符关系"＜"、"＞和"="之一成立。自顶向下的文法中不能有左递归，否则自顶和下的分析过程进入死循环，不能正常进行。E 中 IL(1)为自顶向下的文法外，其他均为自底向上的文法。
【答案】：A：⑥ B：④ C：① D：③ E：②

试题 4（1998 年试题 6）

从供选择的答案中，选出应填入下面叙述中{ }内的最确切的答案，把相应编号写在答卷的对应栏内。
在高级程序设计语言中，使用参数是子程序之间传递信息的一种手段。子程序说明中的参数称为形式参数，调用语句中的参数称为实在参数。调用时，实在参数的个数、类型和顺序要和形式参数保持一致。
知道一种语言(或编译器)使用哪种参数传递方法是很重要的，因为程序的运行依赖于所用的方法。参数传递方法有传值调用(Call by value)、引用调用(Call by reference)、传名调用(Call by name)和宏扩展(Macro expansion)。
传值调用是把实在参数的 A 传递给相应的形式参数，子程序通过这种传值形参 B；引用调用是指把实在参数的 C 传给相应的形式参数，此时子程序对形式参数的一次引用或赋值被处理成对形式参数的 D 访问。
C 语言中的函数，以 E 方式进行参数传递。
供选择的答案
A、C：①地址 ②名 ③值 ④地址和值

⑤值和名 ⑥名和地址
B：①可传回结果的值 ②可传回存放结果的地址
③可传回结果的值和存放结果的地址 ④不可传回任何结果(值或地址)
D：①直接 ②间接 ③变址 ④引用或赋值
E：①传值调用 ②引用调用 ③传名调用 ④宏扩展
【解析】
传值调用是指把实在参数的值传递给相应的形式参数，子程序不能通过这种方式传回任何结果。引用调用是指把实在参数的地址传递给相应的形式参数，此时子程序对形式参数的一次引用或赋值都是对形式参数的间接访问。
C 语言规定，实参变量对形式参数的数据传递是"值传递"，即单向传递，只能由实参传给形参，而不能由形参传给实参。所以说，C 语言中函数是传值调用的。
【答案】A：③ B：④ C：① D：② E：①

试题 5（1997 年试题 1）

从供选择的答案中，选出应填入下面叙述中 内的最确切的答案，把相应编号写在答卷的对应栏内。
一种最早用于科学计算的程序设计语言是 A；一种提供指针和指针操作且不存在布尔类型的、应用广泛的系统程序设计语言是 B；一种适合在互联网上编写程序可供不同平台上运行的面向对象程序设计语言是 C；一种在解决人工智能问题上使用最多的、有较强的表处理功能的函数程序设计语言是 D；一种以谓词逻辑为基础的、核心是事实、规则和推理机制的实用逻辑程序设计语言 E。
供选择的答案
A～E：①Pascal ②Ada ③Smalltalk ④Snobol ⑤C
⑥Alogo 68 ⑦Java ⑧Lisp ⑨Prologo ⑩Fortran
【解析】
程序设计语言从机器语言、汇编语言到今的高级语言，其发展越来越快，功能越来越强，同时，其可理解性也越来越接爱人类的思维方式。
Fortran 是第一种被用于进行科学计算的高级语言，它出现于 50 年代中期，其设计目的主要暖和于科学计算，它在程序设计语言的发展史上起着突出的作用。
Alogo 68 是另一种早期研制出来的高级语言，虽然没有被广泛使用，但对后来的程序设计语言的发展有着重大影响。
C 语言是目前应用最广泛的系统程序设计语言，它提供了指针与指针操作，且不存在布尔类型，对数据的访问灵活广泛，C 语言还可提供很多底层系统调用，与硬件结合紧密，易用性强。
Java 是适应 Internet 发展的需要而产生的通用网络程序设计语言，它提供了更好的网络安全性和平台无关性，并且采用了面向对象的原理，便于扩展，适合在互联网上编写可供不同平台上运行的面向对象程序设计语言。
Prolog 语言在人工智能领域应用较广泛，是一种以谓词逻辑为基础，借助于推理规则从已有事实推出新的事实的实用的逻辑程序设计语言。
Ada 和 smalltalk 两种语言引入了一定的封装机制，实现了信息隐藏。Ada 的主要封装机制为程序包；Smalltalk 是一种纯面向对象程序设计语言，它侧重于动态链接，不进行任何类型检查，并且类和对象之间没有明显区别。
Snobol 于 1962 年由贝尔实验室开发，它主要的操作为字符串操作。该语言一般不作为通用语言使用，而是作为一种研究工具，用于对文艺作品、音乐等进行分析。

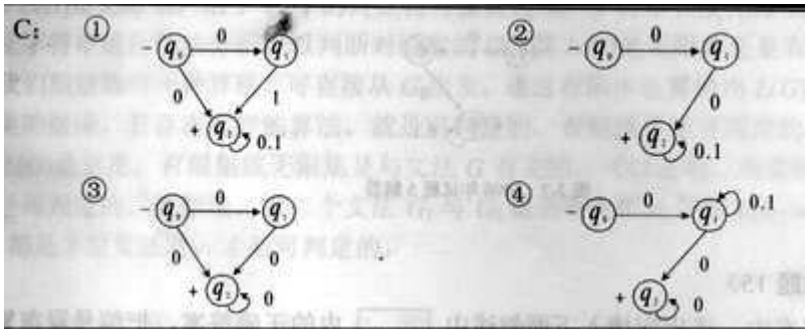
Lisp 是一种在解决人工智能问题上使用最多的、有强的表处理功能的函数程序设计语言。
【答案】A：⑩ B：⑤ C：⑦ D：⑧ E：⑨

试题 6（1997)年试题 7

从供选择的答案中，选出应填入下面叙述中{ }内的最确切的答案，把相应编号写在答卷的对应栏内。
用高级语言编写程序时，子程序调用语句中的实在参数必须与子程序说明中的形式参数在 A 上保持一致。在允许子程序递归调用的高级语言环境中，需用动态存储管理方法，它通常使用一个 B 存放子程序的调用记录，调用记录可包括(1)全局量存储区域的 C；(2)调用点所在子程序的 D；(3)调用点的 E；(4)形式参数和实在参数的通信区域；(5)返回值；(6)本子程序的局部量和临时变量存储区域等。
供选择的答案
A：①个数、类型 ②个数、顺序
③个数、格式、顺序 ④个数、类型、顺序
B：①线性表 ②队列 ③堆 ④下推栈
C～E：①子程序首地址 ②调用记录首地址
③参数地址 ④寄存器地址
⑤返回地址 ⑥开始地址
【解析】
除少数程序语言之外，形式参数和实在参数之间的对应关系通常按位置来确定。子程序定义中有形式参数表，而子程序调用是用实在参数法表。进入子程序时，第一个形式参数就和第一个实在参数所表示的数据或其他信息相关联，依奖对应。这样，子程序调用语句中的实在参数必须与子程序说明中的形式参数在个数、类型和顺序上保持一致，否则就不能在主程序和子程序之间正确地传递信息。
对于允许子程序递归调用的程序语言，一般采用动态存储管理方法，用下堆栈来实现。由于某一子程序可能被调用了若干次，但只有最近一次调用正处于执行状态，而其余各次调用正等待下次调用的返回。这样，前几次该子程序调用的属于该子程序的局部变量存储区中的内容必须保存起来，以便下次调用返回时再继续使用。子程序执行需要一个用来存放有关信息的区域，这个区域为过程的调用记录，一般包含以下几部分：
·全局变量存储区域的开始地址；
·调用点所在子程序的调用记录的地址；
·调用点机器状态，如返回地址和寄存器当时值；
·形式参数和实在参数的通信区域；
·返回值；
·本子程序的局部量和临时变量存储区。
【答案】：A：④ B：④ C：⑥ D：② E：⑤

试题 7（1996 年试题 5)

从供选择的答案中，选出应填入下面叙述中 内的最确切的解答，把相应编号写在答卷的对应栏内。
有限状态自动机可用五元组(VT，Q，δ，q0，Qf)来描述，它可对应于 A。设有一有限状态自动机 M 的定义如下：
VT={0，1} Q={q0,q1,q2}
δ 定义为：

δ(q0,0)=q1,δ(q1,0)=q2
δ(q2,1)=q2,δ(q2,0)=q2
Qf={q2}
M 是一个 B 有限状态自动机，它所对应的状态转换图为 C，它所能接受的语言可以用正则表达式表示为 D，其含义为 E。
供选择的答案
A：①0 型文法 ②1 型文法 ③2 型文法 ④3 型文法
B：①歧义的 ②非歧义的 ③确定的 ④非确定的
C：


D：①(0|1)* ②00(0|1) ③(0|1)*00 ④0(0|1)*0
E：①由 0 和 1 所组成的符号串的集合。
②以 0 为头符号和尾符号、由 0 和 1 所组成的符号串的集合。
③以两个 0 为结束的、由 0 和 1 所组成的符号串的集合。
④以两个 0 为开始的、由 0 和 1 所组成的符号串的集合。
【解析】
先来看一下地分类表 2-3

表2-3　Chomsky 文法分类表

文法类型	文法名称	语言名称	对应的自动机
0型（PSG）	短语结构文法	递归可枚举语言	图灵机（Turing）
1型（CSG）	前后文相关文法	前后文有关语言	线性限界自动机
2型（CFG）	前后文无关文法	前后文无关语言	非确定下推自动机
3型	正规文法	有限状态语言	有限状态自动机

一个有限状态自动机 M 用 5 元组(VT，Q，δ，q0，Qf)来表示，其中 VT={0，1}是一个有穷字母表，它的每个元素称为一个输入字符：Q=(q0,q1,q2)是一个有限集，它的每个元素称为一个状态；δ 定义为 δ(q0,0)=q1,δ(q1,0)=q2,δ(q2,1)=q2,δ(q2,0)=q2 是一个从 VT×Q 到 Q 上的(单值)部分映射；q0 为初态；Qf={q2}为终态集。
有限自动机分为确定的有限自动机和非确定的有限自动机。确定的有限自动机的确定性表现在映射 δ：VT×Q→Q 是一个单值函数，即对任何状态 q　Q 和输入字符 a　VT，映射 δ(q,a)唯一确定下一个状态。由分析可知本题给出的是一个确定的有限自动机，它的状态转换如图 2-7 所示。
它所能接受的语言可以用正则表达式表示为 00(0|1)*，其含义为由两个 0 开始的后跟任意个(包含 0 个或多个)0 或 1 组成的符号串的集号。
【答案】：A：④ B：③ C：/ D：/ E：/
(注：原题中 C 的供选择答案有印刷错误，故 C、D、E 的任何解答均作正确处理。)

试题 8（1995 年试题 15)

从供选择的答案中，选出应填入下面叙述中{ }内的正确答案，把编号写在答卷的对应栏内。

编译技术涉及语言和自动机理论，形式语言可由短语结构文法 G 生成(记为 L(G))，G 可分成 0 型、1 型、2 型、3 型和 4 种类型，与之相对应的自动机分别为图灵机、A、B、C。一个命题的可判定性是指：存在一种算法能给出该命题成立与否的结论。给定文法 G，只有当 G 为 D 时，命题"L(G)是空集，有限集或无限集"才是可判定的，当给出二个不同文法 G1 和 G2，只有当 G1、G2 都是 E 时命题"L(G1)=L(G2)"才是可判定的。

供选择的答案

A～C：①自动机 ②下推自动机 ③双向自动机

④线性界限自动机 ⑤有限状态自动机 ⑥非线性自动机

D～E：①1 型 ②2 型 ③3 型 ④0 型

⑤2 型或 3 型 ⑥1 型或 2 型或 3 型 ⑦0 型或 1 型或 2 型或 3 型

【解析】

用计算机对自然语言进行完全自动处理是一件十分困难的事情。这是因为自然语言歧义性哟，用形式化的语法描述起来很困难。为了便于计算机的自动处理，语言的形式化描述便显得十分重要。现有算法语言在形式上都是形式语言。

形式语言可由短语结构文法来生成。一个短语结构文法 G 是一个 4 元组：G=(VT,VN,P,S)。其中 VT 是终止符的非空有限集，VN 是非终止符的变量的非空有限集，S 为起始符，P 是产生式集。P 中的每个产生式又称重写规则，其一般形式是 α→β，其中 α、β 均为字符串。由起始符开始，不断使用 P 中产生式，可能得到终止符组成的字符串。这种终止符串的全体就是文法 G 生成的语言，记为 L(G)。

按照对产生式中 α 和 β 的不同限制，文法 G 可分成 0 型、1 型、2 型、3 型 4 类，又分别称图灵文法、上下文有关文法、上下文无关文法、正则文法。由文法所生成的语言也相应分成这 4 类。自动机是能由文法 G 产生出 G 所生成的语言 L(G)的一种装置，它可用一个 5 元组表示。自动机也可分好多种，其中有限状态自动机是能由正则文法产生正则语言的自动机，即它对应于 3 型语言；下推自动机则对应于上下文无关语言(2 型)；线性界限自动机则对应于上下文有关语言(1)型；图灵机则对应于 0 型语言。

文法 G 是任意给出的，有可能出现这样的情况：给定某个文法 G，对 VT 中的终止符所组成的任何字符串都无法识别出它能由 G 生成，即 L(G)是个空集。也可能有某个给定的文法 G，其 L(G)是无限制。由于 VT 的终止符可重复出现，字符串长度无限制，因此不可能用列举终止字符串进行句法分析，以判断给定的 G，其 L(G)是无限集还是有限集，或是空集。

我们期望能有一种算法，可直接从 G 出发，通过有限步运算给出 L(G)是空集、有限集或无限集的结论。若存在这样的算法，就是可判定的，否则就不是可判定的。研究中发现可否判定 L(G)是空集、有限集或无限集与文法 G 有关的，可以证明，当文法 G 是 2 型或 3 型时，是可判定的。类似的，对二个文法 G1 与 G2 是否等价即是否有 L(G1)=L(G2)，只有当 G1 和 G2 都是 3 型文法时，才是可判定的。

【答案】A：④ B：② C：⑤ D：⑤ E：③

试题 9（1994 年试题 2）

从供选择的答案中，选出应填入{ }内的正确答案，把编号写在答卷的对应栏内。

在下列程序中：

```
Program test (input, output)；
Var i, j: integer；
Procedure calc (pl, p2: integer)；
Begin p2:=p2*p2; pl:=p1-p2:=p2-p1;end{calc}
Begin{main}i:=2,j:=3
```

Calc(i,j);write(j);
End{main}
当参数传递采用引用方式(Call by reference)时，所得结果 j=A；
当参数传递采用换名方式(Call by name)时，所得结果 j=B；
当参数传递采用赋值方式(Call by value)时，所得结果 j=C；
递归是程序设计中很重要的一种控制结构，通常实现递归时，采用的数据结构是 D。
对那些既可以用递归方式，也可以用循环方式求解的问题，就执行效率而言 E。
供选择的答案
A～C：①0 ②3 ③5 ④6
⑤10 ⑥16 ⑦20 ⑧28
D：①数组 ②栈 ③队列 ④循环链表
E：①数组 ②两者相同 ③循环优于递归 ④递归优于循环
【解析】
一个过程的过程体若包含对其自身的调用，则称此过程是直接递归的。若一个过程的过程体调用某过程，而该过程又调用原过程或经一系列调用后又回到对原过程的调用，则称此原过程是间接递归的。通常实现递归时采用的数据结构是栈，这是因为栈有先进后出的特性，可以保存调用时的"现场"，并在调用结束时恢复"现场"，栈是实现递归的简单途径。对于既可用递归方式求解，也可用循环方式求解的问题，就执行效率和资源而言，显然是循环优于递归，因为递归的开销大。
当用户在调用点调用一个过程时，会通过参数传送信息，一个过程的形式参数用来向过程传送信息的标识符，实在参数用来在调用点向被调用过程传送信息。形式参数和实在参数之间的关系通常按位置来标定，不同程序语言所规定的参数信息传送方式不同。
当采用引用方式(Call by reference)或换名方式(Call by name)时，在过程中对形式参数的调用本质上是对实在参数单元的引用。先是给形式参数赋初值，而后，在过程中对该形式参数的赋值最终引起调用程序中实在参数值的改变。在本题中形式参数为 p1 和 p2。实在参数初值为 i=2 和 j=3，通过引用方式调用这两个参数，将执行以下计算过程：
p1=2,p2=3　p2:=p2*p2=9　p1:=p1-p2=2-3=-7　p2:=p2-p1=9-(-7)=16
所得结果为 j=16。
参数传送采用赋值方式时，从调用点向被调用过程传送的是实在参数的值。这一值成为过程中相应位置上形式参数的初值，此后该形式参数在过程中实际是局部变量，其结果无需返回给实在参数。本题中实在参数 j=3，在过程中仅起向形式参数 p2 赋初值的作用。过程中关于 p2 的运算对 j 不再起作用，因而过程调用结束后 j 的值仍为 3。
【答案】A：⑥ B：⑥ C： ②D： ②E：③

试题 10（1994 年试题 16）

从供选择的答案中，选出应填入下面有关形式语言叙述中{ }内的正确答案，把编号写在答卷的对应栏内。
文法 G=(VT,VN,P,S)的类型由 G 中的 A 决定。若 G0=({a,b},{S,X,Y},P,S)，P 中的产生式及其序号如下：
1：S→XaaY
2：X→YY|b
3：Y→XbX|a
则 G0 为型文法，对应于 C，由 G0 推导出句子 aaaa 和 baabbb 时，所用产生式序号组成的序列分别为 D 和 E。

供选择的答案：
A：①：VT ②VN ③P ④S
B：①0 ②1 ③2 ④3
C：①图灵机 ②下推自动机 ③有限状态自动机 ④其他自动机
D、E：①13133 ②12312 ③12322 ④12333
【解析】
形式语言的文法是一个 4 元组 G=(VT，VN，P，S)，其中 VT 是非空有限集，称为终端符集，VN 也是非空有限集，称为变量集；P 为产生式集；S 为起始符，S∈Vn。形式语言的文法按 P 的特性可分成正则文法、上下文无关文法、上下文有关文法和图灵文法 4 种类型，又分别称为 3 型、2 型、1 型和 0 型文法。所对应的可实现的自动机分别为有限状态自动机、下推自动机、线性有界自动机和图灵机。
本题中给出的文法、生产式左部均是单个变量，因此是上下文无文法。由此文法导出句子 aaaaa 的产生式号的序列及推导过程如下：
S → XaaY 1
→ YYaaY 2
→ aYaaY 3
→ aaaaY 3
→ aaaaa 3
句子 baabbb 的推导过程为：
S → XaaY 1
→ baaY 2
→ baaXbx 3
→ baabbx 2
→ baabbb 2
因而产生式号的序列是 12333 和 12322。
【答案】：A：③ B：③ C：② D：④ E：③

试题 11（1993 年试题 7）

从供选择的答案中，选出应填入{ }内的正确答案，把编号写在答卷的对应栏内。
根据乔姆斯基 50 年代建立的形式语言的理论体系，语言的文法被分成 4 种类型，即 0 型(短语文法)、Ⅰ(上下文有关文法)、Ⅱ型(上下文无关文法)和Ⅲ型(正规文法)。其中Ⅱ型文法与 A 等价，所以有足够的能力描述多数现今程序设计语言的语法缃结构。一个非确定的有限自动机必存在一个与之等价的 B。从文法描述语言的能力来说 C 最强，D 最弱。由 4 类文法的定义可知 E 必是Ⅱ型文法。
供选择的答案
A、B：①确定的有限自动机 ②图灵机
③非确定的下推自动机 ④非确定的有限自动机
⑤有限自动机 ⑥线性有界自动机
C～E：①0 型文法 ②Ⅰ型文法 ③Ⅱ型文法 ④Ⅲ型文法
【解析】
Ⅱ型文法与非确定的下推自动机等价，所以有足够的能力描述现在大多数程序设计语言的语法结构。一个非确定的有限自动机必存在一个与之等价的有限自动机。从文法描述语言的能力来说，0 型文法最强，Ⅲ型文法最弱。
语言的文法是一个 4 元组(Vt，Vn，P，S)，其中 Vt 是终结符号的非空有限集，Vn 是非终结

符号的非空有限集，S 是一个特殊的非终对符号，P 是产生式的有限集，由III型文法的定义：一个文法 G 是III型文法，如果 G 是 II 型文法，并且 G 的每个产生式为 A→σB 或 A→σ，其中 A∈Vt*，A，B∈Vn，可知，III型文法必是 II 型文法。
【答案】A：③ B：① C：① D：④ E：④

试题 12（1992 年试题 2）

从供选择的答案中选出应填入{ }的正确答案，把编号写在答卷的对应栏内。
最早体现结构化程序设计思想的程序设计语言是 A，最早使用 BNF 文法定义程序设计语言语法的语言是 B。最早提出类(即 CLASS)的概念语言是 C，了早完备地体现面向对象并提出继承概念的程序设计语言是 D，最早的人工智能语言是 E。
供选择的答案：
A～E：①Ada ②Pascal ③Algol 68 ④Algol 60 ⑤Simula ⑥Lisp ⑦Prolog ⑧Smalltalk 80 ⑨C ⑩C++
【解析】
在 50 年代末和 60 年代初，由国际委员会设计了 Algol 语言，P.Naur 在随后的修改报告中用严格的 BNF(Backus-Naur Form)形式化方法定义了 Algol 语言的文法。
LISP 语言是由麻省理工学院的 John McCarthy 和他的研究小组最初约在 1960 年设计实现的。结构化程序设计思想是由 EW.Dijkstra 和 C.A.R.Hoare 等在近 60 年代末提出的。N.Wirth 提出的 Pascal 语言是第一个较系统地体现结构化程序设计思想的程序设计语言。
最早提出类(CLASS)概念的语言是 Simula；最早完备地体现面向对象并提出继承概念的程序设计语言是 Smalltalk 80；Lisp 是一种计算机的表处理语言，是最早也是应用最广泛的人工智能语言。
【答案】A：② B：④ C：⑤ D：⑧ E：⑥

试题 13（1991 年试题 7）

从供选择的答案中选出应填入列下叙述中 内的正确答案，把编号写在答卷的对应栏内。编译程序中语法分析器接受以 A 为单位的输入，并产生有关信息供以后各阶段使用。B、LR 分析法和 C 是几种常见的语法分析技术，其中 B 和 LR 分析法属于自下而上分析法，而 C 属于自上而下分析法。LR 分析法主要有 SLR(1)、LR(0)、LR(1)和 LALR(1)4 种，其中 D 的分析能力最强，E 的分析能力最弱。
供选择的答案
A：①：表达式 ②单词 ③产生式 ④语句
B、C：①递归下降法 ②算符优先法 ③语法制导翻译法
④数据流分类法 ⑤自动机分析法
D、E：①SLR(1) ②LR(0) ③LR(1) ④LALR(1)
【解析】
编译程序中语法分析器接受以单元为单位的输入，并产生有关信息供以后各阶段使用。算符优先法、LR 分析法和递归下降法是几种常见的语法分析技术，其中算符优先法和 LR 分析法属于自下而上分析法，而递归下降法属于自下而下分析法。LR 分析法主要有 SLR(1)、LR(0)、LR(1)和 LALR(1)等 4 种，其中 LR(1)的分析能力最强，LR(0)的分析能力最弱。
【答案】A：② B：② C：① D：③ E：②

1.3 操作系统基础知识

1.3.1 主要知识点

掌握操作系统的类型、功能、层次结构和进程概念，以及作业、处理机、存储、文件和设备等管理的原理和方法。

1.3.1.1 操作系统类型和功能

根据使用环境和对用户作业的处理方式划分，操作系统的基本类型可以分为批处理操作系统、分析操作系统和实时操作系统 3 大类型。

分时操作系统使多个用户同时以会话方式控制自己程序的运行，每个用户都感到似乎各自有一台独立的、支持自己请求服务的系统。

实时系统往往是专用的，系统与应用很难分离，常常紧密结合在一起。实时系统并不强调资源利用率，而更关心及时性(时间紧迫性)、可靠性和完整性。实时系统又分成实时过程控制与实时信息处理两种。

网络环境下的操作系统又分成网络操作系统和分布式操作系统。分布式操作系统要求一个统一的操作系统，负责全系统的资源分配和调度，为用户提供统一的界面。它是一个逻辑上紧密耦合的系统。而网络操作系统用户则需指明欲使用哪一台计算机上的哪个资源。

操作系统主要有 5 个功能模块：处理器管理、存储管理、设备管理、文件管理和用户接口。

1.3.1.2 进程和进程管理

(1)进程

进程是一个程序关于某个数据集的一次运行。也就是说，进程是运行中的程序，是程序的一次运行活动。相对于程序，进程是一个的概念，而程序是静态的概念，是指令的集合，因而进程具有动态性和并发性。

在操作系统中进程是进行系统资源分配、调度和管理的最小单位，注意，现代操作系统中还引入了线程(Thread)这一概念，它是处理器分配资源的最小单位。

(2)进程的状态及其转换

多道系统中，进程的运行是时走时停的。它在处理器上的交替运行，使它的运行状态不断地变化着，最基本的状态有 3 种，即运行、就绪和阻塞。

·运行：正占用处理器。

·就绪：只要获得处理器即可运行。

·阻塞：正等待某个事件的发生。

(3)进程控制块

进程是一个动态的概念，在操作系统中，引入数据结构--进程控制块(简记为 PCB)来标记进程。PCB 是进程存在的唯一标志，PCB 描述了进程的基本情况。从静态的观点看，进程由程序、数据和进程控制块组成；从动态的观点看，进程是计算机状态的一个有序集合。

程序是进程运行所对应的运行代码，一个进程对应于一个程序，一个程序可以同时对应于多个进程，这个程序代码在运行过程中不会被改变，常称为纯码程序或可重入程序，他们

是可共享的程序。

进程控制块保存进程状态、进程性质(如优先程度)、与进程有关的控制信息(如参数、信号量和消息等)、相应队列和现场保护区域等。进程控制块随着进程的建立而产生，随着进程的完成而撤消。

PCB 是操作系统核心中最主要的数据结构之一，它既是进程存在的标志和调度的依据，又是进程可以被打断并能恢复运行的基础。操作系统核心通过 PCB 管理进程，一般 PCB 是常驻内存的，尤其是调度信息必须常驻内存。

(4)进程管理

在操作系统中有许多进程，它们对应着不同的或相同的程序，竞争地使用着系统的资源。进程管理涉及到进程控制、队列管理和进程调度等。

进程的生命过程从它被创建时开始，直至任务终止而撤消，其间会经历各种状态的转换，它们都是在操作系统控制下完成的。操作系统提供了对进程的基本操作，也称为原语。这些原语包括创建原语、阻塞原语、终止原语、优先级原语和调度原语。

进程调度即处理器调度，它的主要功能是确定在什么时候分派处理器，并确定分给哪一个进程。在分时系统中，一般有一个确定的时间单位(时间片)。当一进程用完一个时间单位时，就发生进程调度，即让正在运行的进程改变状态并转入就绪队列的队尾，再由调度原语将绪队列的首进程取出，投入运行。

进程调度的方法基本上分为两类：非剥夺调度与剥夺调度。所谓非剥夺调度是指一旦某个作业或进程占有了处理器，别的进程就不能把处理器从这个进程手中夺走；相反，如果别的进程可将处理器从这个进程手中夺走则是剥夺调度。

进程调度的算法采用服务于系统目标的策略，对于不同的系统与系统目标，常采用不同的调度算法，如先来先服务、优先数调度和轮转法等。

(5)管程

管程是一种并发性的构造，包括用于分配一个特定的共享资源或一组共享资源的数据和过程。为了完成分配资源的功能，进程必须调用特定的管程入口。许多进程可能打算在不同的时间进入管程，但在管程边界上严格地实施互斥，在某一时刻，只允许一个进程进入。当管程中已有一个进程时，其他希望进入管程的进程必须等待。这种等待是由管程自动管理的。管程中的数据或者是管程中所有的全局变量，或者是某个特定过程的局部变量。所有这些数据只能在管程内访问，在管程外的进程无法访问管程内的数据，这叫做信息掩蔽。

1.3.1.3 存储管理

现代计算机系统中的存储系统常是多级存储体系，至少有主存(内存)和辅存(外存)两级，有的系统有更多级数。主存大小由系统硬件决定，是实实在在的存储，它的存储容量受到实际存储单元的限制。虚拟存储(简称虚存)不考虑实际主存的大小和数据存取的实际地址，只考虑相互有关的数据这间的相对位置，其容量由计算机的地址的位数决定。

1.3.1.4 设备管理

设备管理是对计算机输入/输出系统的管理。其主要任务有：实现对外部设备的分配和回收；启动外部设备；控制输入/输出设备与处理器或主存间交换数据；实现对磁盘的调度；处理设备的中断；实现虚拟设备等。

外部和主存之间常用的传输控制方式有 4 种：程序控制方式、中断方式、直接存储访问(DMA)方式和通道方式。

1.3.1.5 文件管理

(1)文件系统

操作系统的文件系统包括两个方面：一方面包括负责管理文件的一组系统软件，另一方面包括被管理的对象文件。文件系统的主要目标是提高存储器的利用率，接受用户的委托，实施对文件的操作。主要问题是管理辅助存储器，实现文件从名字空间到辅存地址空间的转换，决定文件信息的存放位置、存放形式和存放权限，实现文件和目录的操作，提供文件共享能力和安全设施，提供友好的用户接口。

(2)文件的结构和组织

文件的结构是指文件的组织形式。从用户观点所看到的文件组织形式，称为文件的逻辑结构；从实现观点考查文件在辅助存储器上的存放方式，常称为文件的物理结构。

文件的逻辑组织是为了方便用户使用。一般文件的逻辑结构可以分为两种：无结构的字符流文件和有结构的记录文件，后者也称为有格式文件。优化文件的物理结构是为了提高存储器的利用效率和降低存取时间。文件的存储设备通常被划分为大小相同的物理块，物理块是分配和传输信息的基本单位。文件的物理结构是指文件在存储设备上的存储主法。文件的物理结构涉及文件存储设备的组块策略和文件分配策略，决定文件信息在存储设备上的存储位置。

1.3.1.6 作业管理和用户界面

作业(Job)是系统为完成一个用户的计算任务或一次事务处理所做的工作的总和。操作系统中用来控制作业的进入、执行和撤消的一组程序称为作业管理程序，这些控制功能也能通过把作业步细化、通过进程的执行来实现。

用户的作业可以通过直接的方式，由用户自己按照作业步顺序操作；也可以通过间接的方式，由用户事先编写作业步依次执行的说明，一次交给操作系统，由系统按照说明依次处理。前者称为联机方式，后者称为脱机方式。

一般操作系统提供两种作业控制方式，一种为联机作业方式，另一种为脱机作业方式。联机作业方式是通过直接输入作业控制命令来提交和运行用户作业。脱机作业方式是通过作业控制语言(JCL，也称为作业控制命令)编写用户作业说明书。在这种方式中，用户不直接干预作业的运行，而是把作业与作业说明书一起交给系统(称为提交)。

作业调度主要是从后备状态的作业中挑选一个(或一些)作业投入运行。根据不同的调度目标，有不同的算法。作业调度算法有许多种，它们与进程调度相似，有的适宜于单道系统，有的适宜于多道系统。它们是先来先服务(FCFS)、短作业优先(SJF)、响应比(HRN)高者优先和成先级调度等。

1.3.1.7 其他管理

(1)死锁问题

如果一个进程正在等待一个不可能发生的事件，则称该进程处于死锁状态。系统发生死锁是指一个或多个进程处于死锁状态。产生死锁的主要原因是共享的系统资源不足，资源分配策略和进程的推进顺序不当。系统资源既可能是可重用的永久性资源，也可能是消耗性的临时资源。处于死锁状态的进程不能继续运行又占有了系统资源，阻碍其他进程的运行。对待死锁的策略主要有：

①死锁的预防。不让任一产生死锁的必要条件发生就可以预防死锁。

②死锁的避免。这种策略不对用户进程的推进顺序加以限制，在进程申请资源时先判断这次分配安全否，只有安全实施分配，典型的算法是银行家算法。

③死锁的检测。这种策略采用资源请求分配图的化简方法来判断是否发生了不安全状态。资源请求分配图是一种有向图，表示进程与资源之间的关系。死锁的检测是在需要的时刻执行的，当发现系统处于不安全状态时，即执行死锁的解除策略。

④死锁的解除。解除死锁的基本方法是剥夺。一种方法是把资源从一些进程处剥夺分给别的进程，被剥夺资源的进程则需回退到请求资源处重新等待执行；另一种主法是终止一个进程，剥夺其全部资源，以后再重新运行被终止的进程。

(2)多重处理器系统与线程

多重处理系统的主要目标是为了提高系统的处理能力，也是为了提高系统的可靠性。多重处理系统的操作系统除了具有单处理器操作系统的功能以外，还应提供处理器的负载平衡、处理器发生故障后的结构重组等功能。一般多重处理系统的操作系统可以分为主从式、分离执行式和移动执行式 3 类。

对称多处理器系统 SMP 是由若干同构甚至相同的处理器构成的一个系统。Solaris 和 Windows NT 等操作系统支持 SMP 系统。操作系统提供了线程(Thread)机制以发挥多个处理器的作用。在多线程系统中，一个进程可以由一个或多个线程构成。进程是资源分配的基本单位，也是被保护的基本单位。一个进程对应于一个保存进程映象的虚地址空间，每一线程可以独立运行一个进程的线程共享这个进程的地址空间。有多种方法可以实现多线程系统，一种方法是核心级线程，另一种方法是用户级线程，也可以把两者组合起来。

1.3.1.8 操作系统的结构

(1)无序结构法，又称整体结构或模块组合结构。它以大型表格和队列为中心，操作系统的各部分程序围绕着表格运行，整个系统是一个程序。这种操作系统常称为面向过程的操作系统。

(2)层次结构法是把一个大型复杂的操作系统分解成若干个单向依赖的层次，由多层的正确性保证操作系统的可靠性。层次结构清晰，且有利于系统功能的增加或删改。

(3)面向对象的操作系统基于面向对象程序设计的概念，采用了各种不同的对象技术。在计算机系统中对象是操作系统管理的信息和资源的抽象，是一种抽象的数据类型。可以把对象作为系统中的最小单位，由对象、对象操作、对象保护组成的操作系统，就是面向对象的操作系统。如 Windows NT 中有执行体对象(进程、线程、文件和令牌等)和内核对象(时钟、事件和信号等)。

(4)微内核结构法把系统的公共部分抽象出来，形成一个底层核心，提供最基本的服务，其他功能以服务器形式建立在微内核之上。它具有良好的模块化和结构化特征，模块之间和上下层之间通过消息来通信。建立在微内核上的服务器可以根据不同的需要构造，从而形成不同的操作系统，如 Windows NT 操作系统。

2.3.1.9 常用操作系统

UNIX 系统是一个分时操作系统。它利用最内层硬件提供的基本服务，向外层提供全部应用程序所需要的服务。应用程序组可以构成应用子系统，如 UNIX 系统的源代码控制系统(SCCS)、图形(X-Window、Motif)等。

Windows NT 系统是 20 世纪 90 年代的操作系统技术，适用于高档工作站平台、局域网服务器或者主干计算机。Windows NT 支持对称处理器结构，支持多线程并行，采用 90 年代操作系统技术(即微内核技术)，在体系结构上采用客户机/服务器模式。

1.3.2 试题解析

操作系统也是每年必考的知识点。从历年试题统计(见表 2-4)来看，考查内容主要集中在存储管理、进程、作业管理等知识点，特别是有关进程的内容反复考查，考查的问题也都差不多。操作系统基础试题还有一个突出的特点，自 1998 年以来，每年都有一道试题与过去考查过的试题基本相同，其中 1998 年试题 7 与 1990 年试题 2 基本相同，1999 年试题 4 与 1995 年试题 1 基本相同，2000 年试题 2 与 1998 年试题 9 几乎完全一样，1991 年试题 6 则原封不动地搬以了 2000 年程序员级上午试卷上。由此可见，复习历年试题是非常重要的。

表 2-4　历年高级程序员级操作系统基础知识试题统计

试题	考查知识点
1990 年试题 2	虚拟存储管理系统
1991 年试题 2	操作系统常识，涉及重定向和批处理（2000 年程序员级上午试题 4 重考）
1991 年试题 6	存储管理
1992 年试题 4	进程
1993 年试题 3	进程间通信，涉及信号量与 P、V 操作
1993 年试题 15	管程
1994 年试题 4	进程的概念和性质
1994 年试题 16	文法
1995 年试题 1	进程
1996 年试题 6	作业管理
1997 年试题 2	作业管理
1998 年试题 4	涉及进程间通信，主要考查信号量与 P、V 操作
1998 年试题 7	虚拟存储管理（基本同 1990 年试题 2）
1998 年试题 9	段页式存储管理
1999 年试题 4	进程（基本同 1995 年试题 1）
2000 年试题 2	段页式存储管理（同 1998 年试题 9）
2000 年试题 4	进程与线程

试题 1 (2000 年试题 2)

从供选择的答案中，选出应填入下面叙述中{ }内的最确切的解答，把相应编号写在答卷的对应栏内。

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

设逻辑地址长度分配如下，其中 x、s、p、d 均以二进制数表示。

21　22　19　14　13　11　10　0

x　s　p　d

其转换后的地址为 E。

供选择的答案

A：①段 ②页 ③区域 ④块

B：①区域 ②页 ③块 ④段

C：①模块号 ②区域号 ③基号 ④区域

D：①一个段表和一个页表 ②一个段表和一组页表

③一组段表和一个页表 ④一组段表和一组页表

E：①$x*2^{20}+s*2^{14}+p*2^{11}+d$ ②(((x)+s)+p)+d

③(((x)+s)+p)*2^{11}+(d) ④(((x)+s)+p)*2^{11}+d

【解析】

本题涉及存储管理知识，主要考查段页式存储管理，与 1998 年试题 9 几乎完全相同。

段页式存储组织综合了段式组织与页式组织的特点，主存被划分为定长的页，段页式系统中的虚地址形式是(段号、页号、位移)。系统为每个进程建立一个段表，为每个段建立一个页表。也就是说，先将程序按逻辑模块(如主程序、子程序和数据段等)分为若干段，再将每个段分为若干页。

对于多道程序环境，每道程序有一个基号与其他程序相区分，每道程序可以有多个段，但只有一个段表，每个程序可以有多个页表。

段页式存储体系中逻辑地址与物理地址的转换：首先由基号段号得到段表的地址，再访问段表得到页表的地址，再由页表得到物理块的地址，此时得到的地址是高 11 位的地址，因此需乘以 2^{11} 再加上页内地址，才得到真正的物理地址。

【答案】A：② B：④ C：③ D：② E：④

试题 2（2000 年试题 4）

从供选择的答案中，选出应填入下面叙述中{ }内的最确切的解答，把相应编号写在答卷的对应栏内。

A 是操作系统中可以并行工作的基本单位，也是核心调度及资源分配的最小单位，它由 B 组成，它与程序的重要区别之一是 C。

在 SMP 系统中，操作系统还提供了 D 机制，它是 E 的最小单位。

供选择的答案

A：①作业 ②过程 ③函数 ④进程

B：①程序、数据和标识符 ②程序、和 PCBD

③程序、标识符和 PCB ④数据、标识符和 PCB

C：①程序可占有资源，而它不可 ②程序有状态，而它没有

③它有状态，而程序没有 ④它能占有资源，而程序不能

D：约束 ②线程 共享 分时

E：①存储器分配 ②资源分配 ③处理器分配 ④网络结点分配

【解析】

本题考查关于进程与线程方面的知识。

进程是一个程序关于某个数据集的一次运行。也就是说，进程是运行中的程序，是程序的一次运行活动。相对于程序，进程是一个动态的概念，而程序是静态的概念，是指令的集合，因而进程具有动态性和并发性。

进程是一个动态的概念，在操作系统中，引入数据结构-PCB(进程控制块)来标记进程，。PCB 是进程存在的唯一标志，PCB 描述了进程的基本情况。从静态的观点看，进程由程序、数据和进程控制块组成。

对称多处理器系统 SMP 是由若干同构甚至相同的处理器构成一个系统，其操作系统提供了线程(Thread)机制以发挥多个处理器的作用在多线程系统中，一个进程可以由一个或多个线程构成。进程是资源分配的基本单位，而线程是处理器分配的最小单位。

【答案】：A：④ B：② C：③ D：② E：③

试题 3（1999 年试题 4）

从供选择的答案中，选出应填入下面叙述中{ }内的最解切的解答，把相应编号写在答卷的对应栏内。

进程是操作系统中的一个重要概念。进程是一个具有一定独立功能的程序在某个数据集合上的一次 A。

进程是一个 B 的概念，而程序是一个 C 的概念。

进程的最基本状态有 D。在一个单处理机中，若有 6 个用户进程，在非管态的某一时刻，处于就绪状态的用户进程最多有 E 个。

供选择的答案

A：①单独操作 ②关联操作 ③运行活动 ④并发活动

B：①静态 ②动态 ③逻辑 ④物理

C：①物理 ②逻辑 ③动态 ④静态

D：①就绪、运行、隐蔽 ②停止、就绪、运行

③运行、就绪、阻塞 ④就绪、撤消、运行

E：①5 ②6 ③1 ④4

【解析】

本题考查操作系统中的进程知识，与 1995 年试题 1 基本相同。

进程是操作系统中基本的并行单位、资源分配单位和调度单位。通常，进程可分为用户进程和系统进程两类，前者控制用户作业的运行，后者完成系统内部分工的管理工作。

从静态的角度看，进程由程序、数据和进程控制块(JCB)组成；从动态的角度看，进程是计算机状态的一个有序集合。进程是一个具有一定独立功能的程序在某个数据集合上的一次运行，其中可能要涉及多个程序；而一个程序的运行过程中可能有若干进程依次或并行活动。进程既是一个运行单位，又是一个调度单位和资源使用单位。进程和程序是有区别的。进程是有状态的，而程序没有状态。进程是一个动态的概念，它可以执行、暂停、继续运行，而程序是一个静态的概念，它体现了某个算法，而且多个进程可以共享一个程序。

进程的基本状态有就绪、运行和阻塞 3 种。阻塞态是指一个进程由于某种原因不具备运行条件时所处的状态，这时它必须等待，引起等待的条件一旦清失，进程便具备了运行的条件，状态转变为就绪态；就绪态是指一个进程具备了运行的条件，但由于没有占有处理机而不能运行所处的状态，一旦处于就绪态的进程轮到该进程占有处理的时间片或处理机空闲，其状态就转变为运行态，投入运行；运行态是指一个进程正占用着处理机时的状态，这时，处理机正在执行该进程的程序，运行过程中进程会因时间片已到等非资源请求原因退出运行转变为就绪态，因资源请求原因而不具备运行条件时，该进程的状态就要转变为阻塞态。

由于只有占有处理机进程才能处于运行态，所以在一个单处理机中，非管态的某一时刻，系统中处于就绪态或阻塞的进程可能有多个，但处于运行态的进程最多只有一个，也有可能是 0 个（系统死锁），这样处于就绪态的进程数只能是进程总数减 1，不可能出现全部处于就绪态而无运行态进程的情况。

【答案】A：③ B：② C：④ D：③ E：①

试题 4（1998 年试题 5）

从供选择的答案中，选出应填入下面叙述中{ }内的最确切的解答，把相应编号写在答卷的对应栏内。

信号量是操作系统中用作互斥和同步机制的一个共享的整数变量。信号量仅可以由初始化、唤醒（Signal）和等待（Wait）3 种操作访问。

对于给定的信号量 S，等待操作 Wait(S)（又称 P 操作）定义为 IF S 大于 O THEN A ELSE 挂起调用的进程:唤醒操作 Signal(S)(又称 V 操作)定义为 IF 为\存在等待的进程 THEN 唤醒这个进程 ELSE B 。

给定信号量 S，要以定义一个临界区来确保其互斥，即保证在同一时刻这个临界区只能够被一个进程执行。当 S 被初始化为 1 时，代码段定义了一个临界区。

C：

（临界区）

D

这样的临界区实际上是将共享数据和对这些数据的操作一起封装起来，通过其互斥机制，一次只允许一个进程进入，这种临界区能常称为 E 。

供选择的答案

【解析】

信号量是操作系统中用作互斥和同步机制的一个共享整数变量，除初始化外，仅能通过两个标准的原子操作（Atomic Operation）-Wait(s)和 Signal(S) 来访问。对于给定的信号量 S，等待操作为：若 S＞则 S：=S-1，禁止其他进程访问此临界资源，否则挂起调用的进程。唤醒操作为：若存在等待的进程，则唤醒它，否则 S：=S+1，允许其他进程访问此临界资源。

每个进程中访问昨界资源的那段代码称为临界区（Critical Section ）。显然，若能保证每个进程互斥地进入自己的临界区，就能实现它们对临界资源的互斥访问。这样，每个进程在进入临界区，。访问该资源，并设置信号量，表示资源正在被访问，否则应等待（挂起）， 这个操作即 Wait(S)。当其访问完临界资源，退出临界区时，检查若有进程被挂起（即在等待访问比临界资源），则唤醒该进程，否则应当恢复信号量，以使其他地程将来能访问此临界资源，这个操作即 Signal(S)。

引入信号量机制的目的是为了消除与时间有关的错误，但如果在使用 Wait(S)和 Sigal（S）操作时出现错误，同样也会出现与时间有关的错误。为了解决这种问题，Dijkstra 在 1971 年提出，把所有进程对某一种临界资源的同步操作都集中起来，构成一个所谓的"秘书"进程，凡是要访问此临界资源的进程都要先报告"秘书"，由"秘书"来实现者进程的同步。后来，"秘书"进程思想又进一步发展为管程的概念，即"一人管理定义了一个数据结构和能为并发进程所执行（在此数据结构上）的一组操作，这组操作能同步改变管程中的数据"。

【答案】A：③ B：② C：⑧ D：⑦ E：③

试题 5 （1998 年试题 7）

从供选择的答案中，选出应填入下面叙述中{ }内的最确切的解答，把相应编号写在答卷的对应栏内。

虑拟存储管理系统的基础是程序的 A 理论。这个理论的基本含义是提程序执行时往往会 B 访问内存储。程序的 A 表现在 C 和 D 上。C 是指最近被访问的存储单元可能马上又要被访问。 D 是指马上被访问的单元，而其附近的单元也可能马上被访问。

根据这个理论，Denning 提出了工作集理论。工作集是进程运行时被频繁也访问的页面集合。在进程运行时，如果它的工作页面都在 E 器内，能够使用进程有效地运行，否则会出现频繁的页面调入/调出现象。

供选择的答案

A： ①局部性 ②全局性 ③动态性 ④虚拟性

B： ①频繁地 ②均匀地 ③不均匀地 ④全面地

C、D： ①数据局部性 ②空间局部性 ③时间局部性 ④数据全局性

E： ①外部存储 ② 主存储 ③辅助存储 ④虚拟存储

【解析】

本题考查虚拟存储管理系统知识，与 1990 年试题 2 基本相同。

虚拟存储管理系统的基础是程序的局部性原理。所谓程序局部性原理是指程序在执行时所呈现的局部性规律，即在一段较短时间内，程序的执行仅限于某个部分。相应地，它所访问的存储器空间也局限在某个空间。局部性原理又是表现为两个方面：

（1）时间局限性

如果某条指令被执行，则不久以后该指令很可能再次被执行；如果某条数据结构被访问，则

不久以后该数据结构很可能再次被访问。产生时间局限性的主要原因是程序中有大量的循环操作。

（2）空间局限性

一旦程序访问了某个内存单元，不久以后，其附近的内存单元也要被访问，即程序在一段时间内所访问的存储器空间可能集中在一定的范围之内，其最常见情况就是程序的顺序执行。工作集是指在某段时间隔内，进程所要访问的页面集合。虽然程序只需少量的几页内存就可以运行，但为了使程序更有效地运行，必须使程序的工作集全部在内存（主存储器）当中，否则会使进程在运行中频繁出现缺页中断，从而出现频繁的页面调入/调出现象。

【答案】A：① B： C：② D：③ E：④

试题 6 （1998 年试题 9）

从供选择的答案中，选出填入下面叙述中{ }内的最确切的解答，把相应编号写在答卷的对应栏内。

在段页式管理的存储器中，实存等分为 A ，程序按逻辑模块分成 B 。在多道程序环境下，每道程序还需要一个 C 作为用户标志号。每道程序都有对应的 D 。一个逻辑地址包括 CX、段号 S 页号 P 和页内地址 d 这 4 个部分。

假设总长度为 22 位的逻辑地址格式分配为：21～20 位 X，19～14 位 s,13～位 P,10～0 位 d。若 x、s、p 和 d 均以二进制数表示，其转换成的物理地址为 E 。

供选择的答案

A～C ①段 ②页 ③基 ④模块

⑤区域 ⑥段号 ⑦页号 ⑧基号

⑨模块号 ⑩区域号

D： ①一个段表和一个页表 ②一个段表和一组页表

③一组段表和一个页表 ④一组段表和一组页表

E： ①$xx2^{20}+sx2^{14}+px2^{11}+d$ ②（(x) +s+p）$x2^{11}+d$

③((x)+s+p)+d ④$((x)+s)+p)x2^{11}+d$

（注：式中（Y）表示地址为 Y 的单元的内容。）

【解析】

请参见本节试题 1（2000 年试题 2）的解答。

【答案】A：①B：② C：③ D： E：④

试题 7 （1997 年试题 2）

从供选择的答案中，选出应填入下面叙述中{ } 内的最确切的解答，把相应编号写在答卷的对应栏内。

在有一台处理机 CPU 和两台输入车输出设备 IOI 和 IO2，且能够实现抢先式多任务并行工作的多道程序环境内，投入运行优先级由高到低的 P1、P2 和 P3P 这 3 个作业。它们使用设备的先后顺序和占用设备时间分别是：

作业 P1：IO2（30ms）、CPU（10ms）、IOI(30ms)、CPU(10ms)

作业 P2：IOI (20ms)、CPU（20ms）、IO2(40ms)

作业 P3：CPU (30ms)、IOI (20ms)

假设对于其它辅助操作时间可以忽略不计，作业 P1、P2、P3 从投入到完成所用的时间分别是 A ms、 B ms、 C ms。3 个作业从投入运行到全部完成，CPU 的利用率约为 D %，IO1 的利用率约为 E %

假设在系统中仅有这 3 个作业投入运行，各设备的利用率是指该设备的使用时间同作业进程

组全部完成所占用量长时间的比率。

【解析】

处理机和输入输出设备采取抢先式多任务并行工作方法，分析完成每个作业所需的时间时，可先分析优先高级的作业。高优先级的作业有优先获取资源的优势，在只有 3 个作业时，最高优先级的作业 P1 发出的申请设备请求会立即得到响应。各设备占用时间为：

在 P1 优先占用设备的基础上，P1、P2 和 P3 可以在剩下的作业中优先得到资源。

从表 2-5 可以看出 P2 在使用 IOI 设备 20ms 后，要使用 CPU20ms，但当其运行 10 ms 后，由于 P1 优先高级并采用抢先方式，所以将 P2 暂停，让 P1 先运行。同理，P3-开始就使用 CPU，但在运行 20ms 后，要让给 P2 和 P1。

P1 投入运行到完成需要 80ms，P2、P3 由行等待资源，运行时间都延长为 90ms。CPU 在 60ms～70ms 和 80ms～90ms 的共 20ms 时间内没有利用，所以利用率为 70/90≈ 78%，IO2 的利用率也为 78%。

表 2-5　　1997 年试题 2

IO1	P_2	P_2			P_1	P_1	P_1	P_1	P_1
IO2	P_1	P_1	P_1			P_2	P_2	P_2	P_2
CPU	P_1	P_1	P_2	P_1	P_2	P_1		P_1	

试题 8 （1996 年试题 6）

从供选择的答案中，选出应填入下面叙述中{ }内的最确切的解答，把相应编号写在答卷的对应栏内。

由于内存大小有限，为使得一个或多个作业能在系统中运行，常需要用外存来换取内存。其中作业为单位对外存进行交换的技术称为 A 技术，在作业内部对内外存进行交换的技术称为 B 技术。

用我存换内存是以牺牲程序运行时间为代价的。为提高 CPU 有效利用率，避免内外存的频繁交换， B 技术常用某种 C 来选择换出内存的页面，它的基础是程序的 D 。据此，操作系统可根据 E 来改善系统的性能。 是一个进程在定长的执行时间内涉及到的页面集合。

供选择的答案

A、B ①SPOOLING ②SWAPPNG ③虑拟存储 ④虑拟机

⑤进程管理 ⑥设备管理

C： ①页面分本策略 ②页面标志策略 ③页面淘汰策略 ④虑拟机

D： ①完整性 ②局部性 ③递归性 ④正确性

E： ①工作集 ②页面集 ③段号集 ④元素集

【解析】

SWAPPING 技术是对换术的一种，用于解决由于内存不足而无法运行多道程序的问题，它以整个进程（作业）为单位进行对换，被对换到外存的作业将在一段时间内停止运行，这种对换广泛应用于分时系统中。所以问题 A 选择②。

对于一道作业来说，受内存容量的限制，作业进程能得到的内存空间会小于作业所需的空间。为使作业能在系统中运行，需要采用作业内部的对换技术，在作业内部以“页”或“段”为单位进行部分对换，即在内存中保留作业进程的一部分，而在外存放置作业进程的副本。作业运行时可以随机地访问在内存的那部分程序和数据，一旦访问不在内存的程序或数据时，就腾出部分暂时不用的内存区域，将它们的内容调出至外存，再将所要访问的那些内容调入内存。实现作业内部的部分对换，从而允许作业的地址空间大于实际分配的内存区域，这就是虚拟存储技术。所以问题 B 选择③。

虑拟存储技术中一个重要问题是如何选择页面置换算法，即如何淘汰页面，以避免内外存的频繁交换，降低系统性能，所以问题 C 选择③。
虑拟存储管理系统的基础是程序的局部性原理。所谓程序局部性原理是指程序在执行时所呈现的局部性规律，即在一段较短的时间内，程序的执行仅限于某个部分。相应地，它所访问的存储器空间也局限于某个空间。
程序局部性有两个方面的含义：时间局部性，即如果一条指令被执行，则在不久以后可能再次执行，产生时间局限的主要原因是程序中有大量的循环操作：空间局部性，即一旦程序访问了某个内存单元，不久以后，其附近的内存单元也要被访问。程序在一段时间内所访问的存储器空间可能集中在一定的范围之内，其最常见的情况就是程序执行。所以问题 D 选择②。
操作系统可以利用工作集模型来改善系统的性能，工作集是指在某段时间间隔内，进程所要访问的页面集合。虽然程序只需少量的几页内存就可以运行，但为了使程序有效地运行，必须使程序的工作集全部在内存（主存储器）当中，否则会使进程在运行中频繁出现缺页中断，从而出现频繁的页面调入/调出现象。所以问题 E 选择 ①
【答案】A：② B：② C：③ D：④ E：①

试题 9（1995 年试题 1）

从供选择的答案中,选出应入下面叙述中{ }内的正确答案,把编号写在答卷的对应栏内.
在操作系统中,进程是一个具有一定独立功能的程序,在某个数据集合上的一次 A ，进程是一个 的 B 的概念，而程序是一个 C 的概念。
在一单处理机中，若有 5 个用户进程，在非管态的某一时刻，处于就绪状态的用户进程最多有 D 个， 最少有 E 个。
供选择的答案：
A：①并发活动 ②运行活动 ③单独操作 ④关联操作
B、C：①组合态 ②并联态 ③运行态 ④等待态
⑤静态 ⑥动态
D、E：①1 ②2 ③3 ④4
【解析】
本题考查进程知识，请参见本节的试题 3（1999 年试题 4）的解答。
在非管态的某一时刻，处理机将处理用户进程，显然应该而且必须有一个进程处于运行态，那么处于就绪态的进程最多只能有 4 个，最少有 0 人，即除了处于运行态的进程外，其余 4 个进程都不具备运行的条件而处于待态，使处于就绪态的进程只有 0 个。

试题 10 （1994 年试题 4）

从下面有关进程的概念和性质量叙述中，选出 5 条正确的叙述，并把编号按从小到大的次序写在答卷的 A～E 栏内。
①唤醒：挂起→就绪。
②封锁：就绪→挂起。
③调度：就绪→运行。
④超时：运行→挂起。
⑤超时：运行→就绪。
⑥用户进程可激发调度进程。
⑦用户进程可激发唤醒进程。
⑧用户进程可激发超时进程。

⑨进程接近 CPU 可接纳的限度时，可降低页面出错的频率。
⑩进程具有引用局部性时，可降低页面出错的频率。
【解析】
操作系统中的处理器管理主要是把 CPU 的处理时间合理地分配给进程，充分利用处理器的运算能力。进程就是系统进行分配和调度的最小单位，有等待、就绪和运行 3 种基本状态。等待状态时不具备运行条件，等待自身状态或系统状态变化；就绪状态时进程已分到资源。进程的状态是可变化的，处于等待状态的进程只有在所等待的时间到来或条件满足时，才能进入就绪状态，这就是唤醒，因此①和⑦是正确的。就绪状态的进程被操作系统调度并选中便进入就绪状态，③是正确的，运行状态下，CPU 时间片用完，这就是超时，此时进程转入就绪状态， ⑤是正确的，④是错误的。运行状态只有在进程受阻的情况下才会进入等待状态，就绪状态不会因陋就简封锁进入挂起状态，②是错误的。从进程基本状态的转换来看，只有唤醒是受时间激发的，其他都是由操作系统进行内部管理，因此用户不能激发调度与超时，⑥和⑧是错误的。操作系统对存储的管理主要是对内存的分配、保护和扩充。内存分配管理办法之一就是分页式管理。当进程具有引用局性时可降低页面出错频率，可见⑩正确，⑨错误。
[答案]A：① B：③ C：⑤ D：⑦ E：⑩

试题 11 （1993 年试题 3）

从供选择的答案中，选出应填入下面关于操作系统叙述中{ }内的正确答案，把编号写在答卷的对应栏内。
在操作系统中，解决进程间的 A 两种基本关系，往往运用对信号量进行 B 的 C 。例如，为保证系统数据库的完整性，可以把信号量定义为某个库文件（或记录）的锁，初值为 1，任何进程存取该库文件（或记录）之前先对它作一个 D ，存取之后对它作一个 E,，从而做到对该文件（或记录）任一时刻只有一个进程可存取，但要注意使用不当引起的死锁。
供选择的答案
A： ① 同步与异步 ②串行与并行 ③调度与控制 ④同步与互斥
B： ① 消息操作 ②P-V 操作 ③开关操作 ④读写操作
C： ① 能信原语 ②调度算法 ③分配策略 ④进程控制
D、E ① 联机操作 ②操作 ③输出操作 ④读操作⑤写操作 ⑥P 操作 ⑦输入操作
【解析】
在操作系统中，进程是可以独立运行的单位，由于进程同处于一个系统中，它们之间不可避免地会有某种联系，同步和互斥是进程间的两种基本关系。同步是指协调关系，即为了完成同一任务，两个进程应有某种约定，彼此协调运行速度：互斥是指竞争关系，即排它性。当一个进程使用某一资源时，另一进程或者等待，或者根据某种优先级算法确定该资源归谁使用。 为了解决进程间的同步和互斥关系，可引用信号量的概念，信号量是一种特殊的变量，它的表现形式是一个整型普量附加一个队列，它只能被 P 操作和 V 操作使用。P 操作操作只能顺序进行，也就是必须在一个操结束以后才开始第二个操作，利用 P、V 操作可以容易地实现进程之间的互斥。
一般 P 操作与 V 操作的定义如下所述：
P 操作：(1) S-1→S;(2)如果 S＜0，则该进程进入等待状态；否则继续进行；
V 操作：(1) S+1→S；(2) 如果 S≥0，则唤醒队列中的一个等待进程。
其实,P 操作就是给进程分配某个特实资源,而 V 操作则是进程在用完该资源后交还给系统。S 信号量为负数时也就表示等待使用资源的进程的个数。这样,。可以保证在任保一时刻对数据库文件只有一个进程可以存取。值得注意的是，同一信号量上的 P、V 操作往往是成对

出现的。P、操作使用不当会引起死锁。

【解答】

A：④ B：② C：① D：⑥ E ②

试题 12 （1993 年试题 15）

从供选择的答案中，选出应填入下面关于操作系统叙述中 内的正确答案，把编号写在答卷的对应栏内。

操作系统中，管程（Monitor）由管程名、局部于管程的变量说明、使用共享资源并在数据结要上进行操作的若干过程以及对变量赋初值的语句等 4 个基本部分组成。每一个管程管理 A 。任何一个进程请求使用某临界资源时，对于其他相关进程而言 B 。其方法是通过调用特定的 C 才能进入管程，然后能过管程中 D 使用临界资源。在执行中发现共享临界资源被告或条件不成立时，调用管程的进程必须等待使用临界资源的另一进程来唤醒。为了表示不同的等待原因，设置了条件变量，条件变量是 E 。

供选择的答案

A： ①一个临界资源 ②一类临界资源
③若干个临界资源 ④所有的临界资源

B： ①可以先后进入管程 ②必须互斥地进入管程
③可以同时进入管程 ④可以按任意次序进入管程

C： ①管程入口 ②访管指令 ③系统调用命令

D： ①使用临界资源的进程 ②访问管程的进程
③使用临界资源的一个过程 ④调用管程的进程

E： ①与普通变量相同的变量
②与普通变量不同的变量，条件变量不能取任何，只是一个排队栈
③与普能变量不同的变量，只能取真、假值，不能取 0，1…，n 值

【解析】

在实现一个管程时必须考虑互斥、同步和条件变量这 3 个问题。管程中对每一个条件变量都予以说明，条件变量与普通变量不同，它不能取任何值，只能是一个排队栈。

当有几个进程调用某管程时，仅允许一个进程进入管程，其他调用者必须等待，也就是申请进程必须互斥地进入管程。方法是可通过调用特定的管程入口进入管程，再通过管程中的一个过程来调用临界资源。当某进程通过调用请求访问某临界资源而未能满足时，管程调用相应同步原语使该进程等待，并将它排在等待队列上。当使用临界资源的进程访问完该临界资源并释放之后，管程又调用相应的同步原语唤醒等待队列中的队首进程。

【答案】A：① B： ② C：① D ：③ E：②

试题 14 （1992 年试题 4）

从供选择的答案中造出应填入{ }的正确答案，把编号写在答卷的对应栏内。

从静态角度看，进程是由 A 、B 和 C 3 个部分组成。用户可通过 D 建立和撤消进程。通常，用户进程被建立后，E 。

供选择的答案

【解析】

A：①JCB ②DCB ③PCB ④PMT

B：①程序段 ②文件体 ③I/O ④子程序

C：①文件描述块 ②数据空间 ③系统调用 ④过程调用

D：①函数调用 ②宏指令 ③系统调用 ④过程调用

E：①便一直存在于系统中，直到被操作人员撤消
②随着作业运行正常或不正常结束而撤消
③随着时间片轮转而撤消与建立
④随着进程的阻塞或唤醒而撤消与建立
【解析】
进程是操作系统中可以并发运行的最基本单位。进程实体是由进程控制块（PCB）、程序段和数据空间组成。从进程的运行来看，进程可并发运行的程序在其数据集合上的运行过程。操作系统一般向用户提供命令级和系统调用级的服务。包括设备控制、文件处理，建立进程和撤消进程以及其他的实用进程等。能常，用户进程被建立后，随着作业运行的正常或非正常结束而撤消。
【答案】A：③ B：① C：② D：③ E：②

试题 15 （1991 年试题 2）

从供选择的答案中选出填入下列叙述中的{ }内的正确答案，把编号写在答卷的对应栏内。
某些操作系统把一条命令的执行结果输出给下一条命令，作为它的输入，并加以处理，这种机制称为 A ，使命令执行的结果不在屏幕上显示，而引向另一个文件，这种机制称为 B ，使命令所需要的处理信息不从键盘接收，而调用一个正文文件。这种机制称为 C 。操作系纺不从键盘逐条接收命令并执行，而调用一个正文文件，执行其中保存的一系列命令，这种方式属于 D 方式，编写这样的文件应符合 E 的语法规则。
供选择的答案
A：①链接 ②管道（线） ③输入重新定向 ④输出重新定向
B：①清屏 ②屏蔽显示 ③输出重新定向 ④管道（线）
C：①输入重新定向 ②读保护 ③管道（线） ④批处理
D：①系统生成 ②初始装入 ③批处理 ④管道（线）
【解析】
一些操作系统（例如 UNIX）采用管道（线）的概念实现两个进程之间信息流的传递。管道是能够连接一个读进程和一个写进程，使二者进行通信的共享文件。通过把一个进程产生的信息流同管道传递给另一个进程，可以实现把一条命令的执行结果输出给下一条命令处理。某些操作系统。具有让命令的执行结果不显示在屏幕上而引向另一个文件的机制，称为输出重新定向；让命令所需的处理信息不从键盘接收，而取自另一个文件的机制，称为输入重新定向。
用户一般通过键盘键入命令，启用操作系统。有时为了提高使用效率，按命令语言的语法规定一系列命令保存在一个正文文件内，让操作系统调用这个批处理文件，成批地执行其中的命令，就是所谓的批处理方式。
【答案】A：② B：③ C： ① D：③ E：④

试题 16 （1991 试题 6）

从供选择的答案中选写出下列叙述关系最密切的存储管理方法，把编号写在答卷的对应栏内。
A.支持多道程序设计，算法简间，但存储器碎片多。
B.能消除碎片多和紧缩处理时间长的缺点，支持多道程序设计，但不支持虑拟存储。
C.克服了碎片多紧缩处理时间长的缺点，支持多道程序设计，但不支持虚拟存储。
D.支持虚拟存储，但不能以自然的方式提供存储器的共享和存取保护机制。
E.允许动态连接和装入，能消除碎片，支持虑拟存储。

供选择答案
A～E ①段页式 ②非请求分页式 ③请求分页式
④可重定位分区 ⑤固定分区 ⑥单一连续分配
【解析】
存储管理是操作系统的一个重要组成部分，包括决定存储分配的策略、分配和回收各存储区域、登记各存储区域的状态表等功能。对于虚拟存储管理还应包括页面的分配、管理和调度等功能。
存储分配方法有如下几种：
（1）固定分区存储管理。这是支持多道程序设计的最简单存储管理方法，它把主存划分成若干个固定的和大小不同的分区，每个分区能够装入一个作业，分区的大小是固定的，算法简单，但是容易生产较多的存储器碎片。
（2）可重定位分区存储管理。这是克服固定分区碎片问题的一种存储分配方法，它能够把相邻的空闲存储空间合并成一个完整的空区，还能够整理存储器内各个作业的存储位置，以在到消除存储碎片和紧缩存储空间的目的。紧缩工作需要花费大量的时间和系统资源。
（3）存储管理。这是支持多道程序设计的存储管理方法，它将存储空间和作业的地址空间分成若干个等分部分，并且进行编号和建立映射关系。分页存储管理分为请求分页式在轮制进程被执行两种，克服了分区存储管理中碎片多和紧缩处理时间长的缺点。非请求分页式在分页式支持虚拟存储，当进程需要用到某个页面时将该页面调入主存，把那些暂时无关的页面留在主存外。请求分页的实现方法比非请求分页的实现方法复杂，它是以页为单位进行存储保护和共享，而不能以自然方式提供存储器的共享和存储保护机制。
（4）连续分区把所有用户区都分配给唯一的用户作业，当作业被调度到时，进程全部进入内存，一旦完成，所有主存恢复空闲，因此，它不支持多道程序设计。
（5）段页式存储管理。这是分段式和分页式结合的存储管理方法。段可以定义为一组逻辑信息，例如子程序、数组或数据区。每个作业的地址空间由一些分段构成，每段有名字，且占有连续的地址空间，段与段之间在物理位置上可以连续。分段管理可以通过的移动来消除碎片，同时支持虚拟存储，实现动态连接和装配。分段法和分页法的结合目的在于可以使用分段法来分配和管理虚拟存储，而使用分页法来分配和管理主存储。把每个分段分成若干页面，借助物理方法管理这些页面，而不理管理一个分段，这样可充分利用分段管理和分页管理的优点。
【答案】A：⑤ B：④ C：② D：③ E：①

试题 17 （1990 年试题 2）

从供选择的答案中选出应填入下列叙述中的{ }内的正确答案，把编号写在答卷的对应栏内。
虚拟存储管理系统的基础是程序的局部性理论。此理论的基要含义是 A 。局部性有两种表现形式：时间局部性和 B 。它们意义分别为 C 和 D 。根据局部性理论，Denning 提出了 E。
供选择的答案
A、B：①程序执行时对主存的访问是不均匀的 ②代码的顺序执行
③变量的连续访问 ④指令局部性
⑤数据局部性 ⑥空间局部性
C、D： ①最新被访问的单元，很可能在不久的将来还要被访问。
②最新被访问的单元，很可能它附近的单元也即将访问。
③结构化程序设计，很少出现转移语句。

④程序中循环语句的执行时间一般很长。
⑤程序中使用的数据局部于各子程序。
E： ①Cache 结构的思想 ②工作集理论
③最近最少使用（LPU）页面置换算法
④先进先出（FIFO）页面置换算法
【解析】本题考查虚拟存储管理系统知识，请参见本节的 试题 5 的答案。
[答案]A：① B：⑥ C：① D：② E：②

1.4 软件工程基础知识

1.4.1 主要知识点

软件工程是计算机软件的一个重要分支，主要应掌握软件工程的基本原理以及软件设计与测试方法。

1.4.1.1 软件生存周期各阶段的任务

软件生存周期指出由软件定义、软件开发和软件维护等阶段组成的全过程。

（1）软件定义阶段

软件定义阶段主要解决软件要"做什么"的问题，也就是要确定软件的处理对象、软件与外界的接口、软件的功能、软件的性能、软件的界面以及有关的约束和限制。软件定义阶段通常可分成系统分析、软件项目计划和需求分析等阶段。系统分阶析的任务是确定待开发软件的总体要求和适用范围，以及与之有关的硬件和支撑软件的要求，该阶段所生产的文档可合并在软件项目计划阶段的文档（项目计划书）中。软件项目计划的任务是确定待开发软件的目标，对其进行可行性分析，并对资源分配、进度安排等做出合理的计划，该阶段所产生的文档有可行性分析报告和项目计划书。需求分析的任务是确定待开发软件的功能、性能、数据和界面等要求，从而确定系统的逻辑模型。该阶段产生的文档是需求规格说明书。

（2）软件开发阶段

软件开发阶段主要解决软件"怎么做"的问题，包括数据结构和软件结构的设计、算法设计、编写程序和测试，最后得到可交付使用的软件。软件开发阶段通常可分成软件设计、编码、软件测试等阶段。软件设计通常还可分成概要设计和详细设计。概要设计的任务是模块分解，确定软件结构、模块的功能和模块的接口，以全国数据结构的设计。详细设计的任务是设计每个模块的实现细节和局部数据结构。设计阶段产生的文档有设计说明书，它也可分为概要设计说明书和详细设计说明书。编码的任务是用某种程序语言为每个模块编写程序，产生的文档有程序清单。软件测试的任务是发现软件中的错误，并加以纠正，产生的文档有软件测试计划和软件测试计划和软件测试报告。

（3）软件维护

软件维护任务就是为使软件适应外界环境的变化，进一步实现软件功能的扩充和质量的改善而修改软件。该阶段产生的文档有维护计划和维护报告。

1.4.1.2 软件开发模型

用不同的方式将软件生存周期中所有开发活动组织起来，形成不同的软件开发模型。常见的软件开发模型有瀑布模型、螺旋模型和喷泉模型等。瀑布模型给出了软件生存周期各阶段的固定顺序，上一阶段完成后才能进入到下一阶段。各阶段结束后，都要进行严格的评审。

1.4.1.3 结构化分析和设计方法

（1）结构化分析（SA）方法

结构化分析（SA）方法是一种面向数据流的需求分析方法，它适用于分析大型数据处理系统。结构化分析方法的基本思想是自顶向下逐层分解，把一个问题分解成若干个小问题，每个小问题再分解成若干个更小的问题，经过多次逐层分解，每个最低层的问题都是足够简单、容易解决的，这个过程就是分解的过程。SA 方法的分析结果由数据流图 DFD、数据词典和加工逻辑说明几个部分组成。

（2）结构化设计（SD）方法

结构化设计（SD）方法是一各面向数据流的软件设计方法，它可以与 SA 方法衔接，SD 方法采用结构图（SC）来描述程序的结构。结构图的基本成分由模块、调用和输入/输出数据组成。在需求分析阶段，用 SA 方法产生了数据流图。面向数据流的设计能方便地将 DFD 转换成程序结构图，DFD 中从系统的输入数据到系统的输出数据流的一连串连续变换将形成一条信息流。DFD 的信息流大体可分为两种类型，一种是变换流，另一种是事务流。SD 方法的设计步骤有：①复查并精化数据流图；②确定 DFD 的信息流类型；③根据信息流类型分别将变换流或事务流转换成程序结构图；④根据软件设计的原则对程序结构图作为改进。

（3）结构化程序设计（SP）

结构化程序设计（SP）采用自顶向下逐步求精的设计方法和单入口单出口的控制结构。自顶向下逐步求精的设计方法符合抽象和分解的原则， 人们解决复杂问题时常用的方法。SA 方法和 SD 方法也采用了自顶向下逐步求精的方法，在详细设计时也同样如此。在设计一个模块的实现算法时，先考虑整体后考虑局部，先抽象后具体，通地逐步细化，最后得到详细的实现算法。单入口单出口的控制结构，使程序的静态和动态结构执行过程一致，使程序具有良好的结构。

1.4.1.4 面向数据结构的设计方法

这类方法以数据结构作为设计基础，根据输入/输出数据结构导出程序的结构。Jackson 方法是一种典型的面向数据结构的设计方法。尽管程序中实际使用中的数据结构有许多种，但这些数据结构中数据元素间的逻辑关系只有顺序、选择和重复 3 类。Jackson 方法的设计步骤为：

（1）分析并确定输入和输出数据的逻辑结构，并用 Jackson 图表示；
（2）找出输入数据结构与输出数据结构间有对应关系的数据单元；
（3）从描述数据结构的 Jackson 图导出描述程序结构的 Jackson 图。

1.4.1.5 软件设计的原则

（1）抽象的原则

软件工程中从软件定义到软件开发要发经历多个阶段，在这个过程中每前进一步都可看

作是对软件设计的抽象层次的一次细化。抽象的最低层次就是实现该软件的源程序代码。在进行曲模块化设计时也可以有多个抽象层次，最高抽象层次的模块用概括的方式叙述题的解法，较低抽象层次的模块是对较高抽象层次模块问题解法描述的细化。过程抽象和数据抽象是常用的两种主要抽象手段。

（2）模块化的原则

模块化是指将一个待开发的软件分解或成若干个小的简单的部分模块，每个模块可独立地开发、测试，最后组装成完整的软件。

（3）信息隐蔽的原则

信息隐蔽是开发整体程序结构时使用的法则，即将每个程序的成分隐蔽或封装在一个单一的设计模块中，定义每一人模块时尽可能少地显露其内部的处理。信息隐蔽原则对提高软件的可修改性、可测试性和可移植性都有重要的作用。

（4）模块独立的原则

模块独立是指每个模块完成一个相对独立的特定子功能，并且与其他模块之间的联系比较简单。衡量模块独立程度标准有两个：耦合和内聚，耦合是指模块之间联系的紧密程度，耦合度越高，则模块的独立性越差。内聚是指模块内部各元素之间的紧密程度，耦合度越高，则模块的独立性越差。内聚是指模块内部各元素之间联系的紧密程度，内聚度越低，模块的独立性越差。模块独立要求每个模块都是高内聚低耦合的。

1.4.1.6 编码

编码阶段的任务就是根据详细的设计说明书编写程序。要编写高质量的程序，应注意选择合适的程序设计语言，明确源程序的质量要求，养成良好的程序设计格。

1.4.1.7 软件测试

软件测试的工作量约占软件开发总工作量的 40%以上，其目的是尽可能多地发现软件产品（主要是指程序）中的错误和缺陷。

测试的关键是测试用例的设计，设计方法可分成两类：白盒测试和黑盒测试。白盒测试把程序看成是装在一只透明的盒子里，测试者完全了解程序的结构和处理过程。白盒测试根据程序的内部逻辑来设计测试用例，检查程序中的逻辑通路是否都按预定的要求正确地工作。黑盒测试把程序看成是装在一只不透明的盒子里，测试者完全不了解（或不考虑）程序的结构和处理过程。黑盒测试根据规格说明书规定的功能来设计测试用例，检查程序的功能是否符合规格说明的要求。

软件测试的主要步骤有单元测试、集成测试和确认测试。单元测试也称模块测试，通常单元测试可放在编码阶段，主要用来发现编码和详细设计中产生的错误，一般采用白盒测试。集成测试也租组装测试，它是对由各模块组装而成的模块进行测试，主要检查模块间的接口和通信。集成测试主要用来发现设计阶段产生的错误，通常采用黑盒测试。确认测试的任务是检查软件的功能、性能和其他特征是否与用户的需求一致，它是以需求规格说明书作为依据的测试，通常采用黑盒测度。

大多数软件生产者使用一种 Alpha 测试和 Beta 测试的过程，来揭露仅由最终用户才能发现的错误。 Alpha 测试是在开发者的现场由客户来实施的，被测试的软件是在开发者从用户的角度进行常规设置的环境下运行的。Beta 测试是在一个或多个客户的现场由该软件的最终用户实施的。与 Alpaha 测试不同的是，进行 Beta 测试时开发者通常是不在场的。

2.4.1.8 面向对象方法的基本概念

面向对象（OO）方法成为软件开发的一种主要方法。它有几个基本概念。

（1）对象

在计算机系统中，对象是指一组属性以及这组属性上的专用操作的封装体。属性可以是一些数据，也可以是另一个对象。每个对象都有它自己的属性值，表示该对象的状态，用户只能看见对象封装界面上的信息，对象的内部实现对用户是隐蔽的。封装目的是使对象的使用者和生产者分离，使对象的定义和实现分开。一个对象通常可由对象名、属性和操作这 3 个部分组成。

（2）类

类是一组具有相同属性和相同操作的对象的集合。一个类中的每个对象都是这个类的一个实例（Instance ）。在分析和设计时，我们通常把注意力集中在类上，而不是具体的对象上。通常把一个类和这个类的所有对象称为类及对象或对象类。

（3） 继承

继承是在某个类的层次关联中不同的类共享属性和操作的一种机制。一个父类可以有多个子类，这些子类都是父类的特例。父类描述了这些子类的公共属性的操作，子类中还可以定义它自己的属性和操作。一个子类只有唯一的一个父类，这种继承称为单一继承。一个了类有多个父类，可以从多个父类中继承特性，这种继承称为重多继承。

（4）消息

消息的对象间通信的手段、一个对象通过向另一对象发送消息来请求其服务。一个消息通常包括接收对象名、调用的操作名和适当的参数（如有必要）。消息只告诉接收对象需要完成什么操作，但并不能指示接收者息样完成操作。消息完全同接收者解释，接收者独立决定采用什么方法来完成所需的操作。

（5）多态性和动态绑定

多态性是指同一个操作作用不同的对象可以有不同的解释，产生不同的执行结果。

与多态性密切相关的一个概念就是动态绑定。传统的程序设计语言把过程调用与目标代码的连接放在程序运行前进行，称为静态绑定。而动态绑定则是把这种连接推迟支动行时才进行。在运行过程中，当一个对象发送消息请求服务时，要根据接收对象的具体情况将请求的操作与实现的方法连接，即动态绑定。

2.4.1.9 面向对象的分析与设计方法

（1）Peter Coad 和 Edward Yourdon 的 OOA 和 OOD 方法

OOA（面向对象的分析）模型由 5 个层次（主题层、对象类层、结构层、属性层和服务层）和 5 个活动（标识对象类、标识结构、定义主题、定义属性和定义服务）组成。在这种方法中定义了两种对象类之间的结构，一种称为分类结构，一种称为组装结构。分类结构就是所谓的一般与特殊的关系。组装结构则反映了对象之间的整体与部分的关系。

OOA 在定义属性的同时，要识别实例连接。实例连接是一个实例与另一个实例的映射关系。

OOA 在定义服务的同时要识别消息连接。当一个对象需要向另一对象发送消息时，它

们之间就存在消息连接。

OOA 中的 5 个层次和 5 个活动继续贯穿在 OOD（画向对象的设计）过程中。OOD 模型由 4 个部分组成。它们分别是设计问题域部分、设计人机交互部分、设计任务管理部分和设计数据管理部分 。

（2）Booch 的 OOD 方法

Booch 认为软件开发是一个螺旋上升的过程。在螺旋上升的每个周期中，有 4 个步骤：标识类和对象、确定它们的含义、标识它们之间的关系、说明每一个类的界面和实现。

（3）OMT 方法

对象建模技术 OMT 定义了 3 种模型，它们是对象模型、动态模型和功能模型，OMT 用这 3 种模型来描述系统。OMT 方法有 4 个步骤：分析、系统设计、对象设计和实现。OMT 方法的每一个步骤都使用这 3 种模型，每一个步骤对这 3 种模型不断地进行细化和扩充。

对象模型描述系统包括对象的静态结构、对象之间的关系、对象的属性和对象的操作。OMT 的对象模型中除了对象、类和继承外，还有链、关联、泛化、聚合和模块等概念。

动态模型用来描述与值的变换有关的系统特征--功能、映射、约束和函数依赖。功能模型用数据流图来表示。

OMT 主要步骤：

①分析是 OMT 方法的第一步，其目的是建立可理解现实世界模型。

②在系统设计阶段将确定整个系统的体系结构，以形成求解问题和建立解答的高层次策略。

③对象设计。在分析的基础上，对象设计阶段将建立基于分析模型的设计模型，并考虑实现的细节。设计人员会根据系统设计期间建立的策略把实现细节加入到设计模型中。

④实现阶段将对象设计阶段开发的对象类及基关系转换成特定的程序设计语言、数据库或硬件实现。

2.4.1.10 软件质量保证

软件质量保证是指为保证软件系统或软件产品最大限度地满足用户要求而进行的有计划、有组织的活动，其目的是生产高质量的软件。有多种软件质量模型来描述软件质量特性，著名的有 ISO/IEC 9126 软件质量模型和 Mc Call 软件质量模型。

软件质量保证环节包括的主要工作有：应用技术方法、进行正规的技术评审、测试软件、标准的实施、控制变动、度量、记录保存和报告。

2．4．l．11 软件开发工具与环境

用来辅助软件开发、运行、维护、管理和支持等过程中的活动的软件称为软件工具，通常也称为 CASE 工具。软件工具大都包含了检测机制，能及时发现一些错误，对提高软件的 质量起着重要的作用。

软件开发环境则把一组相关的工具集成在环境中，环境机制提供数据集成、控制集成和界面集成等机制。数据集成机制为工具提供统一的数据接日；控制集成机制实现工具间的通信和协同工作；界面集成机制使这些工具具有统一的界面风格，从而为软件开发、维护、管理等过程中的各项活动提供连续的、一致的全方位支持。

1.4.2 试题解析

对高级程序员级考试来说，软件工程的重要性非常突出，从历年试题统计（见表 2－6）

来看，在各大知识点中平均所占的比重最大，平均每年有 2 道题，这也说明对高级程序员的软件分析与设计能力的要求比较高。总的来说，软件设计方法、软件测试方法、软件模块划分、软件质量要求等是反复考查的重点内容。软件管理、软件工具、软件开发环境等从未考过的内容也应引起重视。

表 2-6　历年程序员级数据结构基础试题统计

试题	考查知识点
1990 年试题 1	可移植性（属于软件质量要求）
1990 年试题 5	软件模块划分，涉及模块内聚
1991 年试题 1	软件测试
1991 年试题 5	软件模块划分，涉及模块耦合
1992 年试题 1	软件质量要求
1992 年试题 3	Jackson 结构化程序设计方法（面向数据结构）
1992 年试题 5	软件测试
1993 年试题 4	结构化设计方法
1993 年试题 5	软件工程标准
1994 年试题 5	软件文档编制
1994 年试题 6	结构测试
1995 年试题 4	软件维护，涉及可维护性
1995 年试题 6	软件测试

试题 1 （2000 年试题 5）

从供选择的答案中，选出应填人下面叙述中{ }内的最确切的解答，把相应编号写在答卷的对应栏内。

在软件开发过程中常用图作为描述工具。如 DFD 就是面向 A 分析方法的描述工具。在一套分层 DFD 中，如果某一张图中有 N 个加工（Process),则这张图允许有 B 张子图。在一张 DFD 图中，任意两个加工之间 C。在画分层 DFD 时，应注意保持 D 之间的平衡。DFD 中从系统的输入流到系统的输出流的一连串连续交换形成一种信息流，这种信息流可分为 E 两类。

供选择的答案

A：①数据结构 ②数据流 ③对象 ④构件（ComPonent）

B：①0 ②1 ③1-N ④0-N

C：①有且仅有一条数据流

②至少有一条数据流

③可以有 0 条或多条名宇互不相同的数据流

④可以有 0 或多条数据流，但允许其中有若干条名宇相同的数据流

D：①父图与其子图②同一父图的所有子图

③不同父图的所有子图④同一子图的所有直接父图

E：①控制流和变换流②变换流和事务流

③事务流和事件流④事件流和控制流

【解析】

本题考查数据流图（DFD）的基本知识。

在软件需求分析阶段，用 SA 方法产生了数据流图。数据流图是结构化分析方法的一种分析结果，用来描述数据流从输入到输出的变换过程。数据流图的基本成份有数据流、加工。文件和源／宿。

一个软件系统，其数据流图往往有多层。如果父图有 N 个加工，则该父图可以有 0～N 张子图，但是每张子图只能孙对应于一张父图。

在画数据流图时，应注意父图与子图的平衡，即父图中某加工的输入输出数据流必须与

其子图的输入输出流在数量和名字上相同。
DFD 的信息流大体可分为两种类型，一种是变换流，另一种是事务流。
［答案］ A：②B：④C：③D：①E：②

试题 2 （2000 年试题 6）

从供选择的答案中，选出应填入下面叙述中{ }内的最确切的解答，把相应编号写在答卷的对应栏内。
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 A 。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 B 有助于提高软件的易理解性；C 有助于提高软件的易修改性。
在软件质量特性中，D 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；E 是指防止对程序及数据的非授权访问的能力。
供选择的答案
A：①易使用性②易恢复性③易替换性④易测试性
B：①增强健壮性②信息隐蔽原则
③良好的编程风格④高效的算法
C：①高效的算法②信息隐蔽原则③增强健壮性④身份认证
D：①正确性②准确性③可靠性④易使用性
E：①安全性②适应性③灵活性④容错性
【解析】
本题考软件质量要求和软件质量特性。
对于源程序的质量要求，最基本的就是正确性和可靠性，除此之外，更要注重程序的易使用性、易维护性和易移植性。易使用性就是要对用户友好，便于用户使用，做到少培训甚至零培训；易维护性包括易理解性、易测试性和易修改性；移植性则指程序从某一环境移植到另一环境的能力。信息隐蔽是开发整体程序结构时使用的法则，即将每个程序的成分隐蔽或封装在一个单一的设计模块中，定义每一个模块时尽可能少地显露其内部的处理。信息隐蔽原则对提高软件的可修改性、可测试性和可移植性都有重要的作用。
一般使用 ISOflEC 9126 软件质量模型来描述软件质量特性，它有 3 个层次组成：第 1 层是质量特性；第 2 层是质量于特性；第 3 层是度量指标。题中提到的可靠性位于第 1 层，安全性位于第 2 层。
[答案]A：④ B：③ C：② D：③ E：①

试题 3（1999 年试题 5）

从供选择的答案中，选出应填入下面叙述中{ }内的最确切的解答，把相应编号写在答卷的对应栏内。
软件设计中划分模块的一个准则是 A 。两个模块之间的耦合方式中，B 耦合的耦合度最高，C 耦合的耦合度最低。一个模块内部的内聚种类中 D 内聚的内聚度最高，E 内聚的内聚度最低。
供选择的答案
A：①低内聚低耦②低内聚高耦合③高内聚低耦合④高内聚高耦合
B：①数据②非直接③控制④内容
C：①数据②非直接③控制④内容

D：①偶然②逻辑③功能④过程
E：①偶然②逻辑③功能④过程
【解析]
软件设计中划分模块的一个准则是追求更高的内聚度和更低的耦合度。耦合度是对一个软件结构内不同模块之间互连程度的度量。耦合强弱取决于模块间接日的复杂程度、进入访问一个模块的点以及通过接口的数据。在软件设计中应该追求尽可能松散的耦合系统，在这样的系统中可以研究、测试或修改、维护任何一个模块，而不需要对系统的其他模块有很多了解或影响其他模块的实现。此外，当某处发生错误时，低耦合度系统的错误传播的范围相应小些。因此，模块间的耦合程度直接影响系统的可理解性、可测试性、可靠性和可维护性。
如果两个模块中的每一个都能够独立地工作而不需要另一个模块的存在，那么他们之间就没有耦合关系；如果两个模块彼此通过参数交换数据，而交换的信息仅仅是数据，那么这种耦合称为数据耦合；如果传递的信息中也有控制信息，则这种耦合称为控制耦合。
当两个或多个模块通过公共数据环境相互作用时，它们之间的耦合称为公共耦合；如果两个模块之间有下列情况之一，则称两个模块之间的耦合关系为公共耦合。（1）一个模块访问另一个模块的内部数据；（2）一个模块没有通过正常人口而转到另一个模块内部；（3）两个模块有一部分程序代码重叠；（4）一个模块有多个入口。
数据耦合的耦合程度最低，控制耦合次之，适当分解模块可以用数据耦合代替控制耦合。公共耦合的复杂程度随耦合模块的个数和耦合模块对公共数据环境的操作而变化。内容耦合是程度最高的耦合，在程序中应该坚决避免内容耦合。
内聚标志着一个模块内各个元素彼此结合的紧密程度，它是信息隐蔽和局部化概念的自然扩展。简单地说，理想的内聚模块只做一件事情。
如果一个模块完成一组任务，这些任务彼此间即使有关系，也是松散的关系，就叫做偶然内聚；如果一个模块完成的任务在逻辑上属于相同或相似的操作，则称为逻辑内聚；如果一个模块包含的任务必须在同一段时间内执行，就叫做时间内聚。
如果一个模块内的处理元素是相关的。而且必须以特定次序执行，则称为过程内聚；如果模块中所有的元素都使用同一个输入数据和（或）产生同一个输出数据，则称为通信内聚。
如果一个模块内的处理元素和同一个功能密切相关，而 B 这些外排必须顺序执行．则称为顺序内聚 如果模块内所在处理元素属于一个整体，完成一个单一的功能，则称为功能内 聚。
内聚程度从低到高的排列是：偶然内聚、逻辑内聚、时间内聚、过程内聚、通信内聚。顺序内聚、功能内聚。
【答案】A：③B：④C：②D：③E：①

试题 4 （1999 年试题 6）

从供选择的答案中，选出应填入下面叙述中{ }内的最确切的解答，把相应编号写在答卷的对应栏内。
OMT 是一种对象建模技术，它定义了 3 种模型，它们分别是 A 模型、B 模型和 C 模型。其中， A 模型描述了系统中对象的静态结构，以及对象之间的联系，B 模型描述系统中与时间和操作顺序有关的系统特征，表示瞬时的行为上的系统的“控制”特征，通常可用 D 来表示，C 模型描述了与值的变换有关的系统特征，通常可用 E 来表示。
供选择的答案
A：①对象②功能③ER④静态
B：①控制②时序③动态④实时
C：①对象②功能③变换④计算
D：①类图②状态图③对象图④数据流图

E：①类图②状态图③对象图④数据流图

【解析】

OMT（对象建模技术）是一种围绕真实世界的概念来组织模型的软件开发方法。OMT 从问题陈述开始，理解问题陈述中的客观世界，将其本质抽象成模型表示，建立系统的 3 种模型，即对象模型、动态模型和功能模型。

对象模型描述了系统中对象的静态结构以及对象间的联系，用对象模型图来表示。对象模型图是 ER 图的一种拓广形式。动态模型描述了与时间和操作次序有关的系统属性，动态模型由多张状态图组成。各个类的状态图通过共享事件组成系统的动态模型。功能模型描述系统内数据值的变化，它由数据流图组成。数据流图说明数据流是如何从外部输入、经过操作和内部存储而到外部输出的。OMT 的 3 种模型相辅相成，组成系统的一个完整的正交视图。

OMT 方法体现了面向对象的系统开发方法的基本特点，强调对系统结构的理解，而不是系统功能的分解。在 OMT 方法中，对象模型最重要，动态模型次之，最后是功能模型。

使用面向对象的基本思想构造的系统模型与客观系统的结构十分类似，可以使用该模型与用户通信。

在分析阶段产生系统模型后，系统设计的任务主要是细化模型。分析和设计可使用统一表示方法，省略了类似结构方法中从数据流图到模块调用层次的转换过程，而且这种模型用面向对象的程序设计语言来实现也显得十分自然。

相对于传统的软件工程方模学，OMT 的开发重心转移到了分析阶段，使得分析的结果比一般的软件开发方法更为可靠，减少了因分析不透彻引起的问题。OMT 支持系统的无缝开发。在整个开发过程中使用统一的软件概念即对象，所有其他概念都是围绕对象组成的，在分析阶段开发的对象模型也适用于设计和实现阶段。这样，软件开发的阶段性就不那么明显了。由于各阶段是一致吻合的，很容易实现各阶段的反复，而且每一次反复都是对系统的进一步深化。

但 OMT 方法还存在以下几个方面的不足：(1）对问题陈述论述得不足；(2）3 种模型的一致性难以检测和维护，而且对系统约束的描述能力也不足：(3）建模过程描述得不很清晰，建模的结果即各种图形表示也不能体现建模的过程，从而增加了分析人员掌握该方法的难度；(4）OMT 方法使用的图形在分析大系统时显得条理不够清晰。

［答案］A：①B：③C：②D：②E：④

试题 5（1998 年试题 1）

从供选择的答案中，选出应填入下面叙述中{ }内的最确切的解答，把相应编号写在答卷的对应栏内。

设计高质量的软件是软件设计追求的一个重要目标。可移植性、可维护性、可靠性、效率、可理解性和可使用性等都是评价软件质量的重要方面。

可移植性反映出把一个原先在某种硬件或软件环境下正常运行的软件移植到另一个硬件或软件环境下，使该软件也能正确地运行的难易程度。为了提高软件的可移植性，应注意 提高软件的 A 。

可维护性通常包括 B 。通常认为，软件维护工作包括改正性维护、C 维护和 D 维护。其中 C 维护则是为了扩充软件的功能或提高原有软件的性能而进行的维护活动。E 是指当系统万一遇到未预料的情况时，能够按照预定的方式做合适的处理。

供选择的答案

A：①使用方便性②简洁性③可靠性④设备不依赖性

B：①可用性和可理解性②可修改性、数据独立性和数据一致性

③可测试性和稳定性④可理解性、可修改性和可测试性

C、D：①功能性②扩展性③合理性④完善性
⑤合法性③适应性
E：①可用性②正确性③稳定性④健壮性
【解析】
软件的可移植性是指把程序从一种硬件配置或软件系统环境转移到另一种配置和环境时，需要的工作量的多少。提高软件可移植性的关键在于提高软件的设备无关性，即设备不依赖展性。
软件的可维护性通常包括叮理解性、可修改性和可测试性。按照每次维护的具体目标，软件维护工作可分为 3 类：改正性维护、完善性维护和适应性维护。
改正性维护的目的在于纠正开发期间未能发现的错误。由于软件测试的不彻底性，任何大型软件在交付使用后都会发现一些潜藏的错误，对它们进行的诊断和改正就称为改正性维 护，改正性维护约占总维护的 20%。
完善性维护指的是任何软件，无论是应用软件还是系统软件，在使用期间都要不断改善，加强产品的功能和性能，以满足用户日益增长的需求，提高自己产品的市场竞争力。刚投入使用的是第 1 版，以后就可能不断升级为第 2 版、第 3 版等，在整个维护工作量中，完善性维护所占比重最大，约占 50～60%。
适应性维护是指软件为适应运行环境的变化而进行的一种维护，如硬件或支撑软件（如操作系统升级）改变引起的变化，将软件移植到其他的运行平台上等，这类维护大约占总维量的 25%。
软件的健壮性是指在硬件发生故障、输入的数据无效或操作错误等意外环境下，即系统遇到未预料的情况时，系统能够做出适当响应的程度。
［答案］A：④B：④C：④D：③E：④

试题 6 （1997 年试题 6）
从供选择的答案中，选出应填入下面叙述中{ }内的最确切的解答，把相应编号写在答卷的对应栏内。
在设计测试用例时，A 是用得最多的一种黑盒测试方法。在黑盒测试方法中，等价类划分方法设计测试用例的步骤是：
1．根据输入条件把数目极多的输入数据划分成若干个有效等价类和若干个无效等价类；
2．设计一个测试用例，使其覆盖 B 尚未被覆盖的有效等价类，重复这一步，直至所有的有效等价类均被覆盖。
3．设计一个测试用例，使其覆盖 C 尚未被覆盖的无效等价类，重复这一步，直至所有的无效等价类均被覆盖。
因果图方法是根据 D 之间的因果关系来设计测试用例的。
在实际应用中，一旦纠正了程序中的错误后，还应选择部分或全部原先已测试过的测试用例，对修改后的程序重新测试，这种测试称为 E 。
供选择的答案
A：①等价类划分②边值分析③因果图④判定表
B、C：①1 个②7 个左右③一半④尽可能少的
⑤尽可能多的③全部
D：①输入与输出②设计与实现③条件与结果④主程序与子程序
E：①验收测试②强度测试③系统测试④回归测试
[解析]
等价类划分是典型的黑盒测试方法，其方法是把程序的输入域划分成若干部分，然后从每个

部分中选取少数代表性数据当作测试用例。列举所有可能的测试数据进行测试工作量太大，通常选取一部分测试数据进行测试，因此如何选取数据就成为关键的一步。用这种方法设计测试用例需要经过划分等价类、选取测试用例两个步骤。

划分等价类时，首先把数目极多的输入分成若干个等价类。所谓等价类就是某个输入域的集合，对于一个等价共中的输入值来说，它们揭示程序中错误的作用是等效的。如果我们的测试用例全部从一个等价类中选取，测试工作将不能保证软件质量。

根据列出的等价类表，应该按照以下步骤确定测试用例：

（1）为每个等价类规定一个唯一的编号；

（2）设计一个等价类，使其尽可能多地覆盖尚未覆盖的有效等价类。重复这一步，最后使得所有有效等价类都被测试用例所覆盖；

（3）设计一个新的测试用例，使其只覆盖一个无效等价类。重复这一步使所有无效等价类都被覆盖。

应当注意到，(3) 中规定一次只能覆盖一个无效等价类，因为一个测试用例中如果含有多个错误，有可能在测试中只发现其中的一个，另一些被忽视。

因果图法是根据输入与输出之间的因果关系来设计测试用例的，要检查输入条件的各种组合情况，在设计测试用例时，需分析规格说明中哪些是原因，哪些是结果，并指出原因和结果之间、原因和原因之间的对应关系。因果图法最终生成的是判定表；功能图方法利用功能图形式化地表示程序的功能说明，并机械地生成功能图的测试用例；回归测试是在纠正了程序中的错误后，选择部分或全部原先己测试过的测试用例，对修改后的程序重新测试，以 验证对软件修改后有没有引出新的错误；强度测试是检查在系统运行环境发生故障的情况下，系统可以运行到何种程度的测试；系统测试是将通过确认测试的软件作为整个基于计算机系统的一个元素，与计算机硬件、外设、支持软件、数据以及人员等其他系统元素结合在一起，在实际运行环境对计算机系统进行一系列的组装测试和确认测试；验收测试是在系统进行有效性测试及软件配置审查后，以用户为主进行的测试。

【答案】A：②B：⑤C：①D：①E：④

试题 7 （1996 年试题 2）

从供选择的答案中，选出应填入下面叙述中{ }内的最确切的解答，把相应编号写在答卷的对应栏内。

在软件工程的设计阶段中，有 3 种常用的设计方法：结构化设计（SD）方法、Jackson 方法和 Parnas 方法。SD 方法侧重于 A ；Jackson 方法则是 B ；Pmas 方法的主要思想 C 。从 70 年代中期到 90 年代早期，D 是最为常用的设计方法。E 法只提供了重要的设计准则，没有规定出具体的工作步骤。

供选择的答案

A～C：

①使用对象、类和继。②由数据结构导出模块结构。

③模块要相对独立，且功能单一，使块间联系弱，块内联系强。

④将可能引起变化的因素隐藏在某有关模块内部，使这些因素变化时的影响范围受到限制。

⑤用数据流图表示系统的分解，且用数据词典和小说明分别表示数据和加工的含义。

⑥自顶向下、逐步细化，采用顺序、选择和循环 3 种基本结构，以及限制 gotO 语句的使用，设计出可靠的和易维护的软件。

D：①SD②Jackson③Parnas④面向对象

E：①SD②Jackson③Parnas④以上皆非。

【解析】

结构化设计（SD）方法、Jackson 方法和 Parnas 方法是软件工程设计阶段常用的设计方法。结构化设计（SD）采用自项向下逐步求精、模块化的设计方法，单入日、单出日的控制结构，利用程序结构图表达模块之间的关系。可见结构化设计方法是以模块化设计为中心，在开始设计时就把待开发系统划分为若干个相互独立的模块，每一个模块的工作明确清晰，模块之间的耦合量低，可减少修改或重新设计时的工作量。结构化设计方法的关键是要恰到好处地划分模块，采用试探方法处理好模块内部以及模块之间的联系，从而达到逐步疏清条理的目的。所以问题 A 选择③。结构化设计方法是 70 年代中期到 90 年代早期最常用的设计方法，问题 D 选择①。

Jackson 方法是一种面向数据结构的设计方法，设计目标是得出对程序处理过程的描述， 其设计过程是从描绘数据结构的 Jackson 图推导出描绘程序结构的 Jackson 图，该方法适用于比较简单的数据处理系统，所以问题 B 选择②。

Parnas 方法强调在概要设计时应预先估计在未来生存周期中可能发生的情况，并采取相应措施来提高系统的可维护性和可靠性。信息隐藏是提高软件可维护性的重要措施，在分解模块时，就应采取措施，将一些将来可能发生变化的因素隐含在某模块内，使将来因修改造成的影响尽可能地局限在一个或少数几个模块中，这种方法只提供了重要的设计准则，而没有规定具体的工作步骤，所以问题 C 选择④，问题 E 选择③。

［答案］ A：③ B：② C：④ D：① E：③

试题 8 （1996 年试题 7）

从供选择的答案中，选出应填入下面叙述中{ }内的最确切的解答，把相应编号写在答卷的对应栏内。

软件测试的目的是 A 。通常 B 是在代码编写阶段可进行的测试，它是整个测试工作的基础。

逻辑覆盖标准主要用于 C 。它主要包括条件覆盖、条件组合（多重条件）覆盖。判定覆盖、条件及判定覆盖、语旬覆盖和路径覆盖等几种，其中除路径覆盖外最弱的覆盖标准是 D ，最强的覆盖标准是 E 。

供选择的答案

A：①表明软件的正确性②评价软件质量

③尽可能发现软件中错误④判定软件是否合格

B：①系统测试②安装测试③验收测试④单元测试

C：①黑盒测试方法②白盒测试方法③灰盒测试方法④软件验证方法

D、E：①条件覆盖②条件组合覆盖③判定覆盖

④条件及判定覆盖⑤语句覆盖

【解析】

软件测试在软件生存周期中占有重要地位，这不仅是因为测试阶段占用的时间、花费人力和成本占软件开发比重的 40%以上，而且还因为它是保证软件质量的关键步骤。

由于人的主观因素或客观原因，在软件开发过程中不可避免的要产生一些错误。软件测 试的任务是在软件投入运行以前尽可能多地发现并改正软件中的错误，所以问题 A 应选择③。

一个软件产品在交付使用前要经历以下几种测试：

（1）模块测试。也称为单元测试，是针对每个模块单独进行的测试。模块测试一般和程序编写结合起来，在编码阶段由软件编写者进行测试，以保证每个模块作为一个单元能正确运行，所以问题 B 选择④。

（2）于系统测试。把已通过单元测试的模块连接成为子系统来测试，着重检验模块间的接口。

（3）系统测试。把经过测试的子系统连接成为一个完整的系统米测试，主要检查系统是 否实现说明书中指定的功能，系统的动态性能是否符合要求。
（4）验收测试。是为了确认已开发的软件能否满足标准，是否合格。与系统测试不同的是，它是在客户的积极参与下进行的。
（5）平行运行。一些十分重要的软件在经过验收之后，并不立即投入运行，而是再经过一段平行运行的测试，即新旧两个系统同时运行，比较结果有什么不同。
下面再对题目中涉及的一些概念进行解释。
黑盒测试方法又称功能测试，把程序看作一个黑盒子，在完全不考虑程序内部结构的情况下设计测试数据，主要测试程序的功能是否符合软件说明书的要求。
白盒测试方法又称结构测试，它是根据程序的内部结构设计测试数据，检查程序中的每条通路是否都能按要求正确运行。
逻辑覆盖主要用于白盒测试方法，所以问题 C 选择②。由于覆盖的详尽程度不同，又分为语句覆盖、判定覆盖、条件覆盖、条件组合覆盖、条件及判定覆盖和路径覆盖等。
语句覆盖：设计足够多的测试用例，使程序中的每个语句至少执行一次。语句覆盖是最弱的逻辑覆盖准则。
判定覆盖：设计足够多的测试用例，不仅每个语句至少执行一次，而且使得程序中每个判定的每个分支至少执行一次。
条件覆盖：设计足够多的测试用例．不仅每个语句至少执行一枚．而日体每个判定表达式中每个条件都取到可能的结果。
条件组合覆盖：设计足够多的测试用例，使得每个判断的各种可能组合至少出现一次。
条件及判定覆盖：设计足够多的测试用例，使得判断中每个条件的每种可能至少出现一次，而且每个判断的不同判定结果也至少出现一次。
路径覆盖：设计足够多的测试用例，要求程序中所有路径至少执行一次。
在上述几种逻辑覆盖中，除路径覆盖外，最弱的覆盖标准是语句覆盖，最强的覆盖标准是条件组合覆盖。
【答案】A：③ B：④ C：② D：⑤ E：②

试题 9（1995 年试题 4）

从供选择的答案中，选出应填入下面叙述中{ }内的正确答案，把编号写在答卷的对应栏内。
软件维护工作越来越受到重视，因为它的花费常常要占软件生存周期全部花费的 A%左右。其工作内容为 B，为了减少维护工作的困难，可以考虑采取的措施是 C。而软件的可维护性包 D。所谓维护管理主要指的是 E。
供选择的答案：
A：①10～20 ②20～40 ③60～80 ④90 以上
B：①纠正与修改软件中含有的错误。
②因环境已发生变化，软件需作相应的变更。
③为扩充功能，提高性能而作的变更。
④包括上述各点内容。
C：①设法开发出的无错的软件
②增加维护人员数量。
③切实加强维护管理，并在开发过程中就采取有利于未来维护的措施。
④限制个性的范围。
D：①正确性、灵活性、可移植性。
②可测试性、可理解性、可个性性。

③可靠性、可复用性、可用性。
④灵活性、可靠性、高效性。
E：①加强需求分析。 ②重新编码。
③判定修改的合理性并审查修改质量。 ④加强维护人员管理。
【解析】
软件的可维护性、可用性、可靠性构成了衡量软件技师的几个重要尺度。
软件维护是指已经完成发工作，对软件产品所进行的后续活动。一般来说软件维护活动贯穿了软件投入使用直到软件被淘汰的整伸过程，在这个过程中，维护工作需要解决开发阶段所遇到的各种问题和解决某些维护工作本身的特有问题。
软件维护工作包括 3 个方面：改正性维护、适应性维护和完善性维护。改正性维护是在软件运行中发生异常或故障时进行的，这种故障往往是由于软件开发过程中某个环节上的隐患构成的。适应性维护的目的是要使运行的软件能适应外部环境的变动，例如数据格式的变动、数据输入输出方式的变动等都会影响软件的正常工作。完善性维护则是为扩充软件的功能，提高软件的性能而开展的软件维护活动，如用户在使用了一段时间之后对提出了新的要求，这种情况下，就需要完善性维护。在整个软件维护活动中，完善性维护所占的比重最大，平均在 50%左右。统计资料表明，维护阶段的花费占整个软件生存周期花费的 67%，这是一个相当可观的数字。
正确的软件维护工作所应采取的措施是：切实加强维护管理，并在开发过程中就采取有利于软件未来维护的措施。软件的维护不仅是技术性的，而且还需要大量的管理工作与之配合。从原则上讲，维护工作从理解软件开始，在些基础上，如果有明确的维护任务，则针对些任务提出修改建议；在经过部门的审批之后，正确的维护建议被批准；按照修改方案进行的修改结束后，为保证修改的质量，应该进行严格的测试；经过管理部门再次审查后，教授可以对文档进行正式修改。这样，所谓维护管理主要是指判定个性的合理过性并审查修改质量。
【答案】A： ③ B： ④ C： ③ D： ② E： ③

试题 10（1995 年试题 6）

从供选择的答案中，选出应填入下面叙述中{ }内的正确答案，把编号写在答卷的对应栏内。
软件测试是软件质量保证的主要手段之一，测试的费用已超过 A 的 30%以上。因此提高测试的有效性非常重要。"高产"的测试是指 B。根据国家标准 GB8566-88 计算机软件开发规范的规定，软件的开发和维护分为 8 个阶段，其中单元测试是在 C 阶段完成的；组装测试的计划内是在 D 阶段制定的；确认测试的计划是在 E 阶段制定的。
供选择的答案
A：①软件开发费用。 ②软件维护费用。
③；软件开发和维护费用。 ④软件研制费用。
B：①用适量的测试用例，说明被测程序正确无误。
②用适量的测试用例，说明被测程序符合相应的要求。
③用少量的测试用例，发现被测程序尺可能多的错误。
④用少量的测试用例，纠正被测程序尺可能多的错误。
C～E：①可行性研究和计划。②需求分析。③概要设计。④详细设计。
⑤实现。 ⑥组装测试。⑦确认测试。⑧使用和维护。
【解析】
目前，在大中型软件开发项目中，测试都占据着重要地位，同时，测试也是在将软件交付给客户之前所必须完成的步骤。测试所花费用已超过软件开发费用的 30%以上。如何组织好测试，特别是如何选择测试用例，对保障软件质量，降低测试费用有着重要的意义。

一个高效的测试，是指通过对所设计的少量测试用例进行测试，从而发现被测试程序中尺可能我的问题，并完成修改。测试按照被测试的内容可分为 3 种：单元测试(对程序单元或模块单独进行测试)、组装测试(把已通过单元测试的模块连接起来，测试模块间的接口及软件设计中的问题，常用功能测试办法)和确认测试(对软件技师作全面测试，以确认开发的软件是否符合验收标准)。

根据国家标准 GB8566-88 计算机软件开发规范的规定，软件开发和维护分为 8 个阶段，分别为可行性研究和计划、需求分析、概要设计、详细设计、实现、组装测试、确认测试和使用维护。GB8566-88 规定单元测试在实现阶段完成；组装测试在组装测试阶段完成，但组装测试的计划应该在概要设计阶段制订，而确认测试的计划则在需求分析阶段就应该制订好。

【答案】A：① B：③ C：⑤ D：③ E：②

试题 11（1994 年试题 5)

从供选择的答案中，选出应填入{ }内的正确答案，把编号写在答卷的对应栏内。

国家标准《计算机软件产品开发文件编制指南 GB8567-88》中规定，在一项软件开发过程，一般地说应该产生 14 种文件，其中管理人员主要使用的有 A、B、C、开发进度月报、项目开发总结报告。开发人员主要使用的有 A、B、D、数据要求说明书、概要设计说明书、详细设计说明书、数据库设计说明书、测试计划和 E。维护人员主要使用的有设计说明书、E 和 C。

A～E： ①软件需求说明书 ②项目开发计划 ③可行性研究报告

④模块开发卷宗 ⑤测试分析报告 ⑥操作手册 ⑦用户手册

【解析】

《计算机软件产品开发文件编制指击 GB8567-88》是《计算机软件开发规范 B8566-88》的配套文件，由国家标准局在 1988 年 1 月批准并发布，规范中详细规定了软件开发过程中各个阶段及每一阶段的任务、实施步骤、实施要求、完成标志及交付的文本。《计算软件产吕开发文件编制指击 GB567-88》则为应交付的文本提供了编写指南，一般来说，规定在一项软件开发过程中应该产生 14 种文件：

A 可行性研究报告
B 项目开发计划
C 软件需求说明书
D 数据要求说明书
E 概要设计说明书
F 详细设计说明书
G 数据库设计说明书
H 用户手册
I 操作手册
J 模块开发卷宗
K 测试计划
L 测试分析报告
M 开发进度月报
N 项目开发总结报告

其中管理人员主要使用的有项目开发计划、可行性研究报告、模块开发卷宗、开发进度月报和项目开发总结报告；开发人员主要使用的有项目开发计划、可行性研究报告、软件需求说明书、数据要求说明书、概要设计说明书、详细设计说明书、数据库设计说明书、测试计划和测试分析报告；维护人员主要使用的有设计说明书、测试分析报告和模块开发卷宗。

【答案】A：②B：③C：④D：①E：⑤

试题 12（1994 年试题 6）

从供选择的答案中，选出应填入{ }内的正确答案，把编号写在答卷的对应内。
在结构测试用例设计中，有语名覆盖、条件覆盖、判定覆盖(即分枝覆盖)、路径覆盖等，其 A 是最强的覆盖准则。为了对图 2-8 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 2-7)，则实现判定覆盖至少应采用的测试数据组是 B；实现条件覆盖至少采用的测试数据组是 C，实现路径覆盖至少应采用的测试数据组是 D 或 E。

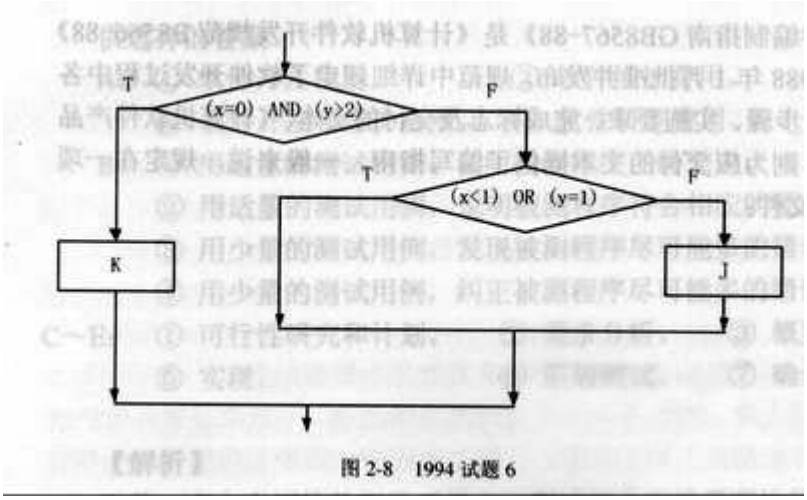


图 2-8　1994 试题 6

表 2-7　1994 试题 6

	x	y
测试数据组Ⅰ	0	3
测试数据组Ⅱ	1	2
测试数据组Ⅲ	−1	2
测试数据组Ⅳ	3	1

供选择的答案
A：①语句覆盖 ②条件覆盖 ③判定覆盖 ④路径覆盖
B～E：①Ⅰ和Ⅱ组 ②Ⅱ和Ⅲ组 ③Ⅲ和Ⅳ组 ④Ⅰ和Ⅳ组⑤Ⅰ、Ⅱ和Ⅲ组 ⑥Ⅱ、Ⅲ和Ⅳ组 ⑦Ⅰ、Ⅲ和Ⅳ组 Ⅰ、Ⅱ和Ⅳ组
【解析】
测试阶段在软件生命周期中占有重要地位。一个软件产品在交付用户正式使用前主要经过 3 种测试：单元测试、集成测试和验收测试。测试的方法有黑盒法和白盒 黑盒方法主要用于动能测试或数据驱动测试；白盒方法主要用于结构测试或逻辑驱动测试。测试用的数据称为测试用例，正确选取测试用例对测试的最终结果有很大影响。白盒测试用例的设计方法有语句覆盖、条件覆盖、分支覆盖、路径覆盖等多种，其中语句覆盖是最弱的覆盖准则，路径覆盖则最强。

本题所给出的 4 组测试数据，要实现分支覆盖即判定覆盖，必须使程序中每个逻辑判断的取真分支和取假分支在行某组测试数据时至少历一次，应采用的测试数据组是第Ⅰ和第Ⅱ组数据；要实现条件覆盖，必须使程序中每个条件的可能取值至少满足一次，应采用的测试数据组是Ⅰ和 Ⅱ组数据；要实现路径覆盖，求程序中的所有路径至少经历一次，则应采用的测试数据组是第Ⅰ、Ⅱ、Ⅲ组或者第Ⅰ、Ⅱ、Ⅳ组。

【答案】A：④ B：① C：④ ⑤ E：⑧

试题 13（1993 年试题 4）

从供选择的答案中，选出应填入下面关于软件设计方法叙述中{ }内的正确答案，把编号写在答卷的对应栏内。

结构化设计方法在软件开发中用于 A，它是一种面向 B 的设计方法。该方法使用的图形工具是 C，C 中矩形表示 D。如果两矩形之间有直线相连，表示它们存在 E 的关系。

供选择的答案

A：①测试用例设计 ②概要设计 ③程序设计 ④详细设计

B：①对象 ②数据结构 ③数据流 ④控制流

C：①程序结构图 ②数据流程图 ③程序流程图 ④实体联系图

D：①数据 ②加工 ③模块 ④存储

E：①链接 ②调用 ③并列 ④顺序执行

【解析】

软件设计工作分成概要设计和详细设计两步。结构化设计(Structured Design，简称 SD)方法是结构化分析(SA)的后续阶段，用于软件的概要设计。SD 是一种面向数据流的设计方法，从整个程序的结构出发，突出程序模块，利用程序结构图表达程序模块之间的关系，尤其适用于变化型结构和事务型结构的数据处理系统。

程序结构图是采用结构化设计方法进行软件概要设计的描述手段，以图形的形式描述软件系统的模块组成及模块之间的调用关系，程序结构图可以精炼地表达模块化的设计思想，清晰地表达模块之间的联系。

构成程序结构图的主要成分有模块、调用和数据，结构图中的模块用矩形表示，在矩形框内可标上模块的名字。模块间如有箭头或直线相连，表明它们之间有调用关系。对于两个处在不同位置的模块，把上面的称为调用模块，下面的称为被调用模块。调用简头表示模块调用时模块间的数据传送，简头的方向指明了传送的方向，图中的数据也用适当的名称加以标识。

【答案】A：② B：③ C：① D：③E：②

试题 14（1993 年试题 5）

从供选择的答案中，选出应填入{ }内的正确答案，把编号写在答卷的对应栏内。

按制定软件工程标准的不同层次和适用范围，软件工程标准可分为 5 级，A 制定的是国际标准，B 是行业标准。GB1256-89 对程序流程图等作了明确、具体规定，这种标准程序流程图的特点有 C、D 和 E。

供选择的答案

A、B：①IEEE，GJB ②IEEE，ANSI ③ISO，IEC

④ISO，IEEE ⑤IEC，GJB ⑥ANSI，ISO

C～E：①箭头表示数据的传递方向。

②允许自行定义多种特定的图形符号。

③对循环结构规定了一对特定的符号。

④它和 ISO 的有关规定有一些差别。

⑤允许有两个以上出口的判断。
⑥特定方向的流线才用箭头标明流向。
⑦不允许在图形符号外加注标识符和描述符。
【解析】
按制定软件工程标准(或规范)的不同层次和适用范围，软件工程标准可划分成国际标准、国家标准、行业标准、企业或地方规范、项目规范等 5 级。ISO(International Standards Organization，国际标准化组织)和 IEC(International Electromechanical Commission，国际电工委员会)制定的是国际标准；IEEE 是指美国电气和电子工程师学会，GJB 是指中华人民共和国军用标准，后两者属于行业标准。
国家标准《GB 1256-89》是《信息处理-数据流程图：程序流程图、系统流程图、程序网络图和系统资源图的文件编符号集约定》，该标准对程序流程图作了明确具体的规定。在这种标准流程图中，对循环结构规定了一对特定的符号，即循环界限为去上角矩形表示的上界限和去下角矩形表示的下界限，分别表示循环的开始和循环的结束：判断用菱形表示，它只有一个入口，但可以有若干个可供选择的出口；控制流的流线用直线表示，流线的标准流向是从左到右和从上到下，沿标准流向的流线可以不用箭头指示流向，但对于沿非标准流向的流线方向，应该用箭头指示方向。
【答案】A：③ B：① C：③ D：⑤ E：⑥

试题 15（1992 年试题 1)

从供选择的答案中选出应填入{ }的正确答案，把编号写在答卷的对应栏内。
软件质量包含多方面的内容，A、B、可移植性和可复用性等是较为重要的质量特征。在软件开发中，必须采取有力的措施，以确保软件的质量，这些措施至少应包括 C、D 和 E
供选择答案
A、B ①稳定性 ②可靠性 ③数据一致性
④可维护性 ⑤可行性 ⑥数据独立性
C、D、E： ①在开发初期制定质量保证计划，并在开发中坚持实行。
②开发工作严格按阶段进行，文档工作应在开发完成后集中进行。
③严格执行阶段评审。
④要求用户参与全部开发过程，以监督开发质量。
⑤开发前选定或制定开发标准或开发规范，亲遵照实施。
⑥争取足够的开发经费和开发人力的支持。
【解析】
软件质量涉及多方面的内容，其中较为重要的质量特性有软件的可靠性、可维护性、可移植性和可复用性等。这些特性可为软件的安全使用、进一步扩充、修改、复制和移植打下良好的基础。稳定性、数据一致性、可行性和数据独立等都不是与软件质量直接相关的概念。
在软件开发中，应采取有力的措施确保软件的质量。有些措施在实践证中已证明是行之有效的。如在开发初期制定质量保证计划，并在开发中坚持实行；严格执行阶段评审，以便及时发现问题；开发前选定或制定开发标准或开发规范，并遵照实施；软件生存期的各阶段都有完整的文档等。
【答案】A：② B：④ C：① D：③ E：⑤

试题 16（1992 年试题 13)

从供选择的答案中选出应填入{ }的正确答案，把编号写在答卷的对应栏内。
Jackson 结构化程序设计方法是英国 M.Jackson 提出的，它是一种面向 A 的设计方法，主要

适用于规模适中的 B 系统的开发，其基本步骤依次是 C、D、E。
供选择的答案
A：①对象 ②数据流 ③数据结构 ④控制结构
B：①数据处理 ②文字处理 ③实时控制 ④科学计算
C、D、E：①建立数据结构 ②列出基本操作 ③建立程序结构
④建立控制结构 ⑤建立对象
【解析】
Jackson 方法由英国的 M.Jackson 提出，Jackson 方法的基本思想同软件设计听 SD 方法是一致的，但它不是面向数据流，而是面向数据结构的。应用该方法的基本步骤依次是：建立系统的数据结构；以数据结构为基础，对应地建立程序结构；列出程序中要用到的各种基本操作，再将这些操作分配程序结构适当的模块中。
Jackson 方法的设计原则是将程序结构和数据结构相对应，该方法特别适用于设计企事业信息管理一类的规模适中的数据处理系统。这些系统处理的数据大多具有层次结构，如文件由记录组成，记录又由数据项组成，所以可以以数据结构为基础，相应地建立模块的层次结构，如处理文件的模块调用处理记录的模块，处理记录的模块又调用处理数据项的模块。
【答案】A：③ B：① C：① D：③ E：②

试题 17（1992 年试题 5）

从供选择的答案中选出应填入{ }内的正确答案，把编号写在答卷的对应栏内。
A 在实验阶段进行，它所依据的模块功能描述和内部细节以及测试方案应在 B 阶段完成，目的是发现编程错误。
C 所依据的模块说明书和测试方案应在 D 阶段完成，它能发现设计错误。
E 应在模拟的环境中进行强度测试的基础上进行，测试计划应在软件需求分析阶段完成。
供选择的答案
A：①用户界面测试 ②输入/输出测试 ③集成测试 ④单元测试
B：①需求分析 ②概要设计 ③详细设计 ④结构设计
C：①集成测试 ②可靠性测试 ③系统性能测试 ④强度测试
D：①编程 ②概要设计 ③维护 ④详细设计
E：①过程测试 ②函数测试 ③确认测试 ④逻辑路径测试
【解析】
一个软件在交付使用前，主要经历 3 种测试：单元测试、集成测试和验收测试。
单元测试也称为模块测试，是针对各个程序单元或模块单独进行的测试，通常称之为"分调"。单元测试在实验阶段进行，一般和程序编写结合起来，由程序员分工进行，并且多模块可以并行展开，单元测试着重发现和解决程序编写中的差错，比较重视对程序结构的检验，单元测试是整个测试阶段的基础。
集成测试把自己通过单元测试的模块连接起来，通常使用功能测试法，着重检验模块间的接口以及设计中的问题。集成测试所依据的模块说明书和测试方案应在概要设计阶段完成。集成测试的目的是发现设计错误。
确认测试也称验收测试，其目的是为了确认已开发的软件能否满足验收标准。确认测试是对软件质量的全面考核，验收测试以前应制定验收标准和验收测试工作。确认测试应在模拟的环境中在强度测试的基础上进行，测试计划应在软件需求分析阶段完成。
【答案】A：④ B：③ C：① D：② E：③

试题 18（1991 年试题 1）

从供选择的答案中选出应填入下列叙述中的{ }内的正确答案，把编号写在答卷的对应栏内。
软件测试的目的是 A。为了提高测试的效率，应该 B。使用白盒测试方法时，确定测试数据应根据 C 和指定的覆盖标准。一般说来与设计测试数据无关的文档是 D。软件的集成测试工作最好由 E 承担，以提高集成测试的效果。
供选择的答案
A：①评价软件的质量 ②发现软件的错误
③找出软件中的所有错误 ④证明软件是正确的
B：①随机地选取测试数据 ②取一切可能的输入数据作为测试数据
③在完成编码以后制定软件的测试计划 ④选择发现错误的可能性大的数据作为测试数据
C：①程序的内部逻辑 ②程序的复杂程度
③使用说明书 ④程序的功能
D：①需求规格说明书 ②设计说明书
③源程序 ④项目开发计划
E：①该软件的设计人员 ②该软件开发组的负责人
③该软件的编程人员 ④不属于该软件开发组的软件设计人员
【解析】
软件测试的工作量约占软件开发期的一半，是软件开发过程的一个重要阶段，直接影响软件的质量，软件测试的目的是为了发现软件的错误，提高测试效率，因此应该选择容易发现错误的数量作为测试数据。
测试数据的方法有两种：黑盒方法和白盒方法。黑盒方法将程序看作不能打开的黑盒，根据程序的功能或程序的外部特性测试数据。白盒测试方法是设计测试用例的一类方法，使用白盒测试方法，允许测试者检查程序的内部结构，以程序的内部逻辑为依据，设计测试数据和指定覆盖标准，因此白盒测试方法又称为逻辑覆盖方法。
一般来说，与设计测试数据有关的文档是需求规格说明书、设计说明书和源程序等文档，与项目开发计划无关。软件集成测试工作最好由不属于该软件开发组的设计人员承担。这样可以使参加测试的人员避免认为自己开发的程序总是正确这种"先入为主"的意识和"当局者迷"的现象，能够用更加客观的态度投入软件集成测试工作，提高测试的效果。
【答案】A：② B：④ C：① D：④ E：④

试题 19（1991 年试题 5）

从供选择的答案中选出应填入下列叙述中{ }内的正确答案，把编号写在答卷的对应栏内。
软件设计中划分程序模块通常遵循的原则是要使各模块间的耦合性尽可能 A。3 种可能的模块耦合是：
B.例如，一个模块直接引用另一模块中的数据。
C.例如，一个模块把开关量作为参数传送给另一模块。
D.例如，一个模块把一个数值量做为参数传送给另一模块。其中 E 的耦合性最强。
供选择的答案
A：①强 ②适中 ③弱
B～E：①公共耦合 ②数据耦合 ③逻辑耦合
④外部耦合 ⑤内容耦合 ⑥控制耦合
【解析】
模块间的联系是评价程序结构质量的重要标准，模块间的互相关联程度由模块的耦合性衡量。耦合性越弱，模块间的联系越小，每个模块越容易独立地被理解、编写和修改；同时某个模块的错误不容易扩散蔓到其他模块，因此，在软件设计中，使各个模块的耦合尽可能弱

将成为划分程序模块的原则。
模块按照其耦合性由强到弱可以分成以下 5 种：
(1)内容耦合 内容耦合指一个模块直接引用另一模块的内部信息；
(2)公共耦合 公共耦合指两个模块引用人同的全局数据区；
(3)控制耦合 控制耦合指一个模块传送给另一模块的参数是用于控制模块内部逻辑的控制信号；
(4)复合耦合 复合耦合指一个模块传送给另一模块的参数是一个复合的数据结构；
(5)数据耦合 数据耦合指一个模块传送给另一模块的参数是单个的数据项。
上述的 5 种耦合中内容的耦合性最强。
【答案】A：③ B：⑤ C：⑥ D：② E：⑤

试题 20（1990 年试题 1）

从供选择的答案中选出应填入下列叙述中{ }内的正确答案，把编号写在答卷的对应栏内。
软件可移植性是用来衡量软件的 A 的重要尺度之一。为了提高软件的可移植性，应注意提高软件的 B。采用 C 有助于提高 B。为了提高可移植性，还应 D。使用 E 语言开发的系统软件具有较好的可移植性。
供选择的答案
A：①通用性 ②效率 ③质量 ④人-机界面
B：①使用的方便性 ②简洁性 ③可靠性 ④设备独立性
C：①优化算法 ②专用设备 ③表格驱动方式 ④树型文件目录
D：①有完备的文档资料 ②选择好的宿主计算机
③减少输入/输出次数 ④选择好的操作系统
E：①Coblo ②APL ③C ④PL/1
【解析】
可移植性是衡量软件质量高低的一个重要尺度之一。可移植性是指把原先在某硬件或软件环境下正常运行的软件，移植以另一个硬件或软件环境下，使该软件也能正确运行的难易程度。设备独立性是指软件对设备的访问不随设备特性的变化而变化，所以，提高软件的设备独立性是提高软件可移植性的重要手段。为提高软件的可移植性，在设计和编写程序时，应注意使程序代码不要随意地与具体环境直接关联，并且不要直接使用硬件环境中的设备，使程序代码尽可能与环境无关，否则，设备变了，程序也要做大量的改动。
对一类常规或有一定通用功能的设备来说，它们都有一套反映其特性的参数。对软件来说 设备之间的差异就是它们的参数不同。为使程序能使用不同参数的设备，可采用表格驱动。表格驱动是指程序内部保留有一张记录着设备特性参数的参数表，这样，软件移植或环境有所改变时，只需修改参数表中若干参数，在程序不作任何修改的情况下，软件便可在新的环境下正确使用。表格驱动方式使程序独立于设备，促进了程序可移植性的实现。
完备的文档资料有助于软件移植的顺利进行。软件移植时，对原来的软件有所修改是难免的，理解原来软件的设计思想和程序采用的算法都依赖于软件的文档资料。
与其创建高级语言相比，C 语言开发的软件具有良好的可移植性，计算机系统大多数都配置有 C 语言编译程序和提供 C 语言的标准库函数。统计资料表明，不同系统上的 C 语言编译程序 80%的代码是公共的。
【答案】A：③ B：④ C：③ D：① E：③

试题 21（1990 年试题 5）

从供选择的答案中选出应填入下列叙述中的{ }内的正确答案，把编号写在答卷的对应栏内。

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。
1.一组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 A 的。
2.将几个逻辑上相似的成分放在一个模块中，该模块的内聚度是 B 的。
3.模块中所有成分引用共同的数据，该模块的内聚度是 C 的。
4.模块内的某成分的输出是另一些成分的输入，该模块的内聚度是 D 的。
5.模块中所有成分结合起来完成一项任务，该模块的内聚度是 E 的。它具有简明外部界面，由它构成的软件易于理解、测试和维护。
供选择的答案
A～E：①功能性 ②顺序性 ③通信性 ④过程性 ⑤偶然性 ⑥瞬时性 ⑦逻辑性。
【解析】
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。在软件设计过程中，设计者在划分程序模块之时，应力求模块的内聚度尽可能强。按模块内聚性分，模块内聚度由弱到强的顺序是：偶然性、逻辑性、瞬时性、过程性、通信性、顺序性和功能性。
一个模块内部的各组语句这宰没有必然的联系，则这个模块是偶然性内聚的；几个逻辑上相似的成分放在一个模块中，这种模块是逻辑内聚的；一些经常需要一起执行的程序段放在一起构成的模块称作瞬时性内聚的；因各成分要引用共同的数据，而把它们组成一个模块，这种模块称作是通信性内聚的；如模块中的一个成分的输出是另一些成分的输入，称这种模块是顺序性内聚的；如模块中的所有成分都是为了完成一项共同的任务，这种模块称作简明的外部界面，其可理解性、可测试性和维护性都比较好，所以功能内聚模块是设计人员应该追求的目标。
【答案】A：⑤ B：⑦ C：③ D：② E：①

1.5 数据库系统基础知识

1.5.1 主要知识点

掌握数据库模型、数据库系统结构和关系数据库结构的基础知识，熟练掌握 SQL 语言的使用 关系代数的运算。

1.5.1.1 关系数据库的数据体系结构

关系模型遵循数据库的 3 级体系结构：

(1)关系模式。数据库的概念模式定义为关系模式的集合。每个关系模式就是记录类型。关系模式的定义包括模式名、属性名、值域名和模式的关键。关系模式仅仅是对数据本身特性的描述。

(2)关系子模式是用户所用到的那部分数据的描述。除了指出用户用到的数据外，还应指出数据与模式中相应数据的联系，即指出子模式与模式之间的对应性。

(3)关系存储是作为文件看待的，每个元组就是一个记录。由于关系模式有键，因此存储一个关系可用散列方法或索引方法实现。

2.5.1.2 关系模型和关系运算

用二维表格结构表示实体集，关键码表示实体间联系的数据模型称为关系模型。在关系中，能唯一标识元组的属性集称为关系候选键，被选用的候选键称为关系的主键。关系中每

一个属性对应一个取值范围，称为属性的值域。关系可以定义为元数(属性个数)相同的元组的组合。关系是一个集合，集合中的成分是元组，这些元组的属性个数应相同。

关系数据库的数据更新操作必须遵循实体完整性规则、引用完整性规则和用户定义的完整性规则。

关系查询语言根据其理论基础的不同分成两大类：一类是关系代数语言，其特征是查询操作是以集合操作为基础的运算；另一类是关系演算语言，其特征是查询操作是以谓词演算为基础的运算。

关系代数是以集合代数为基础发展起来的、以关系为运算对象的一组高级运算的集合。把关系看成集合，集合代数中的运算可以引入到关系运算中来，还有一些运算是针对关系数据库环境专门设计的。

关系代数的 5 种基本操作：并、差、笛卡儿积、投影和选择。

关系代数的 4 种组合操作：交、联接、自然联接和除法。

扩充的关系代数操作：外联接(左外联接和右外联接)、外部并(Out union)和半联接。

2.5.1.3 关系数据库 SQL 语言

结构化查询语言 SQL 是集 DDL、DML 和数据控制功能于一体的数据库语言。SQL 的 DML 是介于关系代数和关系演算之间的一种语言。

SQL 语言的组成：

·一个 SQL 数据库是表的汇集，它用一个或多个 SQL 模式定义。

·一个 SQL 表由行集构成，一行是列的序列，每列对应一个数据项。

·一个表或者是一个基本表，或者是一个视图。基本表是实际存储在数据库中的表，而视图是由若干基本表或其他视图构成的表的定义。

·一个基本表可以跨一个或多个存储文件，一个存储文件也可存放一个或多个基本表。

·用户可以用 SQL 语句对视图和基本表进行查询等操作。

·SQL 用户可以是应用程序，也可以是终端用户。SQL 语句可嵌入在宿主语言的程序中使用，宿主语言可以是常用的高级语言。SQL 用户也能作为独立的用户接口，使交互环境下的终端用户使用。

SQL 包括了所有对数据库的操作，主要有数据定义(SQL DDL)、数据操纵(SQL DML)、访问控制和嵌入式 SQL。

2.5.1.4 关系数据库规范化理论

关系数据库规范化理论主要包括 3 方面的内容：

(1)函数依赖。指数据之间存在的各种联系和约束，例如建就是一种依赖。函数依赖是最基本的一种依赖。

(2)范式。模式分解的标准形式。关系模式分解的两个特性实际上涉及到两个数据库模式的等价性问题，包括数据等价和依赖等价两个方面。数据等价是指两个数据库实例应表示同样的信息内容，用"无损联接"衡量。依赖等价是指两个数据库模式应有相互逻辑关系的函数依赖集，此时数据的语义是不会出差错的。

(3)模式设计方法。设计规范的数据库模式的方法。

衡量关系模式的优劣的标准就是模式的范式(NF)。范式有许多种，与数据依赖有着直接的联系。

①第一范式(INF)

如果关系模式 R 的每个关系 r 的属性值都是不可分的原子值，那么称 R 是第一范式(INF)

的模式，r 是规范化的关系。LNF 的模式的关系数据库最基本的要求。

②第二范式(2NF)

我们把关系模式 R 的属性分为两类：一类是键的属性，称为主属性；另一类是不属于任何键的属性，称为非主属性。若关系模式 R 是 INF，且每个非主属性完全函数依赖于候选键，那么称 R 是 2NF 模式。

③第三范式(3NF)

若关系模式 R 是 INF，且每个非主属性都不传递依赖于 R 的候选集，那么称 R 是 3NF 模式。在 3NF 模式中排除了非主属性的对键的传递依赖。3NF 的模式必定是 2NF 的模式。局部依赖和传递依赖是产生冗余和异常的两个重要原因。由于 3NF 模式中不存在非主属性对候选键的局部依赖和传递依赖，因此具有较好的性能。而对于非 3NF 的 INF 和 2NF，甚至非 INF 的关系模式，于它们性能上的弱点，一般不宜作为数据库模式，通常需要将它们变换成 3NF 或更高级的范式，这种变换过程，称为"关系的规范化处理"。

④BC 范式(简称 BCNF)

若关系模式 R 是 INF，且每个属性都不传递依赖于 R 的候选键，那么称 R 是 BCNF 模式。此时排除了任何属性对键的传递依赖。

上述 4 种范式之间关系：BCNF $\subset$ 3NF $\subset$ 2NF $\subset$ 1NF。

1.5.2 试题解析

高级程序员级考试中，数据库基础试题所占的比重一般，基本上每年一道题。从历年试题统计(见表 2-8)来看，主要考查与关系数据库有关的知识、关系模式、关系模式的范式、关系代数运算等是反复考查的内容，总的来说，难度一般都不大。

试题 1 (2000 年试题 7)

从供选择的答案中，选出应填入下面叙述中{ }内的最确切的解答，把相应编号写在答卷的对应栏内。

域表达式{ab|R(ab)∧ R(ba)}转换为等价的关系代数表达式,所列出的式子中 A 是正确的.SQL 中集合成员资格的比较操作"元组 NOT IN(集合)"中的"NOT IN"与 B 操作符等价。SQL 中涉及属性 AGE 是否是空值的比较操作，写法 C 是错误的。类似于工资在 800 至 5000 之间"这种约束，是属于 DBS 的 D 功能。

设关系模式 Q 是 3NF 模式，那么，E 这种提法是不正确的。

供选择的答案

A: ①Л1,2(б$_{1=4\wedge 2=3}$(R×R)) ②Л1,2(б1=4($R \underset{2=3}{\triangleright\!\triangleleft} R$))

③ Л1,2(б1=4($R \underset{2=1}{\triangleright\!\triangleleft} R$)) ④ R∩Л1,2(R)

B：①＜＞SOME ②=SOME ③＜＞ALL ④=ALL

C：①AGE IS NULL ②NOT(AGE IS NULL)

③AGE=NULL ④AGE IS NOT NULL

D：①完整性 ②并发控制 ③安全性 ④恢复

E：①Q 一定是 2NF 模式 ②Q 可能不是 4NF 模式

③Q 可能不是 BCNF ④Q 一定不是 BCNF

【解析】
问题 A 涉及域表达式与关系代数表达式的转换。域表达式{ab|R(ab) ∧ R(ab)}表示取出二元关系 R 中有对称关系的二元组的集合，即(a，b)∈R，(b，a) ∈R，如果用 D 来表示该域表达式，则有 D⊆R。答案中只有②是正确的。

$R \underset{2=3}{\triangleright\triangleleft} R$是关系 R 与其自身的条件连接，条件是第 1 个 R 的第 2 个元素与第 2 个 R 的第 1 个元素相等。举例说明如下：
R={(1，2)，(2，3)，(3，2)，(2，1)(3，4)}

则$R \underset{2=3}{\triangleright\triangleleft} R$ 的结果是

{(1，2，2，3)，
(1，2，2，1)
(2，1，1，2)
(2，3，3，2)
(3，2，2，3)
(2，3，3，4)}；
再作δ选择运算找出 1，4 列相等的元组{(1，2，2，1)，(2，1，1，2)，(2，3，3，2)，(3，2，2，3)}；
再作π投影运算取 1，2 列，得到{(1，2)，(2，1)，(2，3)，(3，2)}。
问题 B 涉及集合成员资格比较。SQL 中的集合比较有 4 种：集合成员资格比较、集合成员算术比较、空关系测试和重复元组的测试。其中集合成员资格比较有两种形式：
(集合 1)IN(集合 2)
(集合 1)NOT IN(集合 2)
这里"IN"与算术比较中的"=SOME"等介，"NOT IN"与算术比较中的"＜＞ALL"等价。
问题 C 涉及空值的处理。SQL 中允许属性值为空值，用关键字 NULL 表示空值。测试某属性值为空值，用"(某属性)IS NULL"表示，测试某属性值为非空值，用"(某属性)IS NOT NNULL 或"NOT((某属性)IS NULL)"来表示。
问题 D 涉及属性值的约束，属于数据库完整性范畴。
问题 E 涉及关系模式的范式。根据 4 种范式之间的关系：BCNF⊂3NF⊂2NF⊂1NF，可以判断 3NF 模式必定是 2NF 模式，BCNF 必定是 3NF 模式，但是 3NF 可能是 BCNF 模式，也可能不是 BCNF 模式。
【答案】A：② B：③ C：③ D：④

试题 2（1999 年试题 7）

从供选择的答案中，选出应填入下面叙述中{ }内的最确切的解答，把相应编号写在答卷的对应栏内。
最常用的一种基本数据模型是关系数据模型，它用统一的 A 结构来表示实体及实体之间的联系。关系数据库的数据操作语文(DML)主要包括 B 两类操作。

关系运算以关系代数为理论基础，关系代数的最基本操作是并、差、笛卡尔积和 C。用 R$\triangleright\triangleleft$ S 表示关系 R 和关系 S 的 D 。
设关系 R 和关系 S 图示如下：
R：A B C　　S：B C D　　T：A B C D

a b c	b c d	a b c d
b b f	b c e	a b c e
c a d	a d b	c a d b
d a d		d a d b

则关系 T 是关系 R 和关系 S E 的结果。

供选择的答案

A：①树 ②网络 ③图 ④二维表

B：①插入和删除 ②检索和更新 ③查询和编辑 ④统计和修改

C：①投影、联接 ②联接、选择 ③选择、投影 ④交、选择

D：①联接 ②笛卡尔积 ③ 联接 ④自然联接

E：①自然联接 ② 联接 ③笛卡尔积 ④并

【解析】

用二维表格结构表示实体类型、关键码表示实体间联系的数据模型称为关系模型。关系数据库的 DMLA 语言分成查询语句和非查询语句，前者描述用户要进行的各种检索操作，后者描用户要进行的有关数据库更新的操作。

关系运算以关系代数为理论基础，关系代数的最基本操作是并、差、笛卡儿积、选择和投影。设有两个关系 R 和 S，则 R 和 S 的并是由属于 R 或属于 S 的元组组成的集合，记为 RUS。R 和 S 的交是由既属于 R 又属于 S 的元组组成的集合，记为 R∩S。R 和 S 的差是由属于 R 但不属于 S 的元组组成的集合，记为 R-S。设关系 R 和 S 的元数分别为 r、s，则 R 和 S 的笛卡儿积是一个(r+s)元的元组集合，每个元组的前 r 个分量来自 R 的一个元组，后 s 个分量来自 S 的一个元组，记为 R×S。投影操作是对关系进行垂直分割，消去关系中某些列，并重新按排列的次序，再删除重复的元组。选择操作是根据某些条件对关系做水平分割，选择符合条件的元组。

关系 R 和 S 的自然联接用 R⋈S 来表示，其计算过程如下：

设 R 和 S 的公共属性为 A1……Ak，挑选 R×S 中满足 R·A1=S·A1…R·Ak=S·Ak 的元组；去掉 SA1…SAk 列。自然联接是构造新关系的有效方法，是关系代数中常用的一种运算。从试题中 E 的计算结果看，这里应该是经过自然联接得到的结果。

【答案】A：④ B：② C：③ D：④ E：①

试题 3（1988 年试题 2）

从供选择的答案中选出应填入下面叙述中{ }内的最确切的解答，把相应编号写在答卷的对应栏内。

实体联系模型(简称 ER 模型)中的基本语义单位是实体和联系。ER 模型的图形表示称为 ER 图。联系可以同 A 实体有关。实体与实体之间的联系可以是 B 。

利用 ER 模型进行数据库的概念设计，可以分成 3 步：首先设计局部 ER，然后把各个局部 ER 模型综合成一个全局的模型，最后对全局 ER 模型进行 C ，得到最终的 ER 模型。

ER 模型向关系模式的转换规则是把一个实体类型转换成一个关系模式，实体的属性是关系的属性，实体的键是关系的键。把一个联系类型转换成一个 D ，参与该联系类型的各实体的键以及联系的属性转换成 E ，其中的键由实体与实体之间的联系决定。

供选择的答案

A：①0 个 ②1 个或多个 ③1 个 ④多个

B：①一对一和一对多 ②一对一和多对多

③一对多和多对多 ④一对一、一对多和多对多

C：①简化 ②结构化 ③最小化 ④优化
D：①联系模式 ②数据模式 ③关系模式 ④逻辑模式
E：①联系属性 ②关系的属性 ③数据属性 ④关系的候选键
【解析】
实体联系模型中，联系指的是实体之间的联系，实体之间的联系有一对一、一对多和多对多 3 种。如：一个公司有一个经理，而每个经理只有一个公司任职，则公司与经理之间是一对一联系；一个公司有多个副经理，而每个副经理只在一个公司任职，则公司与副经理之间是一对多联系；一个老师可以有多个学生，而一个学生也可以有多个老师，则老师与学生之间是多对多的联系。联系可以同一个或多个实体有关。
利用 ER 模型进行数据库的概念设计，分为 3 步：第 1 步设计局部的 ER 模型；第 2 步综合各个局部 ER 模型成为全局的 ER 模型；第 3 步对全局的 ER 模型进行优化，得到最终的 ER 模型。
ER 模型向关系模式的转换依据如下：把一个实体类型转移为一个关系模式，相应实体的属性转换为对应的关系的属性，实体的键是关系的键；把一个联系类型转换为一个关系模式，所有与该联系相关的实体的键及联系的属性转换成关系的属性，关系模式的键由实体与实体之间的联系所决定。
【答案】A：② B：④ C：④ D：③ E：②

试题 4（1997 年试题 5）

从以下叙述中选出 5 条最确切的叙述，把相应编号依次写在答卷的 A～E 栏内。
①在数据库系统中，数据独立性指数据之间的相互独立，互不依赖。
②SQL 语言的视图定义和视图操作功能不支持逻辑数据的独立性。
③SQL 语言中不提供显式地使用索引的功能，支持了物理数据的独立性。
④用户对"脏数据"的读出是由于数据库完整性规则受到了破坏。
⑤在数据库系统中，数据的安全性是指保护数据以防止未被授权用户的蓄意或者无意使用。
⑥实体完整性规则指主关键字值的任何组成部分都不可以是空值；引用完整性规则则不允许引用不存在的实体(即元组)
⑦在数据库系统中，数据的完整性是指数据的正确性和有效性。
⑧"授权"是数据库系统中采用的完整性措施之一。
⑨事务处理(Transaction)是数据库运行的基本单位。如果一个事物处理成功，则全部数据得到更新和提交；如果失败，则已做的全部更新被恢复成原状，好像整个事务处理示进行过一样。这样使数据库保持了一致性。
⑩对数据库的查找、增添、删除和修改等操作都需由数据库管理员进行完整性定义和安全性授权，由数据库系统具体执行。
【解析】
该题主要查考生对数据库基本概念的掌握情况，下面从基本概念入手进行逐条解析。
(1)数据的独立性分为物理独立性和逻辑独立性。物理独立性指当数据库的物理布局和物理组织形式改变时，不影响数据库的全局逻辑结构的性质；逻辑独立性是指当数据库的全局逻辑结构改变时，不影响某些局部逻辑结构的性质，因而题中①不正确。视图是数据库中满足一定条件约束的数据组成的虚拟关系，视图可作为某用户的专用数据部分，这样既提高了数据的独立性，又利于数据的安全保密，所以题中②认为视图定义产不支持逻辑数据的独立性是不正确的，而叙述③是正确的。
(2)数据的完整性是指保证数据的正确性和有效性，因而叙述⑦是正确的。可以采取多种方法来保证数据的完整性。实体完整性规则指主关键字值的任何组成部分都不可以是空值；引

用完整性规则则不允许引用不存在的实体(即元组)，叙述⑥是正确的。
(3)并发控制是为了防止由于多个用户并行地对数据操作时，他们之间会相互干扰从而导致数据库数据不一致，为此需对并发操作采取控制措施。用户对"脏数据"的读出是因为并发控制没做好，所以叙述处理④不正确。事务处理(Transaction)是数据库运行的基本单位，是数据库系统中保证数据一致必的手段。如果一个事务处理成功，则全部数据得到更新和提交；如果失败，则已做的全部更新被恢复成原状，好像整个事务处理未进行过一样。这样使数据库保持了一致性。叙述⑨正确。
(4)在数据库系统中，数据的安全性是指保护数据以防止未被授权用户的蓄意或者无意使用。"授权"是数据库进行安全保护的措施之一(即允许某一用户以某种方式访问某些数据)，而不是完整性措施，所以叙述⑧不正确，叙述⑤正确。
(5)对数据库的查找、增添、删除、修改等操作都需由数据库管理员进行完整性定义和安全性授权，由数据库系统具体执行。除数据库管理员以外，设计、建立和管理维护数据库的软件人员可参与数据库系统的分析和设计，在数据库运行期间对用户的使用、存取进行监控并统计数据库使用情况，在必要时整理并重新构造数据库或恢复数据库。叙述⑩不够准确。
【答案】A：③ B：⑤ C：⑥ D：⑦ E：⑨

试题 5（1996 年试题 3）

从供选择的答案中，选出应填入下面叙述中{ }内的最确切的解答，把相应编号写在答卷的对应栏内。
设有一图书管理数据库，其关系模式是 R0(L#，B#，BNAME，BPRICE，BPUB)，其属性分别表示个人借书证号、书号、书名、书价、图书出版社。该关系模式 A 。它的主要问题是数据冗余。如把 R0 分解成两个关系模式 R1 B 和 R2 C ，则可以部分地解决这一问题。R1 和 R2 是规范化程度较差的范式 D 。另外一种分解方法可以得到 3 个模式 R3(L#，B#)，R4(B#，BNAME)、R5(BNAME，BPRICE，BPUB)，则 R3、R4 和 R5 都 E 。
供选择的答案
A、D、E：①属于第一范式但不属于第二范式 ②属于第二范式但不属于第三范式
③属于第三范式 ④不是范式
⑤属于第二范式但不属于第一范式 ⑥属于第三范式但不属于第二范式
B、C：①(L#，B#，BPRICE) ②(L#、B#)
③(B#，BNAME) ④(B#，BNAME，BPRICE，BPUB)
⑤(BNAME，BPRICE，BPUB) ⑥(L#，BNAME，BPRICE)
【解析】
本题与 1993 年试题 2 类似。
假定借书证号与书号是主码，可以唯一决定一条记录。由关系模式 R0 可分解成 3 个模式 R3、R4，R5，而得到第三范式，由书名可确定书价与图书出版社，而书号决定书名。
第一范式要求每个属性都是不可分解的，题中关系模式 R0 的所有的属性都不可再分，因而属于第一范式，但它不是第二范式，因为第二范式要求每个非主属性的完全函数依赖于主码，而 R0 中如果将(L#，B#)作为主码的话，则属性 BNAME 只是部分函数依赖于主码(因为它的完全函数依赖于 B#)，所以它不是第二范式。
如果有多人借同一种书，则 BNAME 等信息要重复出现多次。如果将 R0 分解成 R1(L#，B#)和 R2(B#，BNAME，BPRICE，BPUB)两个关系模式，则 R1 和 R2 都成为第二范式，但在 R2 中存在关系 B#→BNAME，BNAME/B#，BNAME→BPUB，因而非主属性 BPRICE B#)，R4(B#，BNAME)和 R5(BNAME，8PRICE，BPUB)，这样每一关系模式都是第三范式，在最大程度上降低了数据冗余。

【答案】A：① B：② C：④ D：② E：③

试题 6（1995 试题 2）

从供选择的答案中，选出应填入下面叙述中{ }内的正确答案，把编号写在答卷的对应栏内。传统 的数据库基本上是由 A 组成的。B 在技术和理论上已经成熟，成为当前商用数据库的主流。C 技术是 80 年低中期引入的。目前，多媒体数据库基本上靠与关系模式相结合的 D 来支持。但当数据量大，数据结构复杂时，靠 D 很难适应。当前，在 DBMS 的研究方面，较活跃的是 E 。

供选择的答案：

A、D：①图形 ②文件 ③元组 ④文件系统

⑤对象 ⑥过程

B：①关系数据库 ②网状数据库 ③层次数据库 ④空间数据库

C：①关系数据库 ②网状数据库 ③层次数据库 ④面向对象数据库

E：①网状数据库 ②层次数据库 ③DBASE ④演绎数据库

【解析】

数据是描述事物的符号记录。数据库是存放数据的仓库，是长期储存在计算机内、有组织的、可共享的数据集合。传统的数据基本上是由大量记录所组成，记录又称为元组。

在数据库中是用数据模型对现实世界进行抽象的，现有的数据库系统均是基于某种数据模型的，数据模型可以对数据进行抽象表示，是我们进行数据模型化的工具。数据库领域中最常用的数据模型有 3 种，它们是层次模型、网状模型和关系模型。其中层次模型和网状模型统称为非关系模型。

数据库按数据模型来分，可分为层次型数据库、网状数据库和关系数据库 3 大类，层次和网状数据库又称为非关系数据库。这 3 类数据库都是在 60 年代末发展起来的。关系数据库由于采用二维表的形式来描述实体和实体间的联系，其数据结构相对简单，对于用户来讲浅显易懂，技术理论上也已经成熟，已成为当前常用数据库的主流。

随着数据库应用领域的不断发展、扩充，传统的数据库已不能完全适应新领域应用的需要。这些领域包括计算机辅助设计(CAD)，计算机辅助基础(CAM)、VISI 设计、计算机辅助软件工程(CASE)、图像处理和多媒体技术等方面。除了对传统数据库的需求外，这些领域还要求数据库能够处理更复杂的结构、递归定义的对象和更大的数据量等，具有更多更复杂的数据库随着这种需求而出现，空间数据库、面向对象数据库、多媒体数据库和演绎数据库等就是其中发展起来的一部分。

空间数据库是随着地理信息系统发展起来的一种数据库。空间数据库的空间数据是用来表示空间物体的位置、形态、大小以及分布特征等信息的数据。空间数据不仅包含物体本身的空间位置和位置信息，还包含物体空间即拓扑关系的信息。空间数据库常用的数据结构有矢量和栅格结构两种。空间数据库研究的主要内容有空间数据表示、数据结构与数据模型空间、数据库管理系统等。空间数据库的研究正在深入系统地进行，但在理论与技术上还不如关系数据库那么成熟。空间数据库的应用领域主要用于地理信息系统，它是地理信息系统的核心。

面向对象数据库是从 80 年代中期开始出现并发展的，它是关系数据库系统和面向对象的方法相结合组织起来的，这种新一代的数据库系统称为面向对象数据库系统(ODBC)。面向对象数据库是符合面向对象数据模型的，其数据结构是建立在对象和类的概念上的，数据具有封装性、继承性和多态性等特点。当前面向对象数据库系统的研究工作主要集虽在面向对象数据模型的形式化与面向对象的程序语言的结合等方面。

多媒体数据系统是一种由文本、图像、声频和和视频等多种形式的数据组织起来的集成数据库系统。目前，相当一部分多媒体数据库是通过关系模式与文件系统相结合来支持的，但当

数据量大、数据结构复杂时，靠文件系统是很难适应的。分布式多媒体数据库系统能对对象进行存储、检索以及支持对象间的通信。这些对象是由图像、文本、声频和视频等数据类型混合而成的。多媒体数据库的多媒体数据模型、系统实现等问题目前正在深入研究。多媒体数据库的发展也与存储介质的发展、输入输出介质的发展、数据模型理论的发展、通信介质的发展等密切相关。多媒体数据库系统改善了用户的工作环境，提高了用户对数据的可利用性。

演绎数据库的数据库模型是一种基于逻辑的模型。Horn 子句的逻辑为演绎数据库提供了递归定义的能力，从而可以定义更复杂的数据，支持更强的数据操作能力以及提供更完善的完整性保护，并提供数据操作与宿主语言统一的说明性语言。因此，演绎数据库具有比传统数据库更强的能力。演绎数据库是数据库技术与逻辑程序和人工智能相结合的结果。目前，相对于传统数据库的研究而言，演绎数据库的研究更为活跃。

【答案】A：③ B：① C：④ D：④ E：④

试题 7（1995 试题 16）

从下列与数据库中数据的独立性、完整性和安全性有关的叙述中，选出 5 条正确叙述，并按编号从小到大的次序写在答卷的 A～E 栏内。

①在数据库系统中，数据独立性指的是数据之间相互独立，互不依赖。

②数据库系统中，由于有封锁机制，所以应用程序对数据的存储结构和存取方法有较高的独立性。

③SQL 语言的视图定义和视图操作功能在一定程度上支持了逻辑数据独立性。

④SQL 语言中不显式提供索引功能，这是对物理数据独立性的支持。

⑤在数据库系统中，数据的完整性是指数据的正确性和相容性。

⑥"授权"是数据库系统中采用的完整性措施之一。

⑦实体完整性和参照完整性是可应用于所有关系数据库的两条完整性准则。

⑧"脏"数据的读出是数据库安全性遭到破坏的一个例子。

⑨在数据库系统中，数据的安全性是指保护数据以防止不合法的使用。

⑩SQL 语言在 COMMIT 语句、ROLLBACK 语句和 LOCK TABLE 语句都具有维护数据库安全的功能。

【解析】

数据库系统中，数据独立性是一个重要概念。数据的物理存储结构和数据在计算机内表示的逻辑结构千差万别，都有多种不同形式。如果用户不必考虑数据在计算机中的表示及其在计算机中的物理存储结构，而是按自己所需的逻辑结构对数据进行操作，就大大地方便了用户；另一方面，若数据在计算机内的逻辑结构是独立于物理存储结构的，即计算机对数据进行操作时，物理结构的变化不会影响数据整体的逻辑结构，这就大大提高了机器的效率。

上述两个方面体现了两层独立性的概念：逻辑数据的独立性(即用户的数据逻辑结构独立于计算机内数据表示的逻辑结构)，物理数据的独立性(即计算机机内的数据表示的逻辑结构独立于物理存储结构)。为了实现这两层数据独立性，把数据库的结构分为外部级(单个用户视图级)、概念级(全局视图)和内部级(存储视图)三级。用户通过数据操作语言进行的操作是对外部级的操作；而外部级数据模式与概念级数据模式的转换、概念级数据模式与内部级数据模式的转换均由数据库管理系统实现。显然，SQL 语言的视图定义和操作功能在一定程度上支持了逻辑数据的独立性；SQL 不显式提供索引同是对物理数据独立性的支持。

本题③、④两条叙述是正确的，第①条叙述是错误的。数据的独立性并不是指数据之间互不依赖，事实上许多数据之间是有依赖关系的。第②条叙述也是错误的，因为封锁机制是解决并发操作问题的，而不是解决数据的结构问题的。

数据库系统中，数据的完整性是指数据的正确性和相容性，数据库管理系统提供完整性保护的功能，系统提供定义完整性约束条件的功能和检查完整性约束条件的方法。系统的完整性子系统就是根据完整性约束条件工作的。完整性约束条件包括对数据值和结构的约束，还包括对数据在操作前后应满足的约束等。实体完整性和参照完整性可适用于关系数据库。
因此本题中⑤、⑦两条叙述是正确的，而第⑥条是错误的。因为"授权"是授于用户对数据库中何种数据文件作何种操作的权力，而为了防止非法使用数据库，是数据库系统中采用的安全性措施，而不是完整性措施。数据库系统中，数据的安全性是指保护数据以防止不合法使用，避免数据泄露或遭到破坏。它通常是通过对用户标识、鉴别及对存取进行控制来实现的，"授权"就是保证安全性的一种措施，因此，本题第⑨条叙述是正确的，第⑧条叙述是错误的。因为"脏"数据一般是指由于并发控制不当，由操作异常所形成的数据，而不是不合法操作所形成的数据。第⑩条也是错的。因为 SQL 语言 LOCK TABLE 是用于并发控制的，而不是用于安全性控制的。
[答案]A：③ B：④　C：⑤ D：⑦ E：⑨

试题 8（1994 年试题 1）

从供选择的答案中，选出应填入{ }内的正确答案，把编号写在答卷的对应栏内。

在数据库理论中，关系 R 和 S 在第 i A 和 B 上的 θ-联结(Jion)写成 $R\underset{i\theta j}{|\times|}S$，其中 θ 是 C 。

若 R 是 r 元关系，则有 $R\underset{i\theta j}{|\times|}S$ =D，D 中的运算符×为 E 乘积符。

供选择的答案

A、B：①行 ②列 ③个记录 ④张表

C：①算术运算符，如+、- ②逻辑运算符，如 ∨、∧

③算术比较运算符，如=、＜ ④集合运算符，如 ∩、∪

D:①$\delta_{(I+j)\theta r}(R\times S)$ ②$\delta_{(I+j)\theta j}(R\times S)$ ③$\delta_{i\theta(r+j)}(R\times S)$ ④ $\delta_{i\theta j}(R\times S)$

E：①算术 ②笛卡尔 ③矢量 ④逻辑

【解析】

关系代数是由一组以关系为运算对象的特定运算组成的，通过这组运算，对一个或多个关系进行分解和组合，构造出新的关系，从而得到所需要的数据。关系代数的运算可分为两类，一类是传统的集合运算，即并、差、交和笛卡尔积；另一类是特殊的集合运算，即投影、选择、联接、自然连接和除法。

关系代数是关系数据库的数学基础，是离散数学的重要组成部分，关系代数结合数据库，为关系数据库的全面开发奠定了基础，关系代数语言是关系型数据库的数据操作语言(DML)中重要的一类。

联接(Join)是关系代数运算中的一种，记号为 $R\underset{i\theta j}{|\times|}S$，它是从两个关系 R 和 S 的笛卡尔积中选取属性间满足一定条件的元组。其中 i 和 j 分别是关系 R 的第 i 列分量和关系 S 中的第 j 列分量。 为算臬比较运算符，它包括"＜"、"＞"、"="等运算符；当 为"="时，上式称为等值联接。若有 $R\underset{i=j}{|\times|}S$ 即表示 R 与 S 的笛卡尔积中的第 r+j 列，实际上就是 S 中的第 j 列。

因此又有 $R\underset{i\theta j}{|\times|}S$ =$\delta_{i\ (r+j)}(R\times S)$，其中 δ 表示投影运算，×表示笛卡尔积。

【答案】A：② B：② C：③ D：③ E：②

试题 9（1994 年试题 3）

从供选择的答案中，选出应填入{ }内的正确答案，把编号写在答卷的对应栏内。
数据库是存储在一起的相关数据的集合，能为各种用户所共享，且 A 。在关系数据库中，若关系模式中的每个关系的属性值均是不可分解的，则该关系模式属于 B 。
关系代数运算是以 C 为基础的运算，其 5 种基本运算是并、差、D 、投影和选择；规范理论研究中，分解 E 主要是消除 E 中多余的数据相关性。
供选择的答案
A：①消除了数据冗余 ②降低了数据冗余度
③具有不相容性 ④由用户控制物理数据的存取方式
B：①1NF ②2NF ③3NF ④BCNF
C：①代数运算 ②关系运算 ③谓词演算 ④集合操作
D：①交 ②连接 ③笛卡尔积 ④自然连接
E：①内模式 ②视图 ③外模式 ④关系模式
【解析】
数据库是存储在一起的相关数据的集合，它能为各种用户所共享，并具有最小冗余度，数据间联系密切而又有较高的数据与程序的独立性。关系数据库是以关系型数据模型为基础的数据库，关系模型就是通过表格结构来表示实体类型及实体间联系的模型。
关系模型中将表格结构中的记录类型为关系模式，将表格中的字段称为属性，字段值称为属性值，将记录称为元组。关系模型中，一个数据库模式是一个关系模式的集合。对同一问题，可以选用不同关系模式集合作为数据库模式，但其性能的优劣大不相同，为了区分优劣，把数据库模式分为各种不同等级的范式。
第一范式(1NF)：关系模式中每个关系的属性值均不再可分。
第二范式(2NF)：若某个关系是第一范式，且每个非主属性完全函数依赖于各关键字。
第三范式(3NF)：若某个关系是第二范式，且每个非主属性不传递依赖于任何关键字。
BC 范式（BCNF）：若某个关系是第一范式，且每个属性均不传递依赖于任何关键字。
可知这些范式之间的关系为：BCNF>3NE>C:2NE>C>1NE.
在数据库设计中，分解的关系模式使其性能优化，实质上，分解关系模式主要是消除关系模式中多余的数据相关性，降低数据的冗余度是设计的主要目标之一，但实际上不可能完全消除数据的冗余，否则会出现操作上的问题。
关系数据库的操作语言分成关系代数语言主导关系演算语言两类，关系代数语言中的运算是以集合为基础的运算，把关系看成元组的集合来处理，其基本运算有并、差、笛卡尔积、投影和选择五种。
[答案]A：② B：① C：④ D：③ E：④

试题 10 （1993 年试题 1）

从下列有关数据库的叙述中，选出 5 条正确的叙述，并把编号按从小到大的次序写在答卷的对应栏内。
① 关系代数的最基本操作有并、差、笛卡尔乘积、选择和投影。
② 视图是用户看到的数据库。它由一个或多个基本表导出。其定义存在于数据库目寻中；其数据在物理上以表的形式直接存储。因此对视图就像对基本表一样能进行查、添、删、改等操作。
③ 一般的完整性规则有两个 :实体完整性规则是指主关键字值的任何组成部分都不是空值；引用完整性规则是如果关系的 R 的属性 A 为外关键字（设为关系 S 的主关键字），则 A 的每个值是 S 的主关键字的某一值或是空值。
④ 对于查、添、删、改操作都需由数据库管理员 DBA 进行完整性定义和安全性授权，由

数据库系统具体执行。
⑤ 多用户的数据库系统的目标之一使它的每个用户好像面对着一个单用户的数据库一样使用它，为此数据库管理系统必须进行并发控制。
⑥ 数据库系统目录（或称数据库字典）也由一些关系组成，所以用户同样可以对其进行查、添、删、改操作。
⑦ 在 SQL 的查询语句中，要对所查询的数据指明存取路径，进行导航，数据库管理系统依此执行代数优化和非代数优化，这样才能有效地访问数据。
⑧ 嵌入式的数据库语言构成的应用程序环境包括主语言（如程序设计 C 和 Fortran 和数据子语言（如 SQL），前者能处理记录和域，后者只能处理表，游标机制起着两种语言的桥梁作用。
⑨ 事务（Transaction）是数据库运行的基本工作单位。如果一个事务执行成功，则全部更新提交；如果一个事务执行失败，则已做过的全部列新被恢复原状，好像整个事务从未有过这些更新。这样就保持了数据库处于一致性状态。
⑩ 既然数据库能实现对不同用户数据共享，所以数据库中数据不应该存在任何冗余。
[解析]
①正确。并、差、笛卡尔积、选择和投影这 5 种运算是基本的关系代数运算，是关系代数运算的最小完备集。
②错误。视图是一种虚拟关系，由数据库中的某一个或几个基本关系通过关系运算导出。视图燕不是以实际的数据库，在数据库中只存储该视图的定义，即从有关的基关系导出该视图的方法。
当用户对数据库中的数据进行存取时，数据库管理系统自动地将其转换为对相应的基关系的存取，然后再执行变换的命令。在关系数据库管理系统中，提供用户使用数据库的不同要求，允许不同的用户从不同的角度看同一个数据库，这是能过视图来实现的。
③正确。在关系数据模型中，一般有实体完整性规则和关联（引用）完整性规则。实体完整性规则是指在任何关系的任何一个元组中，主关键字值的任一分量都不允许为空值。关联完整性规则是指如果某一关系 R 中的一个属性 A 相对于另一关系 S 为外关键字，则 A 的值必须要么是空值，要么等于 S 中某一个关键字值。
④错误。数据完整性定义不一定需要由数据管理员进行。
⑤正确。数据库技术的最大优点之一是数据的共享性，但同时需解决并发控制问题。比如当某一个应用程序正在对某个记录进行修改的过程中，另一程序恰好要读取该记录，，这时读得的数据就可能是不正确的，为了避免这种情况的发生，就必须对并发操作施加某些控制措施，如记录加锁。
⑥错误。系统的全部工作领带于数据字典中数据的正确性，因此不能允许用户随意定义和操纵数据字典中的表。
⑦错误。SQL 是一种过程性语言。对于非过程性语言，由系统来分析该语句应该干什么，然后选择最佳方案，实现该语句。
⑧正确。当查询语句产生多元组结果关系时，为了能把该组元组逐个地提交给宿主语语句处理，数据库管理系统一般都提供了游标概念，包括游标关系和游标指针。
游标机制把结果关系看作一种特殊视图，称为游标关系，用游标指针指明当前所处理元组的位置，通过用游标操作语句改变游标指针，从而逐个地从游标关系中取出元组，赋给宿主语言程序变量，进行信息交流。
⑨正确。为便于维护数据的完整性，数据库管理系统把需要进行多步数据操作才能完成的一项业务称作事务。数据库管理系统规定，事务是数据库中独立执行的最小程序单位。一个事务中的各个数据操作不是独立存在的，它们共同组成一个有机整体，或者全部执行，或者全

部不执行。如果某个事务的某一中间步骤发生了错误，则数据库管理系统将自动恢复数据库原始状态，以保证数据的完整性。
⑩错误。在数据库中必要的数据冗余是必须的。
[答案]A：① B：③ C：⑤ D：⑧E：⑨

试题 11 （1993 年试题 2）

从供选择的答案中，选出应填入下面关于关系数据库叙述中{ }内的正确答案，把编号写在答卷的对应栏内。
设有关系模式 W（C，P，S，G，T，R），其中各属性的含义是：C--课程，P--教师，S--学生，G--成绩，T--时间，R--教室，根据语义有如下数据依赖集：
D={C→P，（S，C）→G，（T，R）→C，（T，P）→R，（T，S）→R}
关系模式 W 的一个码（关键字）是 A，W 的规范化程度最高达到 B。若将关系模式 W 分解为 3 个关系模式 W1（C，P)、W2（S，C，G)、W3（S，T，R，C)。则 W1 的规范化程度最高达到 C，W2 的规范化程序最高达到 D，W3 的规范化程序最高达到 E。
供选择答案
A：①（S，C）②（T，R）③（T，P）④（T，S）
B ～E:①1NE ②2NE ③3NE ④BCNE ⑤4NE
[解析]

设 R（W）是一个关系模式，X，Y$\subseteq$W。关系模式 R 上的函数依赖是形式 f:X→Y 的一个命题。其含义是为对于 R 的任意一个可能的实例 r,如果对任意 t2,t2r，t2[x]=t2[x]，则必有 t1=[y]=t2[y].X$\subseteq$W 是 R 的一个关键字，如果存在函数依赖 X→W，且不存在 X$\subseteq$W，使得 X→W 也成立。
由以上定义可知，（T，S）是 W（C，P，S，G，T，R）的一个关键字。如果 R 中的每一个属性的值域中的每一个值都是不可分解的，则称 R 属于第一范式（1NE)：如果 R 是第一范式，并且 R 中任何一个非主属性都完全函数依赖于 R 的每一个候选关键字，则称 R 是第二范式（2NE)；如果 R 是第一范式，并且 R 中不存在任何非主属性传递函数依赖于 R 的某个候选关键字，则称 R 是第三范式（3NE)；如果 R 是第一范式，并且 R 中不存在任何属性传递函数依赖于 R 的任何一个关键字，则 R 属于 Boyce-Codd 范式（BCNF)；如果对于在 R 上成立的每一个非平凡多值函数依赖 X→→Y，X 都是 R 的一个超关键字，则 R 属于第四范式（4NE)。
由以上定义可知，W1 的规范化程度最高达到 4NE，W2 的规范化程序最高达到 4NE，W3 的规范化程序最高达到 2NE 。
[答案]A：④ B：② C：⑤ D：⑤ E：②

试题 12 （1991 年试题 4）

从供选择的答案中选出应填入下列叙述中□的正确答案，把编号写在答案的对应栏内。
一个数据库系统必须能表示实体和关系。关系可与 A 实体有关。实体与实体之的关系有一对一、一对多对多三种，其中 B 不能描述多对多的联系。
一般地，一个数据库系统 C 外视图 D 概念视图，E 数据子语言。
供选择的答案
A：①0 个 ②1 个 ③2 个 ④3 个或 3 个以上 ⑤1 个或 1 个以上 ⑥0 个或 0 个以上
B：①网状模型 ②层次模型 ③关系模型 ④网状模型和层次模型 ⑤层次模型和关系模型

⑥网状模型和关系模型 ⑦网状模型、层次模型和关系模型
C～E ①只能有一种 ②最多只能有两种 ③至少有两种 ④可以有多种
[解析]
实体是一个范围广泛的概念，从具体的人、物、事件到抽象的状态以及概念，都可以用实体来抽象表示。实体是存在于现实世界中，并且可以根据其自身信息加鉴别的任何事物的抽象表示。联系是现实世界内部或者事物之间语义关系的抽象定义。联系可以同一个或者多个实体有关。实体之间的联系可以是一对一（夫妻关系）、一对多（师生关系）或者多对多的（学生和课程关系）。实体和联系组成模型可以用 E-R 图表示。
层次模型、网状模型和关系模型是数据库的 3 种主要数据库模型。层次模型一种可以用树形表示的分层模型，它有两个特征：一是有且仅有一个无前驱的结点（即树根）；二是共他结点仅有一个前驱。所以，层次模型不能够描述多对多的联系。网状模型是有向图模型。关系模型以集合论为基础，关系是实体的元组集合。
关系数据库中，允许不同的用户从不同角度查看同一个数据库，这是视图来实现的。视图是一种中以用数据库内的一个或多个基关系，通过关系运算导出虚拟关系的方式。用户所看到的是外视图，外视图并不实际地存储在数据库内。外视图所反映的数据库都要通过与概念视的映射，变换到概念数据中。因此，一个数据库可以提供多个不同的外视图，但只存在一个概念视图同物理数据库一一对应。一个数据库系统，由于它的每个模式都有其独立的数据子语言，所以一个数据库可以支持多个数据子语言。
[答案]A：⑤ B：② C：④ D：① E：④

试题 13 （1990 年试题 3）

从下列关于数据库系统特点的叙述中，选出 5 条正确的叙述，把编号依次写在答卷对应栏内。
① 数据库避免了一切数据重复。
② 数据库减少了数据冗余。
③ 各类用户程序均可随意地使用数据库中的各种数据。
④ 用户程序按所对应的子模式使用数据库中的数据。
⑤ 数据库数据可经 DBA 认可的各用户所共享。
⑥ 数据库系统中如概念模式有所改变，则需将与其有关的子模式做相应改变，否则用户程序需改写。
⑦ 数据库系统中概念模式如有改变，子模式不必变，因而用户程序亦不必改写。
⑧ 数据库系统的存储模式如有改变，则概念模式应予调整，否则用户程序会在执行中出错。
⑨ 数据库系统的存储模式如有所改变，概念模式无需改动。
⑩ 数据一致性是指数据库中数据类型一致。
[解析]
①错误。数据库中最在限度地减少数据冗余，但不能避免一切数据重复，有时为了提高效率还会增加重复数据。
②正确。数据库能减少数据冗余。
③错误。数据库系统的安全性措施使每个用户的数据得到合理的保护，防止任何未经许可的其他用户对数据的访问和修改。
④正确。数据库的逻辑结构可分为子模式、概念模式和存储模式三级，其中子模式是用户看到和使用的数据描述，是用户与数据库的接口，常称作用户对数据库的视图。用户使用 DML 对数据库进行操作，实际是对子模式记录进行操作，用户程序按子模式使用数据库。概念模式是数据库中全部数据整体逻辑结构描述。存储模式是数据库中的物理存储结构的描述.存储模式是数据库的物理存储结构的描述。概念模和存储模式对用户是透明的。

⑤正确。数据共享是数据的目的之一，但为了数据的安全性，用户共享数据必须得到 DBA（即数据库管理员）的授权。
⑥错误。在数据库的三级模式结构中，概念模式的改变不影响子模式，也不会影响以子模式方式使用数据库的用户程序，子模式相对于概念模式是独立的。概念模式的改变需改变子模式与概念模式之间的映射关系。
⑦正确。理由同⑥。
⑧错误。如同⑥所述，概念模式相对于存储模式也是独立的，存储模式的改变也不会影响概念模式，而改变的是概念模式与存储模式之间的映射关系。
⑨正确。数据库系统的存储模式如有改变，概念模式无需改动。
⑩错误。数据库中，数据一致性是指表示同一数据的多个副本之间没有矛盾，完全一致。
[答案]A：② B：④ C：⑤ D：⑦ E：⑨

1.6 多媒体基础知识

1.6.1 主要知识点

掌握多媒体的基本概念，熟悉图形、图像格式和基本音频、视频知识。

1.6.1.1 常见的音频格式

（1） WAVE，扩展名为 WAV。

该格式记录了声音的波形，只要采样率高、采样字节长、计算机速度快，利用该格式记录的声间文件能够和原声基本一致。WAVE 的唯一缺点就是文件太大，毕竟它要把声音的每个细节都记录下来，而且不压缩。

（2） MOD，扩展名为 MOD，ST3，XT，S3M，FAR 和 669 等。

MOD 是一类音乐文件的总称，逐渐发展产生了 ST3，XT，S3M，FAR 和 669 等扩展格式，而其基本原理还是一样的。该格式的文件不仅存放了乐谱，而且存放了乐曲使用的各种音色样本，具有回放效果明确、音色种类永无止境的优点。

（3） MPEG-3，扩展名为 MP3。

MPEG-3 压缩较大，是一种有损压缩，其实际音质并不完美。在网络、可视电话能信方面，MP3 大有用武之地。由于本质不同，所以它没法和 MOD、MIDI 相提并论。从 HIFI 角度上讲，MP3 有损失，而 MOD 和 MIDI 则没有。

（4） Real Audio，扩展名 RA。

强大的压缩量和极小的失真度使其在众多格式中脱颖而出。与 MP3 相同，它也是为了解决网络传输带宽资源设计的，因此其主要目标是提高压缩比和容错性，其次才是音质。

（5）Creative Musical Format，的扩展名为 CMF。

这是 Creative 公司的专用音乐格式。它和 MIDI 差不多，只是音色、效果上有些特色，专用于 FM 声卡。不过其兼容性差，且效果无法和别的格式相提并论。

（6）CD Audio 音乐 CD，扩展名为 CDA。

CDA 格式就是唱片采用的格式，又叫"红皮书"格式，记录的是波形流。CDA 的缺点是无法编辑，文件长度太大。

（7）MIDI，扩展名为 MID。

作为音乐工业的数据通信标准，MIDI 可谓是一种非常专业的语言，它能指挥各种音乐设备的运转，而且具有统一的标准格式，甚至能够模仿用原始乐器的各种演奏持巧无法演奏的效果。MIDI 文件长度非常小。MIDI 的一个缺点是不能记录语音。

1.6.1.2 多媒体数据压缩的编码技术标准。

（1）H．261

H．261 是用于音频视频服务的视频编码和解码器（也称 PX64 标准）。应用目标是可视电话和视频会议系统。含有此标准的系统必须能实时地按标准进行编码和解码。H.261 与 JPEG 及 MPEG 标准间有明显的相似性,区别是 H.261 是为动态使用而设计的,并提供完全饮食的组织和高水平的交互控制.

(2)JPEG

JPEG 是静止图像压缩和解压缩算法的标准.JPEG 成为 ISO 的国际标准.

(3)MPEG

MPEG-X 版本是指一组 ITU 和 ISO 制定发布的视频、音频和数据的压缩标准。现在有 3 个版本的 MPEG：MPEG-1、MPEG-2 和 MPEG-4。MPEG 在以下 3 方面优于其他压缩/解压缩方案：具有很好的兼容性；MPEG 能够比其他算法提供更好的压缩比，最高达 200：1；更重要的是，MPEG 在提供高压缩比的同时，数据压缩的损失很小。

（4）DVI

DVI 视频图像压缩算法的性能与 MPEG1 相当，即图像质量可达到 VHS 的水平。压缩后的图像数据率约为 1。5Mbit/s.为了扩大 DVI 技术的应用，Intel 公司最近又推出了 DVI 算法的软件解码算法,称为 Indeo 技术。它能将数字视频文件压缩为五分之一到十分之一。Indeo 技术使用了多类"有损"和"无损"压缩技术。

1.6.2 试题解析

现在多媒体基础知识也列入新的大纲，1999 年度和 2000 年各有一道专门考查多媒体知识的试题，分别考查了音频知识和多媒体压缩技术，此前这方面的试题很少。复习时应掌握基本概念，熟悉有关的多媒体文件容量、量化方面的计算。

试题 1 （2000 年试题 8）

从供选择的答案中，选出应填入下面叙述中{ }内最确切的解答，把相应编号写在答卷的对应栏内。

数据压缩技术是多媒体信息处理中的关键技术之一，数据压缩技术可分为 A 两大类。B 是一种与频度相关的压缩和编码方法 ，C 主要用于视频信息的压缩， D 则常用于静止图片的信息压缩。由三基色（RGB）原理出发的 RGB 彩色空间，在多媒体技术中最常用的，此

外还有多种彩色空间，但 E 不是计算机上用的彩色空间。
供选择的答案
A：①可逆与不可逆 ②高速与低速 ③编码与非编码 ④冗余与非冗余
B：①MIPS ②ISDN ③Huffman ④Gauss
C:①MIPS ②MPEG ③JPEG ④JIPS
D:①MIPS ②MPEG ③JPEG ④JIPS
E:①YUV ②HIS ③XYZ ④IMG
[解析]
本题主要考查多媒体数据压缩技术，还涉及到彩色空间表示。
一般将数据压缩技术分为无损压缩和有损压缩两大类。无损压缩和用数据的统计冗余进行压缩，可完全恢复到原始数据，没有任何失真，因此是可逆的。有损压缩利用人的视、听觉器官对图像或声音的某些频率成分不敏感的特性，在压缩过程中损失一定信息，不能完全恢复原始数据，因此是不可逆的。
Huffman 编码就是一种基于统计的无损压缩方案，是一种与频度有关的压缩编码方法。实际上的问题 B 提供的答案中，只有"Huffman"是压缩编码方法。
JPEG 是静止图像压缩和解压缩算法的国际标准。MPEG 一组由 ITU 和 ISO 制定发布的视频、音频、数据的压缩标准。
由三基色（RGB）原理出发的 RGB 彩色空间，在多媒体技术中是最常用的，此外还有多种彩色空间，如 YUV、HIS、CIEXYZ、CIELAB、CCIR601-2YcbCr。问题 E 提供的答案"IGB"不是彩色空间方面的术语。
[答案]
A: ① B:③ C: ② D:③ E:④

试题 2（1999 年试题 8）

从供选择的答案中,选出相应填入下面叙述中{ }内的最确切的解答,把相应编号写在答卷的对应栏内.
在多媒体的音频处理中,由于人所敏感的声频最高为 A 赫兹(Hz),因此,数字音频文件中对音频的采样频率为 B 赫兹(Hz)。对一个双声道的立体声，保持 1 秒钟声音，其波形文件所需的字节数为 C ，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 D ，它是该领域国际上的一个 E 。
供选择的答案
A：①50 ②10k ③22k ④44k
B:①44.1k ②20.05k ③10k ④88k
C:①22050 ②88200 ③176400 ④44100
D:①语音数字接口 ②乐器数字接口 ③语音模拟接口 ④乐器模拟接口
E：①控制方式 ②管理规范 ③通信标准 ④ 输入格式
[解析]
本题考查多媒体音频的基本常识，非常简单。
在多媒体的音频处理中，采样频率是决定音频质量的一个重要因素。人的听觉带宽一般为 20Hz～20kHz,人敏感的声频最高为 22kHz。目前根据音频质量所确定的频率范围如下：
电话话音为 200Hz～3.4kHz
调幅广播为 50Hz～7kHz
调频广播为 20Hz～15kHz
宽带音频为 20Hz～20kHz

而常用的音频采样频率为 8kHz、2.025kHz、16kHz、22.05kHz、37.8kHz、44.1kHz,数字音频文件中对音频的采样频率为 44.1kHz.
信号编码的位数是决定音频质量的另一个重要因素，它决定数字采样的可用动态范围和信噪比。
对一个双声道的立体声，由于两个声道，在每个采样点的量化位数为 8 位的情况下，保持一秒钟的声音码，则包含 $2\times44.1\times10^3$ 个采样点的数据，那么就是 88200 个字节。
MIDI 的全名为 Musical Instrument Digital Interface,即乐器数字的接口，泛指数字音乐的国际标准。MIDI 的标准规定了电子乐器与计算机连接的电缆和硬件，还指定了在装置间传送数据的通信协议。任何电子乐器，只要有处理 MIDI 信息的处理器和适当的硬件接口，都能够成为 MIDI 装置。利有 MIDI 文件演奏音乐，所需存储量很小，演奏 2 分钟乐曲的 MIDI 文件，文件大小不到 8kB.。
[答案]A：③ B：① C：② D：② E：③

试题 3（1996 年试题 10）

从供选择的答案中,选出应填入下面叙述中{ }内最确切的解答,把相应编号写在答卷的对应栏内.
多媒体技术的关键在于解决动态图像和声音的存储与传输问题.若不经压缩,以 VGA640×480 点阵存储一幅 256 色的彩色图像大约需 A MB 存储空间,以 9600bit/s 的速度传输这幅图像大约需 B 秒,按我国电视 PAL 标准每秒 25 幅,一张 650MB 的光盘可容纳约 C 秒的这样图像画面,播放时传送速率应不低于每秒 D MB.模拟声音数字化存放是通过采样和量化实现的,若采样频率 44.1kHz,每个样本 16 位,存放一分钟双声道的声音约占 E MB 存储空间.
供选择答案
A～E: ①0.3 ②1.4 ③2.4 ④7.5 ⑤10
⑥32 ⑦78.6 ⑧87 ⑨98.4 ⑩256
[解析]
因为 640×480 点阵存储一幅 256 色彩色图像所需的存储空间为 $256=2^8$,所以存储 256 色后个点的信息需 1Byet 存储空间,一幅图像的存储量=640×480×1Byte=307200Byte≈0.3MB,问题 A 的答案应选①.
传输一幅图像的时间为 307200Byte/(9600bit/s)=256s,(注:bit/s 为比特/秒,1 字节为 8bit),所以问题 B 的答案应选⑩.
650MB 光盘可容纳(650MB/0.3MB)/(25 副/秒)=2167 副/（25 副/秒）=87 秒的图像。问题 C 应选⑧.
播放时传送速率为 0.3MB/幅×25 幅/秒=7.5MB/秒,所以问题 D 的答案应为④.
已知声间的采样率为 44.1kHz,样本 16 位,双声道,则存放一分钟声音的存储空间为 44.1kHz×166bit×2×60 秒=84.7Mbit/8=10.6MB,所以问题 E 的答案为⑤.
[答案]A:① B:⑩ C:⑧ D:④ E:⑤

试题 4（1995 年试题 24）

从供选择的答案中,选出相应填入下面关于 CAD 中彩色产生方法叙述中{ }内的正确答案,把编号写在答卷的对应栏内.
彩色图形显示和印刷设备采用三原色方法.CRT 彩色显示器选用 RGB(红、绿、蓝)三原色，是 A 混色系统。彩色喷墨等彩色图形印刷设备选用 CMY（青、品红、黄）三原色，是 B 混色系统，黑色在这两种系统中的表示是 C 。
某种颜色的补色是从白色减去该颜色成分后所呈现的色彩。CMY 分别是 RGB 的补色。CRT

显示的纯红色，在用彩色喷墨印刷输出时其 CMY= D ，而白纸上的青色图形在 CRT 显示器中其 RGB= E 。

供选择的答案：

A、B：①减色 ②加色 ③比例乘积色 ④互补色 ⑤荧光 ⑥油墨 ⑦光电发射⑧印刷

C：①RGB=000，CMY=000 ②RGB=000，CMY=111

③RGB=111，CMY=000 ④RGB=111，CMY=111

D、E：①000 ②001 ③010 ④011 ⑤100 ⑥101 ⑦110 ⑧111

[解析]

不同颜色通过配色产生新的颜色。适当选择 3 种不同颜色相加，可以配成几乎所有可见的颜色，这就是三原色原理。三原色的构成是红、绿、蓝。

在 CRT 彩色显示器中，通过靠得很近的 RGB（红、绿、蓝）3 个发光点产生混色，就是相加混色系统。当 RGB=111 时便呈现白色。人们看到印在纸上的颜色，这是纸反射部分的入射光呈现的颜色。当入射光包含全部可见光颜色（白色），而所用纸吸收了红色光，它反射的便是红色的补色即青色（C）了。这也可看作反射光是由绿和蓝两种颜色相加组成，这便是青色颜料的作用。同样，品红色（M）颜料和黄色（Y）颜料，分别是吸收了白色入射光中绿不色和蓝色后呈现的颜色，这是减色混色系统。当同时使用 3 种颜色 CMY=111 时，它吸收了白色入射光中的红、绿、蓝三色，自然呈现黑色。

RGB 加色混色系统工程的颜色若为 CMY 减色混色系统表示时，可用公式（1）计算。同样 CMY 减色混色系统中的颜色若用 RGB 加色混色系统表示时，可用公式（2）计算。

$$\begin{bmatrix} C \\ M \\ Y \end{bmatrix} = \begin{bmatrix} R_{白} \\ G_{白} \\ B_{白} \end{bmatrix} - \begin{bmatrix} R \\ G \\ B \end{bmatrix} \quad \ldots\ldots(1)$$

$$\begin{bmatrix} R \\ G \\ B \end{bmatrix} = \begin{bmatrix} C_{黑} \\ M_{黑} \\ Y_{黑} \end{bmatrix} - \begin{bmatrix} C \\ M \\ Y \end{bmatrix} \quad \ldots\ldots(2)$$

其中白色在 RGB 中是 R 白=G 白=B 白=1，黑色在 CMY 中是 C 黑=M 黑=Y 黑=1.例如 RGB 中的纯红色是 RGB=100,在 CMY 中便是 CMY=011.白色光被吸收了绿色和蓝色,反射光呈红色.同样在 CMY 中的青色是 CMY=100,在 RGB 中便是 RGB=011,由绿色和蓝色相加得青色.

[答案]A:② B:① C:② D:④ E:④

试题 5 （1993 年试题 9）

从供选择的答案中，选出应填入{ }内的正确答案，把编号写在答卷的对应栏内。

多媒体技术是当前计算机发展的一个热门方向。这里，多媒体的含义主要是指 A 等多种 B 。它强调多媒体信息的 C 。C 是多媒体发展中要解决的关键技术之一。在 SRAM、CD-ROM、磁带和高密度软盘 4 种存储器中，当前最适合用来存储多媒体信息的是 E。

供选择的答案

A：①如磁带、磁盘、光盘 ②如双绞线、同轴电缆、光纤

③传输信息的介质 ④表达信息的形式

B：①输入输出的设备样 ②存储信息的实体 ③传输信息的介质 ④表达信息的形式

C：①分时处理 ②批处理 ③综合与集成处理 ④分布式处理

D：①压缩技术 ②可靠性技术 ③流水线技术 ④加密技术

E：①SRAM ②CD-ROM ③磁带 ④高密度软盘

[解析]
多媒体信息的综合与集成处理，能将文字、声音、静态图形等有机地结合起来。如可将图像经过处理后嵌入图形，配上文字再与声音合成一体。用多媒体计算机模拟施行时，足不出户就能看到动态景物，阅读到景点的文字说明，领略到音响的效果。
声音和活动的图像都是模拟信息，要想将其在计算机中存储与处理，需要对信息进行数字化处理。压缩技术是多媒体发展中要解决的关键技术，压缩信息量，需要有大容量的存储器。在本题中给出的SRAM、CD-ROM、磁带和高密度软盘4种存储器中，只有CD（Compact Disk）-ROM（Read Only Memory ）,即只读光盘最合适用来存储多媒体信息。CD-ROM 的容量可达数百渐兆字节，与压缩技术相结合可存储相当长时间的动态图像，很有实用价值。
[答案]A：③ B：④ C：③ D：① E：②

二 硬件基础知识试题精解

与程序员及试题相比，高级程序员级硬件基础知识试题份是量还有小一些，试题难度也不大。存储器系统知识是考查的重点内容，I/O 系统、虚拟内存、Cache 检错校验码、系统可靠性等知识点值得关注。

1990-2000 年硬件基础知识试题,按涉及知识点统计,考查体系结构和主要部件的有 12 道题,考查存储器系统的有 9 道题,考查安全性与可靠性的有 6 道题,考查体系结构其他基础知识的有 5 道题,还有 3 道综合性试题涉及及并行处理、系统性能评价以及操作系统、网络。

2.1 计算机体系结构的主要部件

2.1.1 主要知识点

掌握机内代码、算术运算和逻辑运算等计算机基础知识，重点掌握海明校验码、循环冗余校验码，系统结构，CPU，I/O 系统和总线结构等方面的知识。由于涉及的知识点较多，这里主要介绍 I/O 系统，其他内容就不一一介绍了。

2.1.1.1 输入输出控制器

输入输出控制器，也称 I/O 模块，负责控制外部设备与主存储器、CPU 的寄存器之间的数据交换。它通过系统接口（内部接口）与主机（CPU、主存储器）交互，通过设备接口（外部接口）与各种外设打交道。

其主要功能有：①控制与定时。I/O 系统必须对通信进行控制和定时，以协调外设和内部资源之间的通信流量，因为内部资源如主存、系统总线等被一系列的操作包括数据的输入输出所共享。②与 CPU 通信。③与设备通信。I/O 系统与外设之间交换控制命令、状态住处以及数据。④数据通信缓冲。I/0 系统需要数据缓冲寄存器来暂存数据，以此来适应通信双方速率的不匹配，从而提高主机利用率。⑤错误检测机制。外设和主机进行数据交换的过程中可能会产生错误，通常使用奇偶校验码、CRC 循环冗余校验码进行检测和纠正。

3．1．1．2 输入输出的工作方式。

输入输出系统主要有 3 种方式与主机交换数据。

（1） 程序控制方式

输入输出完全由 CPU 控制，在整个 I/O 过程中 CPU 必须等待而不能进行其他工作，无法充分发挥 CPU 高速的处理能力。CPU 发出 I/O 命令，命令中包含了外设的地址信息和所要执行的操作，相应的 I/O 系统将执行命令并设置状态寄存器。CPU 不停地定期地查询 I/O 系统以确定该操作是否已经守成。当有多个外设需要和主机交换数据时，CPU 必须定期轮询外设，以确定每个外设的状态。其查询方式可以分为两类：串行点名和并行查询。

（2） 程序中断方式

CPU 利用中断方式完成数据的输入/输出。当 I/O 系统与外设交换数据时，CPU 无需等

待也不必查询 I/O 的状态，而可以处理其他任务。当 I/O 系统完成了数据传输后，则以中断信号通知 CPU。CPU 保存正在扭亏为盈行程序的现场后，转入 I/O 中断服务程序完成与 I/O 系统的数据交换，再返回原主程序继续执行。与程序控制方式相比，中断方式因为 CPU 无需等待而提高的效率。

在系统中具有多个中断源的情况下，常用处理方法有多中断信号线法、中断软件查询法、雏菊链法、总线仲裁法和中断向量表法。

（3） DMA 方式

CPU 只是在数据传输前和完成后才介入，而数据的传输过程由 DMA 控制器来管理，无需 CPU 参与。数据直接写入或读出主存储器，不再经 CPU 中转。

中断法虽然比程序控制法更加有效，但两者都是由软件来完在数据的传输，难以胜任高速的数据传输需求。在主存和 I/O 模块之间进行数据交换仍需由 CPU 控制，并且每一数据都需经过 CPU 中转。因此不仅影响了 CPU 的工作，也降低了 I/O 数据传输率。

DMA 方法使用 DMA 控制器（DMAC）来控制和管理数据传输。DMAC 和 CPU 共享受系统总线，并且具有独立访问存储器的能力。在进行 DMA 传输时，CPU 放弃对系统总线的控制而由 DMAC 来控制总线，由 DMAC 提供存储器地址及必需的读写控制信号，实现外设与存储器之间的数据交换。

DMAC 获取总线的 3 种方式：

①暂停方式。DMAC 请求 CPU 让出系统总线，由 DMAC 使用，直到一组数据全部传送完毕后，再把总线控制权还给 CPU。

②周期窃取方式。请求 CPU 进入空闲态，暂时放弃总线，插入一个 DMAC 周期，传送完一个字后，把总线还给 CPU。

③共享方式。在 CPU 不使用总线时，DMAC 进行 DMA 传输。

DMA 的 3 种组织方式：

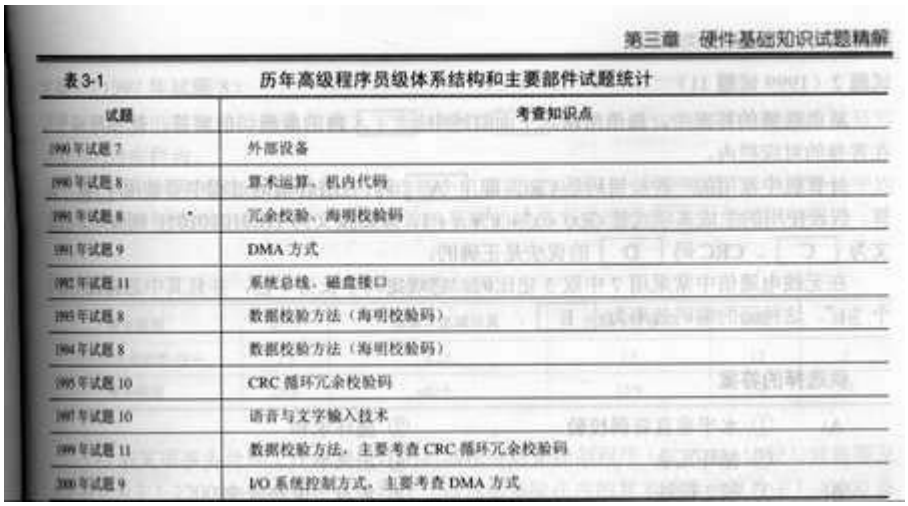
① 单总线分离 DMA 方式；

② 单总线集成 DMA 方式；

③ I/O 总线方式

2.1.2 试题解析

根据考试大纲，存储器系统知识已作为独立的一个大项，所以有关计算机体系结构和主要部件的内容相对少了，归到此类的试题也就不多了。从历年试题统计（见表 3-1）来看，数据校验方法（特别是海明码和循环冗余校验码）、I/O 系统控制方式（特别是 DMA）是反复考查的重点，而考查具体外部设备知识的试题很少，这也反映了对高级程序员的要求较高。值得一提的是，2000 年度试题 9 与 1991 年试题 9 考查的内容基本相同。



第三章　硬件基础知识试题精解

表 3-1　历年高级程序员级体系结构和主要部件试题统计

试题	考查知识点
1990 年试题 7	外部设备
1990 年试题 8	算术运算、机内代码
1991 年试题 8	冗余校验、海明校验码
1991 年试题 9	DMA 方式
1992 年试题 11	系统总线、磁盘接口
1993 年试题 8	数据校验方法（海明校验码）
1994 年试题 8	数据校验方法（海明校验码）
1995 年试题 10	CRC 循环冗余校验码
1997 年试题 10	语音与文字输入技术
1999 年试题 11	数据校验方法，主要考查 CRC 循环冗余校验码
2000 年试题 9	I/O 系统控制方式，主要考查 DMA 方式

试题 1 （2000 年试题 9）

从供选择的答案中，选出应填入叙述中{ }内的最确切的解答，把相应编号写在答卷的对应栏内。

直接存储器访问（DMA）是一种快速传送大量数据常用的技术。工作过程大致如下：

1． 向 CPU 申请 DMA 传送。

2． 获 CPU 申请 DMA 控制器接管 A 的控制权。

3． 在 DMA 控制器的控制下，在存储器和 B 之间直接进行数据传送，在传送过中不需要 C 的参与。开始时需提供要供送的数据的 D 和 E 。

4． 传送结束后，向 CPU 返回 DMA 操作完成信号。

供选择的答案

A：①系统控制台 ②系统总线 ③I/O 控制器 ④中央处理器

B：①外部设备 ②运算器 ③缓存 ④中央处理器

C：①外部设备 ②系统时钟 ③系统总线 ④中央处理器

D：①结束地址 ②起始地址 ③设备类型 ④数据速率

E：①结束地址 ②设备类型 ③数据长度 ④数据速率

[解析]

这是一道考查 DMA 方式的概念题，内容单一，解答起来容易，具体知识请参见 3。1。1。2 节有关 DMA 的内容，以及本节试题 10（1991 年试题 9）。

[答案]A：② B：① C：④ D：② E：③

试题 2 （1999 试题 11）

从供选择的答案中，选出应填入下面叙述中{ }内的最确切的解答，把相应编号定在答卷的对应栏内。

计算机中常用一种检错码是 CRC，即 A 码。在进行编码过程中要使用 B 运算。假设使用的生成多项式是 $G(x)=x^4+x^3+x+1$。原始报文为 11001010101，则编码后的报文为 C 。CRC 码 D 的说法是正确的。

在无线电通信中采用 7 中取 3 定比码，它规定码字长为 7 位，并且其中总有且仅有 3 个"1"。这种码的编码效率为 E 。

供选择的答案

A：①水平垂直奇偶校检 ②循环求和 ③循环冗余 ④正比率

B：①模 2 除法 ②定点二进制除法 ③二-十进制除法 ④循环移位法

C：①1100101010111 ②110010101010011 ③11001010101011100 ④110010101010101
D：①可纠正一位差错 ②可检测所有偶数位错 ③可检测所有小于校验位长度的突发错④可检测所有等于、小于校验位长度的突发错
E：①3/7 ②4/7 ③$\log_2 3/\log_2 7$ ④（$\log_2 35$）/7
[解析]
计算机在存储和传送数据的过程中，为了保证数据的准确性，一般要进行数据校验和纠错。CRC（循环冗余）码是一种常见的校错码。
假设被校验的二进制信息表达 C（X）表示，则 C（X）=CK-2…C1C0 取值为 0 或者 1。将 C（X）左移 n-k 位，则可表示成 $c(x)*2^{n-k}$，这样 C（X）的左边就会空出 n-k 位，这 n-k 位就是校验码的位置。
CRC 码就是用 $c(x)*2^{n-k}$ 除以生成多项式 g(x)，所得到的余数即为所求的校验位。
在本题中，生成多项式是 110011，按照上面的算法进行运算，就能够得到余数 0011，将其添加到原始报文的末尾，就得到编码后的报文。
无线电中常采用的 7 中取 3 定比码，规定码字长为 7 位，并且其中总有且仅有 3 个"1"，那么它的编码效率是这样计算的：
首先，7 位中有且仅有 3 个"1"的数据可以有 C^3_7 个，即（7*6*5）/（3*2*1）=35；
其次，取以 2 为底的对数，得到 $\log_2 35$;
最后，除以总位数 7，得到$(\log_2 35)/7$。
这样我们就可以看出求编码效率的公式：（$\log_2$(码字数)）/总位数。
[答案]A：③ B:① C: ② D:④ E:④

试题 3（1997 年试题 8）

从供选择的答案中，选出应填入下面叙述中{ }内的最确切的解答，把相应编号定在答卷的对应栏内。
某 CPU 的主振频率为 100MHz,平均每个机器周期包含 4 个共振周期.各类指令的平均机器周期数和使用频度见表 3-2,则该计算机系统的速度为平均约 A 兆指令/秒.

表 3-2　　1997 年试题 8

指令类别	访内	一般算术逻辑运算	比较与转移等	乘除	其他
平均机器周期数/指令	2.5	1.25	1.5	15	5
使用频度	25%	40%	25%	5%	5%

(1) 若某项事务处理工作所要执行的机器指令数是控制程序(以访内比较与转移等及其他指令为主)220000 条指令和业务程序（以包括乘除在内的算术逻辑运算为主）90000 条指令，且指令使用频度表 3-2，则该计算机系统的事务处理能力约为 B 项/秒。
(2) 若其他条件不变，仅提高主振频率至 150MHz，则此时该计算机速度平均约为 C 兆指令/秒，对上述事务的处理能力约为 D 项/秒。
(3) 若主频仍为 100MHz，但由于采用了流水线和专用硬件等措施，使各类指令的每条指令平均机器周期数都变为 1。25，此时，计算机的速度平均约 E 兆指令/秒。
供选择的答案：
A～E：①1 ②5 ③10 ④15
⑤33.3 ⑥50 ⑦66.7 ⑧100
[解析]
指令周期（Instruction Cycle）指的是执行一条指令所需的时间；总线周期（Bus Cycle,也称作主振周期）指的是 CPU 从存储器或 I/O 端口存取一个字节所需的时间；时钟周期（Clock

Cycle，也称作主振周期）指的是 CPU 处理动作的最小单位。指令周期划分一个个总线周期，不同的指令所需的总线周期也不相同，有的只需一个，有的只需要若干个，一个总线周期通常包括 4 个时期周期。

CPU 的主振频率（时钟频率）为 100MHz,即该 CPU 每秒有 100M/4=25M 个主振周期.该计算机的平均指令周期为 2.5*25%+1.25*40%+1.5*25%+15*5%+5*5%=2.5 个主振周期,所以该计算机的速度平均约为 25M/2.5=10M 指令/秒.所以问题 A 的答案为③.

某项事务处理工作所要执行的控制程序的平均指令周期为:

(2.5*25%+1.5*25%+5*5%)/(25%+25%25%)=2.2727

业务程序的平均指令周期为:

(15*5%+1.25*40%)/(5%+40%)=2.7777

完成一项事务处理工作所需的机器周为:

220000*2.2727+90000*2.7777=499994+249993=749987

这样该计算机系统的事务处理能力约为 25M/749987=33.33 项/每秒.所以问题 B 的答案为⑥.

若其他条件不变,仅提高计算机主频至 150MHz,此时该 CPU 每秒有 150M/4=37.5M 个主振周期,该计算机的速度平均约为 37.5M/2.5=1.5M 指令/秒.对上述事务的处理能力为 37.5M/749987=50.00 项/每秒.所以问题 C 的答案为④,D 的答案为⑦.

若主频仍为 100MHz,但采用的流水线及专用硬件等措施,使得各类指令的平均机器周期数都变为 1.25,则此时该计算机的速度平均约为 25M/1.25=20M 指令/秒.所以问题 E 的答案为⑤.

[答案]A:③ B:⑥ C:④ D:⑦ E:⑤

试题 4（1997 年试题 10）

从供选择的答案中，选出应填入下面叙述中{ }内的最确切的解答，把相应编号定在答卷的对应栏内。

语音与文字输入技术是研究如何将计算机主要由键盘输入文字数据的方式逐步改变成由人们口述或写入的方式送入信息.语音与文字输入的过程包含有 A、预处理、特征抽取、B 与分类决策等环节。A 阶段通过传感器获得的模拟电信号要经过模数据转换变成数字信号，它需要对模拟电信号抽样测量，将测量的值 C ，以便用二进制数字信号来表示。预处理的主要任务是进行削弱无用住处和增强有用信息的工作。特征抽取将上述环节产生的输入样本以有利于决策的形式表示出来，典型的常用方法是 D 表示法。B 又称为分类器学习。分类决策则通过比较和决策来完成对输入信息的 E 。

供选择的答案

A～E：①辨认识别 ②特征向量 ③频谱分析 ④分级取整

⑤消除噪声 ⑥模型生成 ⑦数据合成 ⑧信息获取

[解析]

语音与文字输入技术是研究如何将人们口述或书写的信息直接输入计算机。实现其输入过程的系统构成如图 3-1 所示。

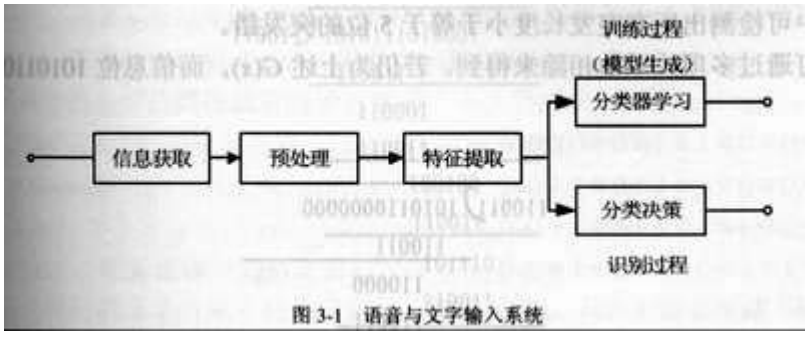


图 3-1　语音与文字输入系统

各部分的主要功能如下：
信息获取是将输入的语音与文字利用传感器转换成电信号，并从模拟量转换成数字量，它要对模拟电信号抽样测量，将测量的值分级取整。
预处理的主要作用是去除信号中的噪音，并增强有用信息。特征抽取（也称为特征提取）是指将经过预处理的信息输入计算机之后，从中提取能代表该信息特点的一组代码或向量，与在此之前已存入机内模型库的标准语音或文字的特征向量进行比较，从库中找出与输入向量最相似的标准语音或文字的特征向量进行比较，从库中找出与输入向量最相似的标准语音或文字作为最终的输入语音或文字，这一比较过程称之为匹配。完成了匹配过程也就完成了输入过程，因此如何提取特征向量是很重要的。
特征向量是根据文字或语音的特点按某一规律转换生成的。不同人写的文字或说话的口音都有所不同，因此增加了语音和文字输入技术的难度。从以上说明可以看到，在生成模型库时需要提取特征向量，作为语音或文字的样本；在应用时需要提取人的语音或文字或语音的特征几量，代机器识别用。分类器学习是指机器通过训练（学习），提取各个已知文字或语音的特征向量，建立模型库。分类决策是指对人输入信号进行辨认识别，最终完成语音或文字的识别工作。
[答案]A：⑧ B：⑥ C：④ D：② E：①

试题 5 （1995 年试题 10）

从供选择的答案中，选出应填入下面叙述中{ }内的最确切的解答，把相应编号定在答卷的对应栏内。
某循环冗余码（CRC）的生成多项工为 G（X）=X^5+X^4+X+1，则它对任意长度的信息位产生 A 位长的冗余位，并可检验出所有突发长度 B 位的突发错误。若信息位于为 1010100，采用此生多项产生冗余位，加在信息位后形成的码字是 C 。在读出或接收端读到的码字中若不满足某种规律则可判断其必然有错。例如 D 和 E 就是出错的码字。
供选择的答案：
A、 B：①小于等于 4 ②4 ③小于等于 5 ④5 ⑤大于等于 5 ⑥大于 5
C ～ E ： ①1010110111111 ②1010111010001 ③1010110001101 ④1010111101010 ⑤1010111011001 ⑥1010110001100
[解析]
循环冗余码中的冗余位是由其生成多项式 G（X）去除信息多项工后得到的余式所决定的。若生成多项式为 K 次多项式，则余式为 K-1 次多项式，它对应的冗余位（即余式）的系数为 K 位。本题中 G（X）=X^5+X^4+X+1 为 5 次多项式，则冗余位为 5 位长。
可以证明由 K 次生成多项式产生的冗余码能检测出所有突发长度小于等于 K 位的突发错误，故本题可检测出所有突发长度小于等于 5 位的突发错觉。
冗余地位可通过多项式系数相除来得到。若仍为上述 G（X），而信息位 10101100，则除法过程如下：

```
          ____________
110011 | 10101100000
         110011
         ------
          110000
          110011
          ------
              110000
              110011
              ------
                 01100
```

故冗余位为 01100，它加在信息位后的码字为 1010110001100。这里，需要指出 3 点值注意的地方。第一，上述除法中，被除数不是信息位，而应为信息位后面加上 K 位 0（这里 K=5）。第二，这里所有的加法为半加，无进位，因而除法中所有减法亦为半减，亦无借位。此时，0-1 和 1-0 一样，结果都为 1，且不向高位借位。第三，上述除式中，我们未写出商，因为我们关心的是余数，商为多少是无关紧要的。

上面产生的码字是问题 C ～E 供选择答案中的⑥。不难看出用正确方式产生的码字再被生成多项式 G（X）的系数中除得到的余数必为 0，即可以整除。若将此码字存入存储器或发往远地，若再读了或接受时码字不满足上述规律，则必然在存取或传输中出现了差错。这就是循环冗余的检错觉功能。考查 C～E 供选择答案中的其余各码字，比如说①，因为：

```
110011 ) 10101110111111
         110011
         110000
         110011
         111111
         110011
```

能整除，故它不是出错的码字。

再看 2，因为不能整除，故该码字有错。码字 3 和码字 6 有相同的信息位（前 8 位），而后 5 位冗余不同，既然 6 是正确的码字，则 3 必然是错误的码字。这样，我们已经找到了题目所要求的两个错误码字。实际上，若我们将码字 4 或 5 用 110011 去除，可以看到都能被其整除，即它们都不是出错的码字。

```
110011 ) 10101110010001
         110011
         110001
         110011
         100100
         110011
         101110
         110011
         111011
         110011
           1000
```

[答案]A：④ B：③ C：⑥ D：② E：③

试题 6 （1994 年试题 8）

从供选择的答案中，选出应填入下面叙述中{ }内的最确切的解答，把相应编号定在答卷的对应栏内。

某机器中码字长 15 位（包括信息位和海明校验位），采用了可纠正一位错的海明校验。识码字位从左到右用 b1,b2…b15 编号，其海明校验方程式为

b1⊕b3⊕b5⊕b7⊕b9⊕b11⊕b13⊕b15=0

b2⊕b3⊕b6⊕b7⊕b10⊕b11⊕b14⊕b15=0

b4⊕b5⊕b6⊕b7⊕b12⊕b13⊕b14⊕b15=0

b8⊕b9⊕b10⊕b11⊕b12⊕b13⊕14⊕b15=0

若在答案中给出的码字最多只有一位错，请找出无错的识字 A ；第 2 位 b2 错的码字 B ；b4 错的码字 C ；b6 错的码字 D 和 b8 错的码字 E 。
供选择的答案：
A～E ①000100101101010 ②010100101100010 ③0101001111101010
④010101101101010 ⑤001111010000111 ⑥001111010001111
⑦101111010000111 ⑧001011010000111
[解析]
从题中给出的海明校验的方程式可知：
（1） 若码字无错，则将其相应位代入 4 个校验方程式的左边，计算后应有尽有全为"0"；
（2） 若只是一位错，仔细观察校验方程式的构成可见，b1 只出现在第 1 个校验方程中，因而会使其值由"0"变为"1"，而其余 3 个方程的值仍为"0"，若将 4 个校验议程的值由下至上排列为 0001，正好是二进制值 1，同样， b2 只出现在第 2 个校验方程中：
（3） 若只有 b2 值，则 4 个校验方程的值由下至上排列后为 0010，正好是十进制值 2。我们可得到 bi(I=0,2,…,15)一位错，代入校验方程左边求值，并由下至上排列后正好是 I 的二进制值。
我们可分别计算各个选择答案的校验方程的左边的值：
b1=0,b2=0,b3=0,b4=0,b5=0,b6=0,b7=0,b8=0,b9=0,b10=0,b11=0,b12=0,b13=0,b14=0,b15=0, 分别代入 4 个校验方程的左边得到：
0⊕0⊕0⊕1⊕1⊕0⊕0⊕0=0
0⊕0⊕0⊕1⊕1⊕0⊕1⊕0=1
1⊕0⊕0⊕1⊕1⊕0⊕1⊕0=0
0⊕1⊕1⊕0⊕1⊕0⊕1⊕0=0
①$(0010)_2$=2,故第 2 位 b2 错.对于其他供选择的答案可进行类似的计算得到 4 个样验方程左边的值.
②$(1100)_2$=12.第 12 位 b12 错.
③$(1000)_2$=8,第 8 位 b8 错
④$(0110)_2$=6,第 6 位 b6 错
⑤$(0000)_2$=0,无错.
⑥$(1100)_2$=12,第 12 位 b12 错.
⑦$(0001)_2$=1,第 1 位 b1 错.
⑧$(0100)_2$=4,第 4 位错 b4.
[答案]A:⑤ B:① C:⑧ D:④ E:③

试题 7（1993 年试题 8）

从供选择的答案中，选出应填入下面叙述中{ }内的最确切的解答，把相应编号定在答卷的对应
栏内。
假设机器中存有代码 0100011.
87 1
若将该码视为海明码,其校验方程为 b1⊕b3⊕b5⊕b7=0,b2⊕b3⊕b6⊕b7=0,b4⊕b5⊕b6⊕b7=0,经校验其出错位为第 A 位.
若将该码的第 7 位至第 4 位视为信息位,它的(7,4)循环码的生成多项式为 $g(x)=1+x+x^3$,则信息位后随冗余位构成的循环码为 B .将该码第 8 位添加偶校验后,若视为十六进制数为 C ;若视为余 3 码,对应的十进制数为 D ;若视为移码,代表的十进制数为 E .

供选择的答案

A:①2 ②4 ③5 ④6 ⑤7

B:①0100011 ②0100111 ③0100100 ④0100101 ⑤0100110

C ～E:①A3 ②B3 ③23 ④35 ⑤46

⑥53 ⑦70 ⑧73 ⑨83 ⑩C6

[解析]

若某位序号的二进制为 X3X2X1，则当 Xi=0(Xi=0 或 1,I=1,2,3)时，该位不出现在第 I 个校检方程中；而 Xi=1 时，该位出现在第 I 个校验方程。因此，若将代码的值代入 b1⊕b3⊕b5⊕b7=0、b4⊕b5⊕b6⊕b7=0 的左面，分别求出其值 x1、x2、x3，如果全为 0 则无错；如果不全为 0，则单个出错位是 X3X2X1。

对本题中具体给出的代码来说：

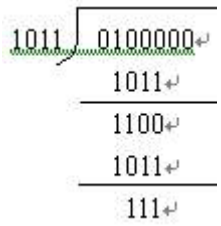
1⊕0⊕0⊕0=1=X1

1⊕0⊕1⊕0=0=X2

0⊕0⊕1⊕0=1=X3

X3X2X1=101 $_{(2)}$ 即其错位为第 5 位。

若题中给出代码的第 7 至第 4 位为信息位，则 0100，其对应的信息多项式 M（X）是 X2；循环码生成多项式 g(x)=x3+x+1,其系数为 1011；冗余位的位数即生成多项式 g(x)的次数，为 3。按照循环码的法则，利用多项式系数相除可求出余式 R（x）的系数，所有减法都是"半减"，不论 1 减 0 或都 0 减 1 所得的差都是 1，无"错位"的问题，见下式：



故 R（x）的系数，即冗余位，为 111。信息位后随冗余位构成的循环码为 0100111。

将题中给出代码在第 8 位添加偶校验位后成 10100011。若将它视为十六进制数则为：10100011 $_{(2)}$ =A3 $_{(16)}$

若将它视为余 3 码，因 1010 对应于十进制数位 7，而 0011 对应于十进制数位 0，故其对应的十进制数为 70。若将 10100011 视为移码，它所对应的 M 进制真值为 100011，化为等值的十进制数是 35。

因此，若将机器中代码 0100011

87 1

视为海明码，其校验方程为 b1⊕b3⊕b5⊕b7=0,b2⊕b3⊕b6⊕b7=0, b4⊕b5⊕b6⊕b7=0,经校验其出错位为第 5 位。若把该码的第 7 位至第 4 位视为信息位，它的（7，4）循环码的生成多项式为 $g(x)=1+x+x^3$，则信息位后随冗余位构成的循环码为 4。将该码第 8 位添加偶校验后，若视为十六进制为 A3；若视为余 3 码，对应的十进制数为 70；若视为移码，代表的十进制数为 35。这里指的是能纠正一位错的海明码。在无错情况下，3 个校验方程应全部满足。若从 b1 到 b7 有某位出错，必然造成校验方程不能全部满足，即校验方程左面式子的运算结果不全等于 0。

[答案]A：③ B：② C：① D：⑦ E：④

试题 8 （1992 年试题 11）

从供选择的答案中，选出应填入下面叙述中{ }内的最确切的解答，把相应编号定在答卷的

对应栏内。
计算机系统部线是连接器、主存、I/O 控制卡等部件的一组信息线。
例如，A、B 和 C 都是系统总线。A 是 8 位的用于工业控制领域的主流总线，已成为 IEEE961 标准： B 是与工业标准结构的 AT 总线兼容并扩展了其功能的 32 位总线； C 则是 IBM 公司推出的与 AT 总线不兼容的首先在 PS/2 机器上采用的 32 位总线。
设备接口线是 I/O 控制卡和 I/O 设备之间的一组连接线。例如，D 和 E 是两种磁盘接口的标准。E 的传输速度更高，也可支持更大的硬盘空间。
供选择的答案
A～E①ISA ②ST506 ③PS232 ④STD ⑤SCSI ⑥MCA ⑦MULTIBUS ⑧EISA
[解析]
计算机系统总线是一组用来连接计算机中央处理器（CPU）、主存和 I/O 控制卡等部件的信息线，是各部件间传送信息的公共通信，I/O 控制卡通过设备接口线 I/O 设备连接。
支持工业控制领域应用的总线有 STD、STE 和 G-64 等，通常只支持 8 或 16 位的微处理器，但 I/O 扩展能力强、组合灵活、价格低廉，它们属于低端总线。STD 总线的数据总线是 8 位的，是目前最广泛应用于工业控制领域的一种主流总线，也已成为电气电子工程师协会 IEE961 标准；STE 总线可以看成是欧洲版的 STD 总线，已成为 IEEE1000 标准；G-64 总线是另一种欧洲用户使用较广泛的适合于工业控制应用的低端总线。
另一类系统总线向提高系统的处理能力方向发展，如支持 32 位微处理器等，EISA、MCA 和 MULTIBUSI 等都是高端总线。扩展工业标准结构 EISA（Extended industrial Standard Architecture）总线是对原有的 AT 总线即 ISA 总线的扩展，但又仍保持与 ISA 兼容，它的插槽分为两段，较浅的部分是原 ISA 总线的信号线，而较深的部分是 EISA 的信号线，后一部分有一个突出的卡口。这样一来，原有的 ISA 卡可以插入较浅的部分继续使用，而新的 EISA 卡上与插槽突出卡口配合有一凹口，所以可插得列更深。其性能的扩展包括提供快速、单周期的突发传送 32 位数据通道，可达 33Mbit/s 的数据传输率，允许访问多达 8 个 DMA 控制器，可独立编程，支持 6 个级别的集中总线仲裁，可以通过软件对配置构成进行管理等方面。
MCA 是微通道结构（Micro Channel Architecture）总线，在电气性能和机械性能方面都与原有的 AT 总线不兼容，不支持任何现存的 ISA 卡。MULTIBUS 只支持 8 位和 16 位微处理器，为了适应 32 位微处理器的要求，Inter 等公司后来又联合推出第二代的总线 MULTIBUSII.
设备接口线是 I/O 控制卡和 I/O 设备之间的连接线。ST（Standard Type）506 和 SCSI（Small Computer System Interface）是两种可用于连接磁盘设备的接口标准。其中，ST506 主要用于 5。25 英寸的盘上，它是 IBMPC/XT 机的标准硬盘接口。SCSI 在超级微机、工作站和小型计算机中广泛采用，其数据传输率要比 ST506 高，它可支持的最大盘空间也比 ST506 大。
RS232 则是一种串行设备接口线的标准，通常用来连接终端和远程通信设备，不能用于连接磁盘设备。
[答案]A：④ B：⑧ C：⑥ D：② E：⑤

试题 9

从供选择的答案中，选出应填入下面叙述中{ }内的最确切的解答，把相应编号定在答卷的对应栏内。
根据"冗余校验"的思想,码距可用来判断使校验码制冗余的程度,并估价其查错、纠错能力。"8421"码的码距为 A ，因而它 B 。若一组海明（Hamming）码有效信息位 k=4，校验位 r=3，则其码距为 C ，用它能够发现 D 位错，并可纠正 E 位错。
供选择的答案
A、C、D、E：①0 ②1 ③2 ④3 ⑤4 ⑥7

B：①能发现 1 位错 ②能纠正 1 位错 ③能发现并纠正 1 位错 ④不能查错、纠错
[解析]
码距是两个码字间不同位的个数，如 1010 和 1100 中间两位不同，两码字的码距为 2。一个码字集的码距则指该码字集中所有码字对码距的最小值。"8421"码字集中许多码字对（如 0110 和 0111）的码距都是 1，故该码的码距为 1。
"冗余校验"的基本思想是：对于任意给定的信息位，按照一定的编码规则，可求得确定的冗余校验位，一起构成该编码的合法码字。可以证明：一种"冗余校验"码合法码字集的码距离若为 2d+1，则它能够发现 2d 位错，并能纠正 d 位错。因为若合法码字集的码距为 2d+1，则任何 2d 位的错，必然只可能使一个合法码字变成一个非法码字，而不可能变为另一个合法码字，因而能被检测出来。进一步，由于合法码字集的码距为 2d+1，那么任何一个合法码字由于 d 位错而产生的非法码字子集和另一个合法字由于 d 位错而产生的非法码字通过判断它属于哪一个子信息位数 k 和校难验位数 r 间满足海明不等式 $2^r \geq k+r+1$
本题中给出 k=4,r=3，恰好满足该不等式的最低要求（取等号），因而用它能发现 1 位错，并可纠正 2 位错，码距为 3。
[答案]A：② B：④ C：④ D：③ E：②

试题 10 （1991 年试题 90）
从供选择的答案中，选出应填入下面叙述中{ }内的最确切的解答，把相应编号定在答卷的对应栏内。
为了快速传送大量数据，微型计算机中采用存储器直接访问技术，简称 DMA。用 DMA 方式传送进，在存储器和 A 之间直接建立高速传输数据的通路，不需要 B 的干预。利用 DMA 方式传送数据时，数据的传送过程完全由称为 DMA 控制器的硬件控制。DMA
（1） 向 CPU 申请 C 传送。
（2） 在 CPU 允许 DMA 工作时，处理总线控制的转交。
（3） 在 DMA 期间管理 D ，控制数据传送。
（4） 确定数据传送的起始地址和 E ，并在传送过程中不断修正。
（5） 数据传送结束时，给出表示 DMA 操作完成的信号。
供选择的答案
A～E:①控制台 ②硬件 ③外部设备 ④数据长度 ⑤CPU ⑥存储器 ⑦DMA ⑧系统总线 ⑨数据方向 ⑩传输速率
[解析]
直接存储器访问 DMA(Direct Memory Access)是一种不需要 CPU 干预,在存储和外部设备之间直接通过系统总线高速传输数据的方法.在此情况下数据的传送过程完全由 DMA 控制器控制,其过程如下:由 DMA 期间由 DMA 控制器控制,其过程如下:由 DMA 控制器向 CPU 发出 DMA 控制器管理系统总线,控制数据传送,确定数据传达室送的起始地址和传送的数据长度 ,并在每个传送一个数据单元后,对地址和长度分别作增加或减少的相应调整;当长度值减为零后数据传送结束,由 DMA 控制器给出表示 DMA 操作完成的信号,重新由 CPU 接管对系统总线的控制.
用 DMA 方式传送时,在存储器和外部设备之间直接建立高速传输数据的通路,不需要 CPU 的干预.利用 DMA 方式传送数据,数据的传送过程完全由称为 DMA 控制器的硬件控制.DMA 控制器具有如下功能:向 CPU 申请 DMA 传送;在 CPU 允许 DMA 工作时,处理总线控制的转交;在 DMA 期间管理系统总线,控制数据传送;确定数据传送的起始地址和数据长度,并在传送过程中不断修正;数据传送结束时,给出表示 DMA 操作完成的信号.
[答案]A:③ B:⑤ C:⑦ D:⑧ E:④

试题 11（1990 年试题 7）

从供选择的答案中，选出应填入下面叙述中{ }内的最确切的解答，把相应编号定在答卷的对应栏内。

1. 查找时间是 A

2. 光盘可以极大地提高 B .

3. 微型计算机常配的滚筒式绘图机中 C.

4. 与激光打印机有关的概念是 D

5. 阅读条码的硬件设备是 E.

供选择的答案

A: ①使磁头移动到要找的柱面上所需的时间

②在柱面上找到要找的磁道所需的时间

③在磁道上找到要找到的扇区所需的时间

④在扇区中找到要找的数据所需的时间

B:①可移动性 ②传送速率 ③奇偶校验能力 ④存储容量

C:①只能配一支绘图笔. ②绘图笔沿两条坐标轴运动.

③绘图笔沿一条坐标轴运动;图纸沿另一条坐标轴运动

④图纸沿两条坐标轴运动.

D:①光纤、聚焦、折射 ②曝光、显影、定影

③光笔、点阵、扫描 ④光栅、映像、合成

E：①读卡机 ②光扫描器 ③光符阅读器 ④磁条阅读器

[解析]

1． 查找时间（Seek Time）就是磁头沿切线方向移动到要找的磁道或柱面所需的时间，在磁道上找到查找目的扇区所需的时间是等待时间（Latency Time）.

2. 光盘可以极大地提高存储容量。但并不一定能极大地提高可移动性、传送速率及奇偶校验能力。

3. 滚筒式绘图机的绘图笔沿着滚筒的轴向作横向运动；图纸随着滚筒滚动，相对于绘图笔作纵向运动，两个方向运动的合成使绘图笔在图纸上画出的需的各种图形。这种绘图机通常也配有多支绘图笔。

4. 激光打印机的工作原理：计算机输出的信息控制激光图像发生器，形成激光图像照射到感光鼓鼓面上。鼓面上未被光照部分保留原有充电电荷，而被光照射的部分原充电电荷消失，这样使鼓面曝光，形成静电潜像，已有潜像的鼓面转入显像器时，由于显像器内有色粉，能靠静电吸引力吸附在已曝光的鼓面上，将静电潜像显影为色粉图像。然后，再将色粉图像转印到记录纸面上，通过加热使色粉熔化定影在纸面上，形成印刷拷贝。因此，供 D 选择的答案②中的曝光、显影和定影都是和激光打印机有关的概念。

5. 光扫描器是阅读条形码原硬件设备,读卡要是用来穿孔卡上的信息读入计算机.光符阅读器是读入字符的一种计算机输入设备.

[答案]A: ① B:④ C:③ D:② E:②

试题 12（1990 年试题 8）

从供选择的答案中选出应填入下面叙述中的{ }内的正确答案,把编号写在答卷的对应栏内.

用二进制加法器对二-十进制编码的十进制求和,当和的 4 位二-十进制编码(相当于一位十进制数)小于等于 1001(相当十进制数 9)且向高位无进位进, A ;当和小于等于 1001 且向高位有进位时, B ;当和大于 1001 时, C .

按照国家《信息交换用汉字编码字符集--基本集》（即 GB2312）规定，一个汉字由 D 个字

节组成。为了达到中西兼容的目的，区分汉字与 ASCII 码，汉字编码的最高位为 E 。
供选择答案
A、B、C：①不需修正 ②必须进行减 6 修正
③必进行加 6 修正 ④修正方法不确定
D、E：①0 ②1 ③2 ④2.5 ⑤3 ⑥4
[解析]
用二进制加法器对于二-十进制编码的十进制求和，当和小于等于 1001（十进制数的 9）且向高位无进位时，不需修正，例如：

```
   0011          3
+) 0011       +) 3
------        ----
   0110          6
```
对应于十进制运算

当和不于、等于 1001 且向高位有进位时，则必须进行修正即再加上 6，我们称为"加 6"修正，才会得到正确的结果。例如：

```
   1000
+) 1000
------
  10000          8
+) 0110       +) 8
------        ----
  10110          16
```
对应于十进制运算

当和大于 1001 时，也要进行加 6 修正。例如：

```
   0110
+) 0110
------
   1100          6
+) 0110       +) 6
------        ----
  10010          12
```
对应于十进制运算

根据国家标准 GB2312 规定，在《信息交换用汉字编码字符集--基本集》中，一个信息交换用汉字由 2 个字节组成，为了达到中西文兼容的目的，区分汉字与 ASCII 码，规定汉字编码的最高位为 1。
[答案]A：① B：③ C：③ D：③ E：②

2.2 存储器系统

2.2.1 主要知识点

掌握各类存储器的功能、特性和使用，重点把握虚拟存储器、Cache（高速缓冲存储器）和多级存储器的有关内容。

2.1.1.1 RAID 存储器

廉价磁盘冗余阵列 RAID，用多个较小的磁盘驱动器替换单一的大容量磁盘驱动器，同时合理地在多个磁盘上分布存放数据以支持同时从多个磁盘进行读写，从而改善了系统的 I/O 性能。RAID 机制中共分 6 个级别，工业界公认的标准分别为 RAID0～RAID5，其共同特征是：①RAID 由若干个物理驱动器组成，但对操作系统而言仍是一个逻辑驱动器；②数

据分布在阵列中多个驱动器上；③冗余的磁盘容量用以保存奇偶信息，以便在磁盘失效时进行恢复（只有 RAID0 不支持该特征）。

RAID 应用的主要技术有：①分块技术。对主机请求读定的数据进行分块，使之分布于多台磁盘上。②交叉技术。对存放在多台磁盘上数据的读写，采取交叉方式进行。③重聚技术。对多台磁盘上下班存储空间进行重新编址，使数据按重新编址后的空间进行存放。

RAID0 级（无冗余和无校验的数据分块）：具有最高的磁盘空间利用率，易管理，但系统的故障率高，属于非冗余系统。

RADI1 级（磁盘镜像阵列）：由磁盘对组成，每一个工作盘都有其对应的镜像盘，上面保存着与工作盘完全相同的数据拷贝，具有最高的安全性，但磁盘空间利用率只有 50%。RAID2 级（采用纠错海明码的磁盘阵列）：采用了海明码纠错技术，用户需增加校验盘来提供单纠错和双验错功能。对数据的访问涉及到阵列中的每一个盘。大量数据传输出时 I/O 性能较高，但不利于小批量数据传输，实际应用中很少使用。

RAID3 和 RAID4 级（采用奇偶校验码的磁盘阵列）：把奇偶校验码存入在一个别独立的校验盘上。如果一个盘失效，其上的数据可以通过对其他盘上的数据进行异或运算得到。读者数据很快，但因为写入数据时要计算校验收位，速度较慢。

RAID5（无独立校验盘时的奇偶校验磁盘阵列）：与 RAID4 类似，但没有独立的校验盘，校验信息分布在组内所有盘上，对于大、小批量数据读写性能都很好。RAID4 和 RAID5 使用了独立存取技术，阵列中每一个磁盘都相互独立地操作，所以 I/O 请求可以并行处理。

2.2.1.2 Cache 存储器

通常在 CPU 和主存储器之间设置小容量的高速存储器 Cache.Cache 容量小但速度愉，主存储器速度较低但容量大。，通过优化调度算法，系统的性能会大大改善，其存储系统容量与主存相当，而访问速度近似 Cache。在计算机的存储系统体系中，Cache 是访问速度最快的层次。

使用 Cache 改善系统改性能的依据是程序的局部性原理，即程序的地址访问流有很强的时序相关性，未来的访问模式与最近已发生的访问模式相似。依据局部性原理，把主存储器中访问概率高的内容存放在 Cache 中，当 CPU 需要读取数据时就首先在 Cache 中查找是否有所需内容。如果有则直接从 Cache 中读取；若没有再从主存中读取该数据，然后同时送往 CPU 和 Cache。

Cache 存储器的组织结构与主存储器不一样，它以行作为基本单元/每一行又分为标志和数据域两部分，数据域中存放着若干项数据，而标志项是是这一块数据的地址标识。

当 CPU 发出对存储器的读命令后，其访问地址先送给 Cache 控制器，Cache 检查其地址标识符目录以确定是否匹配项。若发现匹配项（命中），则根据其访问地址确定是对该行数据埠中的第几项进行读取，然后该项即进入 Cache 的数据寄存器。如果没有命中，则到主存器读取数据。

当 CPU 发出访存请求后，存储器地址先被送到 Cache 控制器以确定所需数据是否已在 Cache 中，若命中则直接对 Cache 进行访问，这个过程称为 Cache 的地址映射（Mapping）。常见映射方法有直接映射、相联映射和组成映射。

当 Cache 存储器产生了一次访问合职之后，相应的数据应同时读入 CPU 和 Cache。但是当 Cache。但是当 Cache 已存满数据后，新数据和须淘汰 Cache 的地址映射（Mapping）。常见的映射方法有直接映射、相当映射和组相联映射。

当 Cache 存储器产生了一次访问未命中之后，相应的数据应同时读入 CPU 和 Cache。但是当 Cache 已存满数据后，新数据必须淘汰 Cache 中的某些旧数据。最常用的淘汰算法有随机淘汰法、先进先出法（FIFO）和近期最少使用淘汰法（LRU）。

2.2.1.3 虚拟存储器

计算机里的程序和数据通常都存放在外存储器（辅助存储器）上，直到 CPU 需要的才调入到主存储器中。虚拟存储系统的作用是给程序员一个更大的"虚拟"的存储空间，其容量可远远超过主存储器的容量，而与辅助存储器容量相当。

在使用虚拟存储器体系的系统中，由程序（CPU）使用的访存地址称为虚拟地直，程序（CPU）直接访问的存储空间称为虚拟地址空间，而主存储器的地址则称为物理地。通常虚拟地直空间远远大小主存储器容量。

2.2.2 试题解析

在硬件基础知识部分，存储器系统始终是考查的重点。从历年试题统计（见表 3-3）来看，考查的内容主要集中在 Cache、虚拟存储器和 RAID 磁盘阵列。

表 3-3　历年高级程序员级存储器系统试题统计

试题	考查知识点
1992 年试题 10	Cache 存储器与虚拟存储器
1993 年试题 10	辅助存储器，涉及软盘、光盘和磁带机
1994 年试题 7	RAID 磁盘阵列
1996 年试题 8	Cache 存储器
1997 年试题 9	SCSI 接口，RAID 磁盘阵列
1998 年试题 8	Cache 存储器
1999 年试题 9	内存种类
1999 年试题 10	虚拟存储系统
2000 年试题 10	存储器有关的计算

试题 1 （2000 年试题 10）

从供选择的答案中选出应填入下面叙述中的{ }内的正确答案,把编号写在答卷的对应栏内.

假设某计算机具有 1MB 的内存（目前使用计算机往往具有 64MB 以上的内存），并按字节编址，为了能存取该内存各地址的内容，其地址寄存器至少需要二进制 A 位。为了使 4 字节组成的字能从存储器中一次读出，要求存放在存储器中的字边界对齐，一个字节的地址码应 B 。若存储器周期为 200ns，且每个周期可访问 4 个字节，则该存储器带宽为 C bit/s。假如程序员可用的存储空间为 4MB，则程序员所、用的地址为 D ，而真正访问内存的地址称为 E 。

供选择答案

A：①10 ②16 ③20 ④32

B：①最低两位为 00 ②最低两位为 10 ③最高两位为 00 ④最高两位为 10

C：①20M ②40M ③80M ④160M

D：①有效地址 ②程序地址 ③逻辑地址 ④物理地址

E：①指令地址 ②物理地址 ③内存地址 ④数据地址

[解析]

这是一道关于内存的计算题。

1M=2^{20}，故 1MB 内存按字节编址（即寻找空间为 1M），地址寄存器至少需要 20 位。

如果采用字节编址，4 字节一次读出，即字长为 32 位，每个字有 4 个单独编址的存储字节，字地址是该字高位字节的地址，总是等于 4 的倍数，正好用地址码的最低两位（为 0）来区分同一字的 4 个字节。

若存储周期为 200ns，每个周期可访问 4 个字节，其带宽为：

（$1/200\times10^{-9}$）×4=20×10^{6}（字节/秒）=160M（位/秒）

因为可用的 4M 内存空间超大型出了实际的物理内存 1M，称为逻辑地址，实际访问内存的地址为物理地址。
[答案]A：③ B：① C：④ D：③ E：②

试题 2 （1999 年试题 9）

从供选择的答案中选出应填入下面叙述中的{ }内的正确答案,把编号写在答卷的对应栏内.
用作 存储器的芯片有不同的类型。
可随机读写，且只要不断电则基本存储信息就可一直保存的，称为 A 。
可随机读写，但即便在不断电的情况下其存储的信息也要定时刷新才不致丢失失的，称为 B 。
所存信息由生产厂家用来掩膜技术写好后就无法再改变的称为 C 。
通过紫外线照射后可擦除所有信息，然后重新写入新的信息并要可多次进行的，称为 D 。
通过电信号可在数秒钟内快速删除全部信息，但不能进行字节级别删除操作的。称为 E 。
供选择答案
A、B：①RAM ②VRAM ③DRAM ④SRAM
C、D：①EPROM ②PROM ③ROM ④CDROM
E： ①E2PROM ②Flash Memory ③EPROM ④Virtul Memory
[解析]
本题考查计算机存储器知识，比较容易。存储器是用来存放程序如数据的，因此对它的要求是既能写入又能读入，这样存储器称为随机存储器（RAM）。目前计算机中广泛采用的是金属氧化物半导体（MOS）存储器，随机存储器在停电的情况下会丢失所存储的信息。
随机存储器分为静态与动态两种。静态存储器（SRAM）的内容，在不停电的情况下通史长时间保留不变；动态存储器（DRAM）的内容光焕发，即使不停电的情况下，隔一定时间以后也会自动消失，因此在消失前要根据原来内容重新写一遍，这就是再生和刷新，一般以 2ms 为周期进行刷新。
另外，根据是否可以写入内容，又可以分为随机存储器（RAM）和只读存储器（ROM）。通常工厂提供的只读存储器（掩膜 ROM）在出厂时已经写好内容，还有一种只读存储器，在出厂时芯片内没有内容。允许用户使用特殊的仪器写入内容，然后安装到计算机上，但在计算机运行过程中却不允许写入内容，这叫做可编程的只读存储器（PROM）。
在可编程只读存储器中，有些是在写入内容后就无法改变其内容的，而有些却是可以通过特殊的仪器将原先的内容擦掉再重新写入的,这种 PROM 叫做可擦除只读存储器，（EPROM），在 EPROM 中，有一种可以通过电来擦除内容的 EPROM，称为电擦除可编程只读存储器（EEPROM 或者 E2PROM）。
至于 Flash Memory,通常称为闪存，通过电信号可以在数秒钟内快速删除全部信息，但不能进行字节级别删除操作。
[答案]A：④ B：③ C：③ D：① E：②

试题 3 （1999 试题 10）

从供选择的答案中选出应填入下面叙述中的{ }内的正确答案,把编号写在答卷的对应栏内.
虚拟存储器的作用是允许 A 。它通常使用 B 作为它一个主要组成部分。对它的调度方法与 C 基本相似，即把要经常访问的数据驻留在高速存储器中，因为使用虚拟存储器，指令招待时 D 。在虚拟存储系统中常使用相联存储器进行管理，它是 E 寻址的。
供选择的答案
A：①直接使用外存替内存

②添加此地址字长允许的更多内存容量
③程序直接访问比内存更大的地址空间
④提高内存的访问速度
B：①CDROM ②硬盘 ③软盘 ④寄存器
C：①Cache ②DMA ③I/O ④中断
D：①所需的数据一定能在内存中找到 ②必须先进行"虚"、"实"地址交换
③必须事先使用覆盖技术 ④必须将常用子程序先调入内存
E：①按地址 ②按内存 ③寄存器 ④计算
[解析]
本题主要考查虚拟存储系统。在同一个作业内部，作业进程能够得到的存储空间会小于某些作业的地址空间，为使这样的作业也能够在系统中运行，计算机系统使用了虚拟存储技术。所谓虚拟存储技术，即在内存在中保留作业/进程的一部分程序或数据，在外存中放置整个地直空间的副本。作业运行过程中可以随机访问内存中的数据或程序，但需要的程序或数据不在内存时，就将内存中部分内容根据情况写回外存，然后从外存调入所需程序或数据，实现作业内部的局部对换技术，从而允许作业的地址宽间大于实际分配的存储区域。
虚拟存储技术可以是基于手段式的，可以基于页式的，也可以是基于段页式的。影响虚存性能的关键是命中率，即一次操作中其对象在内存的概率。虚拟存储之所以在行，关键在于程序具有局部性。程序局部性有两个方面的含义：(1) 时间局部性，如果一条指令被执行，则在不久以后可能再次被执行；(2) 空间局部性，一段时间里程序所使用的操作地址相对集中在较小的范围内。
虚拟存储技术常用的页面调度技术是请求式页面调度，即除了在页表中存放逻辑页呈与物理页号的对应关系等数据外，还要标训该页是否在内存。当一条指令的操作对象所在页不在内存时，发出缺页中断，转入操作系统处理。这时系统按照某种淘汰算法挑选某一物理页，根据情况决定是否将其写回外存，然后从外存在中调入相应页面覆盖之，修改页表，缺页中断逻辑地址，而内存中存储的数据和程序是以物理地址来表示的，这样，在执行一条指令时，必须先进行从逻辑地址的交换，即地址映射。
Cache 即高速缓冲存储器。内部 Cache 对程序操作是透明的，而且不需要使用外部总线。通常,Cache 中存放一部分内存的内容，当执行读操作时，首先查找地址是否存在于 Cache 中，如果存在，就立即从 Cache 读取，否则从内存中读取，同时将内存中该部分内容写入 Cache，以便今后可以直接读取 Cache。这与虚拟存储技术的页面调度方式相似。
[答案]A：③ B：② C：① D：② E：③

试题 4 （1998 年试题 8）

从供选择的答案中选出应填入下面叙述中的{ }内的正确答案,把编号写在答卷的对应栏内.
设有 3 个指令系统相同的处理机 X、Y 和 Z，它们都有 4kB 的高速缓冲存储器（Cache）T 和 32MB 的内容，但是其存取周期都不一样，见表 3-4（Tic 和 Tim 分别表示 I 处理机 Cache 存取周期和主存存取周期）。

表 3-4　处理机存取周期

处理机 / 存取周期	X	Y	Z
Tic	40ns	100ns	120ns
Tim	1μs	0.9μs	0.8μs

若某段程序，所需指令或数据在 Cache 中取到的概率为 P=0.5，则处理机 X 的存储器平均存取周期为 A us。假定指令招待时间与存储器的平均存取周期成正比，则此时 3 个处理机执

行该段程序由快到慢的顺序为 B 。
若 P=0.65 时，则顺序为 C。
若 P=0.8 时，则顺序为 D。
若 P=0.85 时，则顺序为 E。
供选择的答案
A：①0.2 ②0.48 ③0.52 ④0.6
B～E:①X、Y、Z ②X、Z、Y ③Y、X、Z ④Y、Z、X
⑤Z、X、Y ⑥Z、Y、X
[解析]
在高速缓冲存储器（Cache）与主存构成的存储体系中，存储器的平均存取周期为 T=Tic*P+(1-P)*Tim，其中 Tic 为 Cache 的存取周期，Tim 为主存存取周期，P 为 Cache 的命中率，即指令或数据在 Cache 中取到的概率。具体计算见表 3-5。

表 3-5　　1998 年试题 8　　单位：μs

P / 处理机	X	Y	Z	快慢次序
0.5	0.52	0.50	0.46	Z、Y、X
0.65	0.376	0.38	0.358	Z、X、Y
0.8	0.232	0.26	0.256	X、Z、Y
0.85	0.184	0.22	0.222	X、Y、Z

[答案]A：③ B：⑥ C：⑤ D：② E：①

试题 5 （1997 年试题 9）

从供选择的答案中选出应填入下面叙述中的{ }内的正确答案,把编号写在答卷的对应栏内.
SCSI 是一种通用的系统级标准输入/输出接口，其中 A 标准的数据宽度为 16 位，数据传送率达 20MB/s。
大容量的辅助存储器常采用 RAID 磁盘阵列。RAID 的工业标准共有 6 级。其中 B 是镜像磁盘阵列，具有最高的安全性； C 是无独立校验盘的奇偶校验码磁盘阵列； D 是采用纠错海明码的磁盘阵列；E 则是既无冗余也无校验的磁盘阵列，它采用了数据分块技术，具有最高的 I/O 性能和磁盘空间利用率，比较容易管理，但没有容错能力。
供选择的答案
A：①SCSI-I ②SCSI-II ③FAST SCSI-II ④FAST/WIDE SCSII-II
B～E：①RAID0 ②RAID1 ③RAID2 ④RAID3 ⑤RAID4 ⑥RAID5
[解析]
SCSI（Small Computer System Interface，小型计算机系统接口）标准的性能见表 3-6。

表 3-6　　SCSI 接口性能参数

名称	数据宽度	数据传输率
SCSI-I	8 位	5Mbit/s
SCSI-II	16 位	10Mbit/s
FAST SCSI-II	16 位	20Mbit/s
FAST/WIDE SCSI-II	32 位	40Mbit/s

因此数据宽度为 16 位，传输率为 20MB/s 的输入/输出接口为 FASTSCSI-II。
RAID（Redundent Array of Inexpensive Disks）磁盘阵列的工业标准由 RAID0～RAID5,共 6 级，其特点如下：
RAID0 无冗余和无效验的数据分块磁盘阵列;
RAID1 镜像磁盘阵列，具有最高安全性；

RAID2 采用纠错海明码的磁盘阵列；
RAID3 和 RAID4 采用奇偶检验码的磁盘阵列（有独立校验盘）；
RAID5 无独立校验盘的奇偶检验码磁盘阵列。
[答案]A：③ B：② C：⑥ D：③ E：①

试题 6 （1996 年试题 9）

从供选择的答案中选出应填入下面叙述中的{ }内的正确答案,把编号写在答卷的对应栏内.
在多级存储系统中，Cache 处在 CPU 和主存之间，解决 A 问题。若 Cache 和主存的存取时间分别 T1 和 T2，Cache 的命中率为 H，则计算机实际存取时间为 B 。当 CPU 向存储器执行读操作时，首先访问 Cache，如命中，则从 Cache 中取出指令或数据，否则从主存中取出，送 C ；当 CPU 向存储器执行写操作时，为了使 Cache 内容和主存的内容保持一致，若采用 D 法，同时写入 Cache 和主存。由于 Cache 和主存。由于 Cache 容量比主存容量小，当 Cache 已写满，但要把主存信息写入 Cache 和主存。由于 Cache 容量比主存容量小，当 Cache 已满，但要主存信息写入 Cache 时，就要淘汰 Cache 中已有信息。为了提高 Cache 的命中率，常采用一种 E 替换算法。
供选择的答案
A：①主存容量扩充 ②主存和 CPU 速度匹配 ③多个请求源访问主存在 ④BIOS 存放
B:①HT1+T2 ②(1-H)T1+HT2 ③T2-HT1 ④HT1+(1-H)T2
C:①Cache ②CPU ③Cache 和 CPU ④Cache 或 CPU
D:①写回 ②写通 ③映照 ④特征
E;①LRU ②FIFO ③FILO ④RANDOM
[解析]
计算机存储体系一般分成 3 个层次:Cache、为第 1 层次，存取速度最快，但存储容量最小，价格也最高；辅助存储器容量最大，速度最低，价格也最便宜，处于第 3 层次；主存储器居中，无论从存取速度、容量或价格等方面考虑均处于两者之间。
由于集中电路的工艺的发展，CPU 的速度越来越快，而相对来说，存储器速度跟随不上 CPU 的发展，因此阻碍了计算机整体性能的提高，为了解决主存和 CPU 速度匹配问题，在 CPU 与主存储器之间设置了 Cache。Cache 的速度是主存的数倍，基本上与 CPU 匹配。因此本题的答案 A 应选择②。
CPU 取指令或取数据时，首先检查要访问单元的内容是否已在 Cache 中，若是，则从 Cache 中读取；若不是，则到主存中读取，并将取出的内容同时送往 CPU 与 Cache 和从主存取读出操作后，能保证 Cache 和主存内容的一致性。Cache 的命中率为 H，即从 Cache 从主存读取内容的概率分别为 H 和 1-H，由此计算出平均存取时间为 HT1+（1-H）T2。所以 B 选④,C 选③.
Cache 中的内容只是主存相应部分的一个副本，当 CPU 该部分的某个单元执行写操作时，为了使 Cache 内容与主存的内容保持一致，通常采用两种办法，即"写回"和"写通"。如果在执行写操作时，数据同时写入 Cache 和主存称为写通（Write-through），该方法操作简单，可随时保持 Cache 和主存数据的一致性，但可能增加多次不必要的写入。如要向 Cache 某一单元写入多次，也要向主存相应单元写入多次，另一办法称之为"写回"（Write-back），执行写操作时一般只写入 Cache，只有当该 Cache 单元被替换(或多道程序转换)时,才将修改过的内容写回主存,这种方式写入极快.因此本题答案 D 应选择②"写通"
在供选择的答案 E 中 LRU(Least Recently Used)为"近期最少使用"算法，根据页面调入的先后来淘汰页面；FIFO（First In First Out） 为"先进先出"算法，根据页面调入的先后淘汰页面，但页面调入的先后并不能反映页面的使用情况；FILO（First In Last Out）是"先进后出"

算法；RANDOM 为"随机"算法。以上都是 Cache（或虚拟存储器）的替换算法。常用的是 LRU 和 FIFO 两种算法。尤其是 LRU，通常可取得比 FIFO 更高的命中率，所以 E 选①。
[答案]A：② B：④ C：③ D：② E：①

试题 7 （1994 年试题 7）

从供选择的答案中选出应填入下面叙述中的{ }内的正确答案,把编号写在答卷的对应栏内.
RAID 是一种经济的磁盘冗余阵列,它采用 A 和 B 以提高数据传输率.RAID 与主机连接较普遍使用工业标准接口为 C .
假脱机（Spooling）打印与脱机打印机有相似之处，但实际上其输出结果首先送往 D 保存，然后再在适当的时候将其调出打印出来。整个过程是由 E 控制的。
供选择的答案
A、 B：①智能控制器 ②磁盘镜像 ③磁盘双工技术 ④多磁盘驱动器
C：①RS-232 ②FDDI ③SCSI ④ST506
D、E：①主存储器 ②设备控制器 ③操作系统 ④外部存储器
⑤打印机缓冲存储器 ⑥编译程序
[解析]
RAID（Redundant Arrays of Inexpensive Disks）是一种经济的磁盘冗余阵列，采用智能控制器和多磁盘控制器以提高数据传输率。RAID 与主要连接较普遍使用的工业标准接口为 SCSI。目前，RAID 的产品多是由许多"独立"的磁盘构成冗余阵列，以提高其性能和可靠性。为了提高数据传输率，RAID 采用多磁盘驱动器和智能控制器，使得阵列中的各个磁盘可独立并行工作，减少了数据存取等待时间。
假脱机（Spooling）打印与脱机打印有相似之处，实际上其输出结果首先送往外部存储器保存，然后再在适当的时候将其调出打印。整个过程是由操作系统控制的。
RS-232 是用来连接终端或调制解调器的串行接口；FDDI（光纤分布式数据接口）是一种光纤环形网络的名称；ST506 则是个人计算机上用来连接温彻斯特硬盘的较低速的接口，不能用来连接 RAID。
[答案]A：① B：④ C：③ D：④ E：③

试题 8 （1993 年试题 10）

从供选择的答案中选出应填入下面叙述中的{ }内的正确答案,把编号写在答卷的对应栏内.
软盘使用前先要进行格式化（FORMAT），是为了确定 A 。光盘存储器利用光束在记录表面读写信息，使用的是 B 。光盘当前受到用户重视是因其有下述突出优点：C 。一台高性能 9 磁道、半英寸（1 英寸=25.4mm）的磁带机，记录密度是 256B/mm,带速 4m/s ,启停时间 10ms，带长 900m，以 EBCDIC 码按块记录文件，每个记录长为 128B，块间间隔 10mm，块化系数为 16，那么整盘带的记录容量最接近 D 字节（取整数），从磁带上读出全部记录，需要 E ms 时间。
供选择答案
A：①磁道数 ②道密度 ③编码格式 ④记录格式
B：①红外光 ②紫外光 ③激光 ④可见光
C：①容量大、可靠性高、便于携带 ②容量大、可靠性高、高速
③容量大、便携带、可反复读写 ④可靠性高、高速、可反复读写
D：①160M ②400M ③100M ④60M
E：①125010 ②125000 ③215000 ④225010
[解析]

一个计算机系统上使用一张软盘，必须首先将其格式化成该系统所能识别的记录格式。为了确定软盘的记录格式，软盘使用前先要进行格式化。一张软盘片上有多少磁道，磁道密度有多少，有统一的国际标准，因而一张空白的软盘可以用在任何计算机系统上。但是软盘。不同容量的软盘，其记录格式也不一定相同。同一计算机系统中，也支持不同容量的软盘。不同容量的软盘，其记录格式也不同，如 720KB 的软盘与 1.44MB 的软盘，其磁道数就不同。光盘存储器是租用激光束照射到记录表面，利用热作用和介质表面熔化以改变其光学反射特性或改变介质表面局部磁场方向来记录信息，而后再利用激光束照射到记录介质上，根据反向光的强弱来读出信息的。光盘因其容量大（一般容量为 650MB）、可靠性高和能进行高速存取等优点。软盘虽然容量小（常用的 3.5 英寸软盘容量为 1.44MB）、速度慢慢，但有可反复读写又极易携带的优势，因此使用方便。

磁带一般是按块存储数据的，每块中又有若干记录，每块中的记录数就称为块化系数。块与块间有块间间隔。

带记录容量

=带块数*块记录数据长度*记录密度

=(带长/(块记录长度+块间间隙))×块化系数×(记录字节数/记录密度)×记录密度

=(900m/(8mm+10mm))×16×(128Byte/(256Byte/mm))×256Byte/mm

$=5\times10^4\times16\times128Byte$

$=1024\times10^5Byte$

=100MB

从磁带上读出全部记录所需时间：

所需时间

=启停时间+带长/带速

=10ms+900m/4m/s

=225010ms

[答案]A：④ B：③ C：② D：③ E：④

试题 9 （1992 年试题 10）

从供选择的答案中选出应填入下面叙述中的{ }内的正确答案,把编号写在答卷的对应栏内.

Cache 介于 A 之间，由 B 完成信息动态调度，目的是使用 C ；虚拟存储器是为了使用权用户运作比主存在容量大得多的程序，它要在 D 之间进行住处动态调度，这种调度是由 E 来完成的。

供选择答案

A、D：①CPU 和 I/O BUS ②地址寄存器和数据寄存器

③CPU 和主存 ④ 双机系统

C：①打印信息不丢失 ②主存和 CPU 速度匹配

③显示器分辨率提高 ④汉字功能增强

B、E：①软件 ②硬件 ③操作系统和硬件 ④固件

⑤BIOS ⑥操作系统

[解析]

在多次存储体系中，高速缓冲存储器 Cache 外在 CPU 和主存储器之间，其容量比主存小得多，但存取速度却比主存快得多。Cache 中的内容是主存中部分内容的拷贝，当 CPU 需访问这部分内容时就可以访问 Cache，而不必访问主存，此时称为命中；若需访问的内容不在 Cache 中（即不命中），则需从主存储器取出。Cache 和主存中信息的映象关系以及它们之间的住处调度是由硬件来实现的。Cache 的存在对编程者来说是透明的。采用 Cache 后，只要

其容量足够大，并有恰当的调度算法，利用程序和数据的局部性可以获得相当高的命中率。这样就使得用起来等价于有一个速度与 Cache 相当，比主存更容易与 CPU 速度匹配，容量又与主存相当的存储器。

虚拟存储器允许用户用比主存容量大得多的地址空间来编程，以运行比主存实际容量大得多的程序。用户编程所用的地址称为逻辑地址（又称虚地址），而实际的主存地址则称为物理地址（又称实地址）。每次访问内存时都要进行逻辑地址到物理地址的转换。实际上，超过主存在实际容量的那些程序和数据是存放在辅助存储器中，当使用时再由辅存调入。地址变换以及主存和辅存间的信息动态调度是硬件和操作系统两者配合完成的。

[答案]A：③ B：② C：② D：⑤ E：③

2.3 安全性、可靠性和性能评价

2.3.1 主要知识点

了解计算机数据安全和保密、计算机故障诊断与容错技术、系统性能评价方面的知识，掌握数据加密的有关算法、系统可靠性指标和可靠性模型以及相关的计算方示。

2.3.1.1 数据的安全与保密

（1） 数据的安全与保密

数据加密是对明文（未经加密的数据）按照某种加密算法（数据的变换算法）进行处理，而形成难以理解的密文（经加密后的数据）。即使是密文被截获，截获方也无法或难以解码，从而阴谋诡计止泄露信息。数据加密和数据解密是一对可逆的过程。数据加密技术的关键在于密角的管理和加密/解密算法。加密和解密算法的设计通常需要满足 3 个条件：可逆性、密钥安全和数据安全。

（2） 密钥体制

按照加密密钥 K1 和解密密钥 K2 的异同，有两种密钥体制。

① 秘密密钥加密体制（K1=K2）

加密和解密采用相同的密钥，因而又称为密码体制。因为其加密速度快，通常用来加密大批量的数据。典型的方法有日本的快速数据加密标准（FEAL）、瑞士的国际数据加密算法（IDEA）和美国的数据加密标准（DES）。

② 公开密钥加密体制（K1≠K2）

又称不对称密码体制，加密和解密使用不同的密钥，其中一个密钥是公开的，另一个密钥是保密的。由于加密速度较慢，所以往往用在少量数据的通信中，典型的公开密钥加密方法有 RSA 和 ESIGN。

一般 DES 算法的密钥长度为 56 位，RSA 算法的密钥长度为 512 位。

（3） 数据完整性

数据完整性保护是在数据中加入一定的冗余信息，从而能发现对数据的修改、增加或删除。数字签名利用密码技术进行，其安全性取决于密码体制的安全程度。现在已经出现很多

使用 RSA 和 ESIGN 算法实现的数字签名系统。数字签名的目的是保证在真实的发送方与真实的接收方之间传送真实的信息。

（4） 密钥管理

数据加密的安全性在很大程度上取决于密钥的安全性。密钥的管理包括密钥体制的选择、密钥的分发、现场密钥保护以及密钥的销毁。

（5） 磁介质上的数据加密

常用的方法有：硬加密的防复制技术、软加密的防解读技术和防跟踪技术。硬加密技术常用的 3 种方式是：①利用非标准格式的磁介质记录方式；②激光加密技术；③利用专用的硬件。

2．3．1．2 计算机系统可靠性

计算机系统的可靠性是指从它开始运行（t=0）到某时刻 t 这段时间内能正常运行的概率，用 R（t）表示。所谓失效率是指单位时间内失效的元件数与元件总数的比例，以表示，当为常数时，可靠性与失效率的关系为：

$R(t)=e^{-\lambda t}$

两次故障之间系统能正常工作的时间的平均值称为平均无故障时间 MTBF：

$MTBF=1/\lambda$

通常用平均修复时间（MTRF）来表示计算机的可维修性，即计算机的维修效率，平均修复时间指从故障发生到机器修复平均所需要的时间。计算机的可用性是指计算机的使用效率，它以系统在执行任务的任意时刻能正常工作的概率 A 来表示。

$A=MTBF/(MTBF+MTRF)$

计算机的 RAS 技术，就是指用可靠性 R、可用性 A 和可维修性 S 这 3 个指标衡量一个计算机系统。但实际应用中，引起计算机故障的原因除了元器件以外还与组装工艺、逻计算机可靠性模型有关。

常见的系统可靠性数学模型有以下 3 种：

① 串联系统。假设一个系统由 N 个子系统组成，当且仅当所有的子系统都能正常工作时，系统才能正常工作，这种系统称为串联系统。

设各子系统的可靠性为 R1、R2、…Rn，则整个串联系统的可靠性为：

$R=R1\times R2\times\ldots\times Rn$

设各子系统的失效率为 ，则整个串联系统的失效率为：

$\lambda=\lambda1+\lambda2+\ldots+\lambda n$

② 并联系统。假如一个系统由 N 个子系统组成，只要有一个子系统正常工作，系统就能正常工作，这样的系统称为并联系统。

设各子系统的可靠性为 R1、R2、…、Rn，则整个并联系统的可靠性为：

$R=1-(1-R1)(1-R2)\ldots(1-Rn)$

设各子系统的失效率为 ，则整个并联系统的失效率为：

$$\mu=\frac{1}{\frac{1}{\lambda}\sum_{j=1}^{n}\frac{1}{j}}$$

③ N 模冗余系统。N 模冗余系统由 N 个（N=2n+）相同的子系统和一个表决器组成，

表决器把 N 个子系统中占多数的相同结果的输出作为系统的输出。

设各子系统的可靠性均为 R_0，则整个 N 模冗余系统的可靠性为：

$$R=\sum_{i=N+1}^{N}\binom{j}{N}*R_0^i(1-R_0)^{N-i}$$

（其中 $\binom{j}{N}$ 表示从 N 个元素中取 i 个元素的组合数）

2．3．1．3 计算机系统的性能评价

性能评测的常用方法：①时钟频率；②指令执行速度；③等效指令速度法；④数据处理速率 PDR 法；⑤核心程序法。

基准程序法（Benchmark）是目前一致承认的测试性能的较好方法，有多种多样的基准程序，如主要测试整数性能的基准程序、测试浮点性能的基准程序等。

2．3．1．4 计算机故障诊断和容错

计算机的故障根据其表现出的特点，可以分为永久性故障、间歇性故障及瞬时性故障 3 类。故障诊断包括故障检测定位两个方面。

容错是采用冗余方法来消除故障影响。针对硬件，有时间冗余两种方法。主要容错技术有简单的双机备份和操作系统支持的双机容错。

2.3.2 试题解析

从历年安全性和可靠性方面的试题统计（见表 3-7）来看，主要考查系统可靠性，涉及计算机可靠性模型及相关的计算，有时与其他硬件知识类试题结合起来考查，此类试题看似复杂，其实只要复习一下相关内容，解答起来比较简单。

表 3-7　历年高级程序员级安全性与可靠性试题统计

试题	考查知识点
1990 年试题 9	涉及计算机可靠性模型，主要考查 N 模冗余系统
1991 年试题 10	系统可靠性计算
1992 年试题 8	系统可靠性计算
1996 年试题 9	系统可靠性计算
1996 年试题 11	数据加密与解密
2000 年试题 12	系统可靠性计算，海明码

试题 1（2000 年试题 12）

从供选择的答案中选出应填入下面叙述中的{ }内的正确答案,把编号写在答卷的对应栏内.

为提高数据传输的可靠性，可采用"冗余校验"的方法，海明码是常用方法之一。在此方法中，若要求能校检测出所有双位错觉，并能校正单位错，则合法码字集中的码距至少为 A 。若原始数据的字长为 5 位，则采用海明码对其样验位至少为 B 位。

对下面图 3-2（a）所示系统，仅当部件 1、部件 2 和部件 3 全部正常时系统才能政党工作，图中数字为各部分的可靠性，整个系统的可靠性近似为 C 。如果将部件 2 和部件 3 改成由两个器件构成，如图 3-2（b）所示，只要器件 a 和 b 中有一个正常就能使部件 2 正常工人，只要器件 c 和 d 中有一个正常就能使部件 3 正常工作。图中数字是各器件可靠性，则部件 2 的可靠性是 D ，整个系统的可靠性近似为 E 。

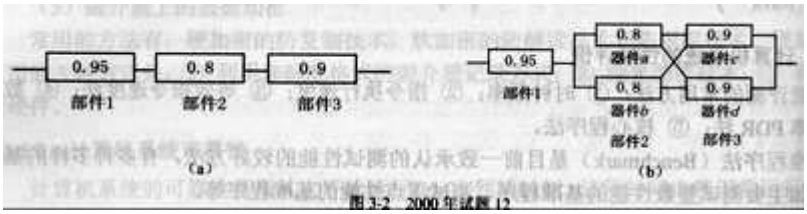


图 3-2　2000 年试题 12

供选择的答案

A：①1 ②2 ③3 ④4

B：①1 ②2 ③3 ④4

C：①0.68 ②0.72 ③0.80 ④0.92

D：①0.64 ②0.88 ③0.96 ④0.99

E：①0.82 ②0.90 ③0.94 ④0.96

[解析]

这是一道考查可靠性的综合题，除了可靠性计算之外，还涉及用于检错校验的海明码。

关于海明码问题的解答请参见 3。1。1。2 节的试题 9（1999 年试题 8）

图 3-2（ a）所示系统符合串联、系统可靠性模型，根据系统可靠性公式，求得其可靠性为：

R=R1×R2×R3=0.95×0.8×0.9≈0.68

图 3-2（b）所示系统是一个由串联和并联组合成的可靠性模型，其中部件 2 由两个并联的器件 a 和 b 构成，其可靠性为：

R2=1-（1-Ra）×（1-Rb）=1-（1-0.8）×(1-0.8)=0.96

最后与部件 1 一起计算整个系统（串联模型）的可靠性为：

R=R1×R2×R3=0.95×0.96×0.99≈0.90

[答案]A：③ B：④ C：① D：③ E：②

试题 2（1996 年试题 9）

从供选择的答案中选出应填入下面叙述中的{ }内的正确答案,把编号写在答卷的对应栏内.

设在图 3-3 和图 3-4 系统中，R1、R2、R3 为 3 个加工部件，每个加工部件的失效率均为 λ，可靠性均为 R。则图 3-3 系统的失效率为 A ，可靠性为 B 。图 3-4 中系统的失效率为 C ，可靠性为 D 。若每个加工部件的平均无故障时间为 5000 小时，则图 3-4 中系统的平均无故障时间为 E 小时。

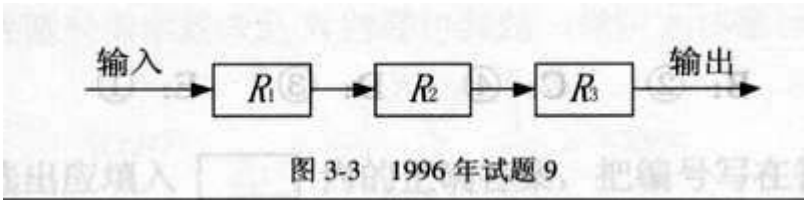


图 3-3　1996 年试题 9

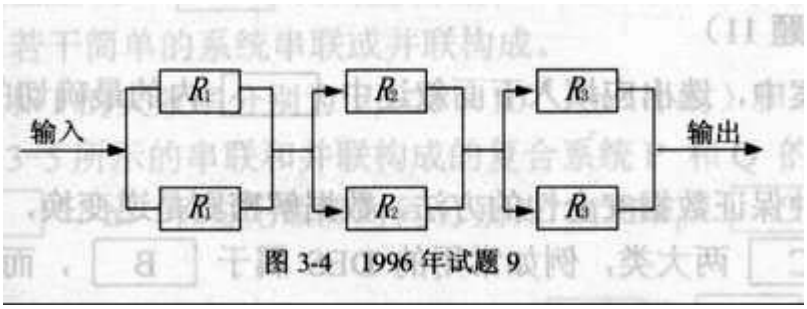


图 3-4　1996 年试题 9

供选择答案

A：①λ/3 ②λ^3 ③3λ ④1-λ^3

B：①R/3 ②R^3 ③3R ④1-R^3
C：①(3/2)λ ②(2/3)λ ③(6/11)λ④2λ
D：①（1-R^2）^{3} ②3（1-R^2） ③R^3（2-R）^{3} ④1-3（1-R^2）
E：①2500 ②5000 ③7500 ④3333
[解析]
在任意控制系统中，已知每个部件的失效率为 λ，可靠性为 Ri，对串联系统而言，系统的失效率为 λ1+λ2+λ3=3λ，问题 A 选③；系统的可靠性为 R1·R2·R3=R^3，问题 B 选②。
对并联系统（有 n 个部件，每个部件的失效率均为 λ）而言：
系统的失效率=λ/(1+1/2…+1/n); 系统的可靠性为=1-(1-R)^n。
图 3-4 所示乃串并联系统的总失效率为：
λ 系=λ 并 1+λ 并 2+λ 并 3
=λ/（1+1/2）+λ/（1+1/2）+λ/（1+1/2）=2λ，所以问题 C 选择④。
一个并联部分的可靠性=1-（1-R）^{2}=（2-R）R；
整个系统可靠性=[（2-R）R][（2-R）R]=[（2-R）R]^3=（2-R）$^3R^3$，所以问题 D 的答案为③。
由于系统的总失效率 λ 系为 2λ，且每个部件的平均无故障时间=1/λ=5000h，所以系统的平均无故障时间=1/λ 系=1/2λ=1/2·1/λ=2500h，问题 E 的正确答案为①。
[答案]A：③ B：② C：④ D：③ E：①

试题 3（1996 年试题 11）

从供选择的答案中选出应填入下面叙述中的{ }内的正确答案,把编号写在答卷的对应栏内.
数据加密是一种保证数据安全性的方法，数据解密则是逆变换，即 A 。密码体制可分为 B 和 C 两大类，例如常用的 DES 属于 B ，而 RSA 则属于 C 。DES 的密钥长度为 D 位。破密都面临多种不同的问题，其从易到难排列依次为 E 。
供选择的答案
A：①由加密密钥求出解密密钥 ②由密文求出明文
③由明文求出密文 ④由解密密钥求出加密密钥
B、C：①公开密钥 ②替代密码 ③换位密码 ④对称密钥
D：①32 ②48 ③64 ④128
E：①选择明文、已知明文、仅知密文 ②已知明文、仅知密文、选择明文
③已知明文、选择明文、仅知密文 ④仅知密文、已知密文、选择明文
[解析]
数据加密是利用加密密钥加密算法将明文（数据）转换成密文，而数据解密是利用解密密钥将密文变换成明文。所以问题 A 的答案为②。
密码体制按其对称性可分为对称密钥体制与非对称密钥体制两大类。在对称密钥体制中，加密算法之间存在一定的相依的关系，加密和解密往往使用相联系的密钥，或由加密密钥很容易推同的密钥，或由加密密钥很容易推出解密密钥；在非对称密钥体制中，有两个密钥，其中一个公开密钥，另一个为秘密密钥，因此加密密钥和解密密钥是不同的，而且很难从加密密钥导了解密密钥。DES（Data Encyrption Standard）算法采用对称密钥，DES 的密钥长度为 64 位，RSA 算法采用非对称密钥，其中包含公开密钥，所以 B 的答案为④问题 C 的答案为①，问题 D 的答案为③。
破密者，也即解密者，指的是截取到密文，而且也知道相应的明文，由此推算出用来加密的密钥或加密算法，从而解密密文。这个难度比仅知密文要小一些。
选择明文是指破密者不仅可得到密文和相应的明文，而且也可以选择被加密的明文，这比已知明文容易，因为破密者能选择特定的明文去加密，从而得到更多关于密钥的信息，继而可

以更容易地推出用来加密的密钥或算法，因此问题 E 的选择答案应为①。
[答案]A：② B：④ C：① D：③ E：①

试题 4 （1992 年试题 8）

从供选择的答案中选出应填入下面叙述中的{ }内的正确答案,把编号写在答卷的对应栏内.
一个复杂的系统可由若干个简单的系统串联或并联构成。
已知两个简单系统 I 和 J 的失效率分别为 $\lambda i=25\times10^{-5}/h$(小时)和 $\lambda j=5\times10^{-4}/h$（小时），则由 I 和 J 经如图 3-5 所示的串联和并联构成的复合系统 P 和 Q 的失效率分别为 λp= A /h 和 λq= B /h.平均无故障时间分别为 MTBFp= C h 和 MTBFq= D h.
系统 P 开始运行后 2 万小时内能正常运行的概率 Rp= E .

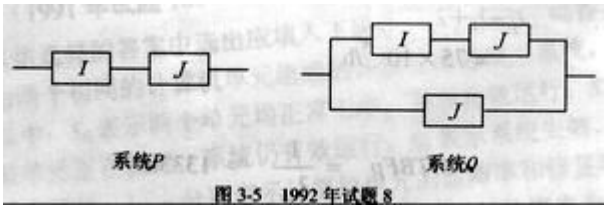


图 3-5 1992 年试题 8

供选择的答案
A、B：①$25\times10^{-5}$ ②$33\times10^{-5}$ ③$66\times10^{-5}$ ④$75\times10^{-5}$
C、D：①1333 ②1500 ③3000 ④4000
E：①e^{-5} ②e^{-10} ③$e10^{-5}$ ④e^{-20}
[解析]
系统的失效率 λ 指的是单位时间内的系统数与系统总数的比例。在稳定使用的阶段苘以认为 λ 是常数。系统从开始运行到某一时刻 t 这段时间内能正常运行的概率又称为系统的可靠性，是 t 的函数，记为 R（t），可以证明：$R(t)=e^{-\lambda t}$.
如果把系统故障发生的时刻看成是随机变量，则该随机变量的概率分布函数为：
F（t）=P{ε≦t}=
1- P{ε>t}=
1- R(t)=
$1-e^{-\lambda t}$
平均无故障时间 MTBF（Mean Time Between Failures），就是从时刻 0 开始到故障发生时刻间间隔的平均数，即随机变量 ε 的平均值，可算得：

$$MTBF=\int_0^{\infty} tF'(t)dt=\int_0^{\infty}\lambda te^{-\lambda t}dt=\frac{1}{\lambda}$$

N 个可靠性分别为 Rk、失效率分别为 λk(k=1,…，n)的子系统串联构成的复合系统，只有在每个子系统都可靠时才可靠，故其可靠性 R 及失效率 λ 分别为：
R=R1R2…Rn=

$$\prod_{k=1}^{N}e^{-\lambda_k t}=e^{-\left(\sum_{k=1}^{n}\lambda_k\right)t}$$

$$\lambda=\sum_{k=1}^{n}\lambda_k$$

由此即可求复合系统 P 的失效率为：

$\lambda p=\lambda i+\lambda j$

$=75\times10^{-5}/h$

$MTBFp=1/\lambda p\approx1333h$

$Rp(t)=e^{-75\times10-5}t$

当 $t=2\times10^{4}$ 时有：

$Rp=e^{-15}$

对于 N 个子系统并联的情况，复合系统只有在所有子系统均失效时复合系统才失效。故有：

$$R=1-(1-R_1)(1-R_2)\mathrm{K}\ (1-R_N)$$

$$=1-\prod_{k=1}^{N}(1-R_k)$$

若 N 个子系统的失效率都是一样的，即 $\lambda k\equiv\lambda(k=1,\ldots，N)$则有

$R（t）=1-(1-e^{-\lambda t})N$

$F(t)=(1-e^{-\lambda t})N$

复合系统的平均无故障时间为：

$$MTBF=\int_0^{\infty}tF'(t)dt=\frac{1}{\lambda}\sum_{k=1}^{N}\frac{1}{k}$$

本题复合系统 Q 由系统 J 和两个 I 串联构在的复合系统（不妨记为 I'）再经并联构成。复合系统 I'的失效率为：

$\lambda'I=\lambda i+\lambda j=$

$5\times10^{-4}/h$

$\lambda'I=\lambda j$

N=2 时并联系平均无故障时间：

$$MTBF_q=\frac{1}{\lambda_j}\sum_{k=1}^{N}\frac{1}{k}=3000h$$

且

$\lambda q=1/MTBFq\approx33\times10^{-5}$

[答案]A：④ B：② C：① D：③ E：③

试题 5（1991 年试题 10）

从供选择的答案中选出应填入下面叙述中的{ }内的正确答案,把编号写在答卷的对应栏内.

由两个相同的计算机单元组成的计算机维修双工系统，其状态转移图如图 3-6 所示。

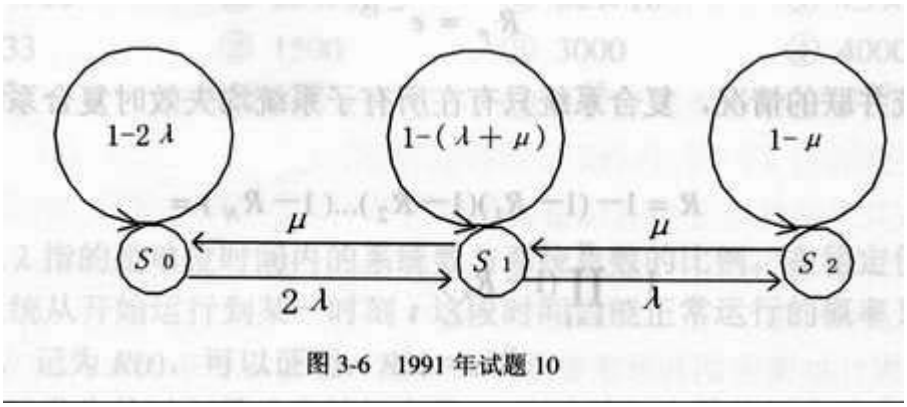


图 3-6　1991 年试题 10

其中：S0 表示两个单元均正常工作，系统有效运行；S1 表示其中一个单元正常工作，一个故障单元正在修理，系统仍有效运行；S2 表示系统失效，一个故障单元正在修理，另一个故障单元待修。λ、μ 分别表示计算机单元的故障率和修复率。该系统的状态概率转移矩阵为 A 。系统处于稳定状态时，系统无故障运行的概率为 B ；系统有效运行的概率为 C 系统失效的概率和故障的关系是 D ，和修复率关系是 E 。

供选择的答案

A：

①
$$\begin{bmatrix} 1-2\lambda & 1-(\lambda+\mu) & 1-\mu \\ \mu & \mu & 0 \\ 2\lambda & \lambda & 0 \end{bmatrix}$$

②
$$\begin{bmatrix} 1-\mu & 1-2\lambda & 1-(\lambda+\mu) \\ \mu & \mu & 0 \\ 2\lambda & \lambda & 0 \end{bmatrix}$$

③
$$\begin{bmatrix} 1-2\lambda & \mu & \mu \\ 1-(\lambda+\mu) & 2\lambda & \lambda \\ 1-\mu & \lambda & 0 \end{bmatrix}$$

④
$$\begin{bmatrix} 1-2\lambda & 2\lambda & 0 \\ \mu & 1-(\lambda+\mu) & \lambda \\ 0 & \mu & 1-\mu \end{bmatrix}$$

B,C：

① $\frac{\mu^2}{2\mu^2+2\lambda\mu+\lambda}$　② $\frac{\mu^2}{\mu^2+2\lambda\mu+2\lambda^2}$

③ $\frac{\lambda^2}{2\mu^2+2\lambda\mu+\lambda^2}$　④ $\frac{\lambda^2}{\mu^2+2\lambda\mu+2\lambda^2}$

⑤ $\frac{\mu^2+\lambda\mu}{2\mu^2+2\lambda\mu+\lambda^2}$　⑥ $\frac{\mu^2+2\lambda\mu}{\mu^2+2\lambda\mu+2\lambda^2}$

⑦ $\frac{\lambda^2+2\lambda\mu}{2\mu^2+2\lambda\mu+\lambda^2}$　⑧ $\frac{\lambda^2+2\lambda\mu}{\mu^2+2\lambda\mu+2\lambda^2}$

D：①随 λ 增大而减小，并生成线性关系 ②随 λ 增大而减小，但不成线性关系
③随 λ 增大而增大，并生成线性关系 ④随 λ 增大而增大，但不成线性关系
E：①随 μ 增大而减小，并生成线性关系 ②随 μ 增大而减小，但不成线性关系
③随 μ 增大而增大，并生成线性关系 ④随 μ 增大而增大，但不成线性关系
[解析]
设 P0、P1 和 P2 分别表示状态转移前系统处于 S0、S1 和 S2 状态的概率；P'0、P'1 和 P'2 则分别表示状态转移后系统处于 S0、S1 和 S2 状态的概率。
根据题中给出的状态转移图，写出该系统的状态转移方程如下：
P'0=（1-2λ）P0+μP1
P'1=2λP0+[1-（λ+μ）]P1+μP2
P'2=λP3（1-μ）P2
将状态转移方程写成矩阵形式为：

$$\left(P_0' P_1' P_2\right)=\left(P_0 P_1 P_2\right)\begin{bmatrix}1-2\lambda & 2\lambda & 0\\ \mu & 1-(\lambda+\mu) & \lambda\\ 0 & \lambda & 1-\mu\end{bmatrix}$$

后一个 3×3 矩阵，即为状态率转移矩阵。在系统处于稳定状态时，P'0=P0，P'1=P1，P'2=P2，P0+P1+P2=1，可得出的系统无故障运行的概率：

$$P_1=\frac{2\lambda}{\mu}P_0$$

$$P_2=\frac{\lambda}{\mu}P_1=\frac{2\lambda^2}{\mu^2}$$

$$P_0+\frac{2\lambda}{\mu}P_0+\frac{2\lambda^2}{\mu^2}P_0=1$$

$$P_0=\frac{\mu^2}{\mu^2+2\lambda\mu+2\lambda^2}$$

系统有一个单元正常工作，另一个单元有故障正在修理，系统仍能有效动行的概率：

$$P_1 = \frac{2\lambda}{\mu} P_0 = \frac{2\lambda\mu}{\mu^2 + 2\lambda\mu + 2\lambda^2}$$

总的系统可有效运行的概率为：

$$P_0 + P_1 = \frac{\mu + 2\lambda\mu}{\mu^2 + 2\lambda\mu + 2\lambda^2}$$

系统失效的概率为：

$$P_2 = \frac{2\lambda^2}{\mu^2 + 2\lambda\mu + 2\lambda^2}$$

由此可知，系统失效的概率和故障率的关系是随 λ 增大而增长率大,但不成线性关系;它和修复率的关系随 μ 增大而减小,但不成线性关系.

[答案]A:④ B:② C:⑥ D;④ E:②

试题 6（1990 年试题 9)

从供选择的答案中先出填入下面关于 N 模冗余系统的叙述中的{ }内的正确答案的对应栏内.

N 模冗余系统如图 3-7 所示,由 N(N=2n+1)个相同部件的副本和一个(n+1)/N 表决器把 N 个副本中占多数的输出作为系统的输出.

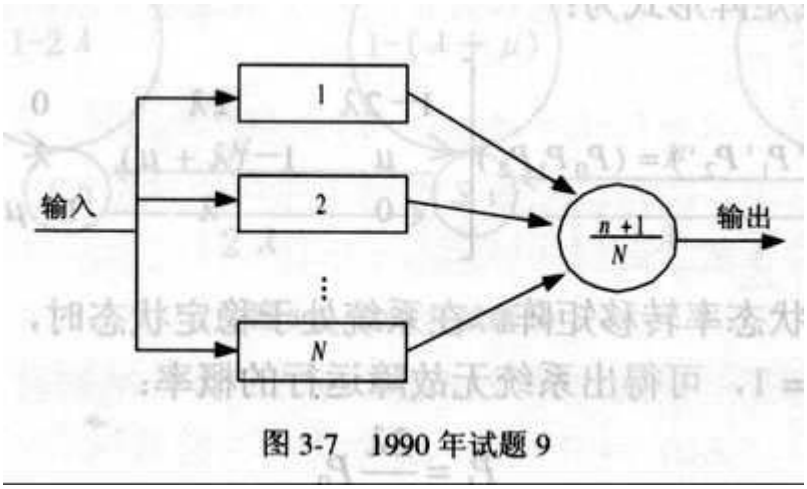


图 3-7　1990 年试题 9

设表决器完全可靠,且每个副本的可靠性为 R0,则该 N 模冗余系统的可靠性 R= A .当 $R0=e^{-\lambda t}$, 当 λt= B ,R0= C ,R 为不依赖于 N 的恒定值 D ;当 R0 小于 C ,R 是 N 的 E .

供选择的答案

A:

B、C、D：①0.1 ②0.347 ③0.5 ④0.693 ⑤0.869 ⑥0.9

E:①单调递增函数 ②单调递减函数 ③对数函数 ④指数函数

[解析]

试题中图 3-7 给出的由 N（N=2n+1）个相同的副本和一个（n+1）/N 表决器组成的 N 模冗余系统中，只要有 n+1 个以上的副本正常工作，就认为系统工作正常，输出正确。因此，

若假定表决器完全可靠，且每个副本的可靠性 R0，由概率论的有关理论可推出该 N 模冗余系统的可靠性为：

$$R=\sum_{i=n+1}^{n} C_N^i R_0^i (1-R_0)^{N-i}$$

若 $R0=e^{-\lambda t}$，则 R 是 λt 和 N 的函数，对于不同的 N 可画出 R 随 λt 变化的曲线簇如图 3-8 所示。

只有当 $\lambda t=Ln2=0.693$ 时，$R0=e^{-\lambda t}=e^{-Ln2}=0.5$

$$R=\sum_{i=n+1}^{N} C_N^i (0.5)^i (1-0.5)^{N-i} = (0.5)^N \sum_{i=n+1}^{2n+1} C_N^i = (0.5)^N \cdot \frac{1}{2}\sum_{i=0}^{N} C_N^i = (0.5)^N \cdot \frac{1}{2}(1+1)^N = 0.5$$

R 为一不依赖于 N 的恒定值 0.5。

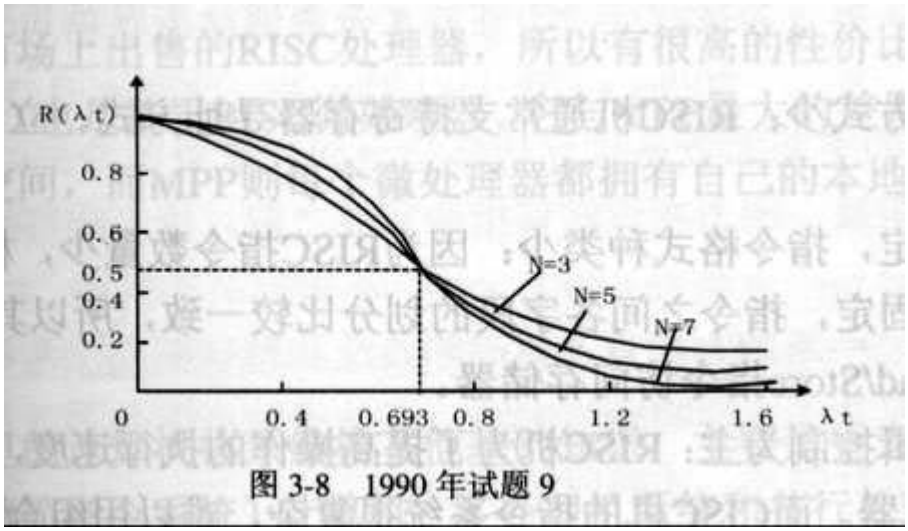


图 3-8　1990 年试题 9

当 R0 小于 0.5 时，意味着单个副本产生正确结果的可能性小于产生错误结果的可能性（I-R0），而 N 模冗余系统又是以多数副本的表决结果作为系统的输出。显然，此时副本的个数（N）越多，就越容易引起错误，即整个系统的可靠性（R）就越小。换言之，R 是 N 单调递减函数。从上面给出的 R 依赖于 λt 变化的曲线族图中也可直观地看了，在 $\lambda t>0.693$（即 $R0<0.5$）处，N 越大的曲线越靠下，即 R 越小。

[答案]A：④ B：④ C：③ D：③ E：②

2.4 体系结构其他基础知识

2.4.1 主要知识点

了解掌握流水线技术、RISC 计算机、并行处理技术、多处理机系统方面的基本知识。

2.4.1.1 流水线技术

流水线技术是通过并行硬件来提高系统性能的常用方法。计算机流水线技术包括指令流水线和运算操作流水线。

计算机中一条指令的执行需要若干步，通常采用流水线技术来实现指令的执行，以提高

CPU 性能。流水线设计的基本思想其实很简单，任何一个可以分解的任务都可以用流水线来做，可以设置多个处理机构，分别执行相应的子任务。为了提高流水线性能，有些处理时间长的步骤还需分解成更小的步骤，使流水线上所有步骤的处理时间相同。实际上，流水线技术对性能的提高程度取决于其执行顺序中最慢的一步。

在实际情况中，流水线各个阶段可能会相互影响，阻塞流水线，使其性能下降。阻塞主要由两种情形引起：执行转移指令和共享资源冲突。

指令流水线技术其实是把处理瓶颈从 CPU 子系统转移给了存储子系统。在存储系统中也需要使用流水线技术的 CPU 系统的处理能力。流水线计算机的存储器分成若干（4 个）独立存储体，以支持流水线方式并发访问。流水线计算机也使用了 Cache，通常分为指令 Cache 和数据 Cache，各自用于存放指令和操作数。

2．4．1．2RISC 计算机的主要特点

①指令数量少：RISC 机优先选取使用频率最高的一些简单指令以及一些常用的指令，避免使用复杂指令。

②复杂的寻址方式少：RISC 机通常支持寄存器寻址方式、立即数寻址方式以及相对寻址方式。

③指令长度固定，指令格式种类少：因为 RISC 指令数量少，格式也相对简单（与 CISC 比较）其指令长度固定，指令之间各字段的划分比较一致，所以其译码也相对容易。

④只提供了 Load/Store 指令访问存储器。

⑤以硬布线逻辑控制为主：RISC 机为了提高操作的执行速度，通常采用硬布线逻辑（组合逻辑）来构建控制器。而 CISC 机的指令系统很复杂，难以用组合逻辑电路来实现控制器，通常采用微程序控制。

⑥单周期指令执行：RISC 机因为简化了指令系统，所以很容易利用流水线技术使得大部分指令都能在一个机器周期内完成。

⑦拥有相当多的寄存器：RISC 机拥有相当数量的寄存器，可以用于存放中间数据和优化对操作数的访问，从而提高了处理器的性能。

⑧优化的编译器：RISC 的精简指令集使编译工作简单化。

采用 RISC 技术和 CPU 硬件一般具有寄存器数量多、采用流水线组织、控制器的实现采用硬布线控制逻辑电路等特点。

2．4．1．3 并行处理机

并行处理机有时也称为阵列处理机，其得名于在单一控制部件控制下的由多个处理单元构成阵列。主要用于要求进行大量高速向量或矩阵运算的应用领域。

并行处理机的并行性来源于资源重复，把大量相同的处理单元 PE 通过互联网络连接起来，在统一的控制器（CU）控制下，对各自分配来的数据并行地完成同一条指令所规定的操作。PE 是不带指令控制部件的算术逻辑运算单元。

并行处理机的特点：

① 强大的向量运算能力

② 并行方式。并行处理机的并行性来源于资源重复而不是时间重叠，利用的是并行性中的同时性而不是并发性。并行机提高运算速度的主要方法是增加处理单元的个数。

③ 适用于专门领域，如矩阵运算、向量运算等。

④ 标量运算速度对系统性能的影响也很明显。

⑤ 具有向量化功能的高级语言编译程序有助于提高并行处理机的通用性，以及减少编

译时间。

2.4.1.4 多处理机

多处理机具有两个或两个以上的处理机，共享输入/输出子系统，在统一的操作系统控制下，通过共享主存或高速通信网络进行通信，协同求解一个在而复杂的问题。多处理机通过利用多台处理机进行多任务处理来提高速度，利用系统的重组能力来提高可靠性、适应性和可用性。多处理机是新一代计算机结构的基本特征，多处理机具有共享存储器和分布存储器两种不同结构。

多处理机属于 MIMD 系统，与 SIMD 的并行处理机相比，有很大的差别。其根源就在于两者的并行性等层次不同，多处理机要实现的是更高一层的作业任务间的并行。

大规模并行处理机（MPP）是由众多的微处理器（从几百到上上万）组成的大规模并行处理系统。MPP 可以采用市场上出售的 RISC 处理器，所以有很高的性价比。

2.4.2 试题分析

根据考试大纲，这里的计算机体系结构其他基础知识，主要是更为高级的硬件技术，包括流水线操作、RISC（精简指令系统）计算机、多处理机系统和并行处理。从历年的试题来看，主要考查 RISC（精简指令系统）计算机的相关知识和流水线技术。

试题 1 （2000 年试题 11）

从供选择的答案中选出应填入下面叙述中的{ }内的正确答案,把编号写在答卷的对应栏内.
现采用 4 级流水线结构分别完成一条指令的取指、指令译码和取数、运算以及送回运算结果 4 个基本操作，每步操作时间依次为 60ns、100ns、50ns 和 70ns。该流水线的操作周期应为 A ns。若有一小段程序需要用 20 条基本指令完成（这些指令完全适合于流水线上执行）则得到第一条指令结果需 B ns ,完成该段程序需 C ns。
在流水线结构的计算机中，频繁执行 D 指令时会严重影响机器的效率。当有中断请求发生时，若采用不精确断点法，则将 E 。
供选择的答案
A：①50 ②70 ③100 ④280
B：①100 ②200 ③280 ④400
C：①1400 ②2000 ③2300 ④2600
D：①条件转移 ②无条件转移 ③算术运算 ④访问存储器
E：①公影响中断响应时间，不影响程序的正确执行。
④ 不仅影响中断响应时间，还影响程序的正确执行。
⑤ 不影响中断响应时间，但影响程序的正确执行。
⑥ 不影响中断响应时间，也不影响程序的正确执行
[解析]
本题主要考查流水线技术。
由流水线技术的基本特征可知，其平均时间取决于流水线最慢慢的操作，所以该流水线的操作周期为 100ns。
由题中条件可知，完成 1 条指令需要 4 个基本操作，每个操作需要 1 个周期，执行第 1 条指令时，还不能充分发挥流水线的技术优势，需要执行 4 个周期，才能得到第 1 条指令的运行结果，共需要 400ns。由于采用流水线技术，从第 1 条拽令的第 2 步基本操作开始，后续指令开始并行执行，可将 20 条指令的执行过程看作 20 段流水线，由于基本操作重叠执行，除第 1 条指令外，每条指令的执行可视为需要 1 个周期，所以总共需时间为：

400ns+(20-1)×100ns=2300ns
影响流水线性能的主要因素是执行转移指令和共享资源冲突。当流水线性执行转移指令时，会引起流水线的阻塞，因为在该转移指令完成之前，流水线都不能确定下一条提令的地址。流水线计算机处理中断的方法有不精确断点法和精确断点法两种，采用不精确断点法，当发生中断后，计算机并不立即响应中断，而是先禁止指令再时入流水线，然后等待已在流水线中的所有指令执行完毕，才响应该中断。
[答案]A：③ B：④ C：① D：②

试题 2 （1999 年试题 12）

从供选择的答案中选出应填入下面叙述中的{ }内的正确答案,把编号写在答卷的对应栏内.
计算机执行程序所需时间 P，可用 P=I·CPI·T 来估计，其中 I 是程序经编译后的机器指令数，CPI 是执行每条指令所需的平均机器周期数，T 为每个机器周期的时间。RISC 计算机是采用 A 来提高机器的速度。它的指令系统具有 B 的特点。指令控制部件的构建，C 。RISC 机器又通过采用 D 来加快处理器的数据处理速度。RISC 的指令集使编译优化工作 E 。
供选择答案
A：①虽增加 CPI，但更减少 I。 ②虽增加 CPI，但更减少 T。
③虽增加 T，但更减少 CPI。 ④虽增加 I，但更减少 CPI。
B：①指令种类少。 ②指令种类多。 ③指令寻址方式多。 ④指令功能复杂
C：①CISC 更适于采用硬布线控制逻辑，而 RISC 更适于采用微程序控制。
②CISC 更适于采用微程序控制，而 RISC 更知于采用硬布线控制逻辑。
③ CISC 和 RISC 都只采用微程序控制。
④ CISC 和 RISC 都只采用硬布线控制逻辑。
D：①多寻址方式 ②大容量内存 ③大量的寄存器 ④更宽的数据总线
E：①更简单 ②更复杂 ③不需要 ④不可能
[解析]
本题考查 RISC（精简指令集）计算机有关知识。
在指令系统已确定的情况下，在硬件设计时要尽量使大多数指令能实现"单周期执行"，即在采用流水线组织的计算机中，每个机器周期能完成一条指令功能，而不是一条指令从取指令到完成指定功能只要一个机器周期。另外尽量缩短机器周期，提高主频，从而提高机器运算速度也是硬件设计的主要目标。
采用 RISC 技术的 CPU 硬件一般具有以下特点：
（1） CPU 中寄存器数量相当多；
（2） 采用流水线组织；
（3） 采用硬布线控制逻辑。
RISC 精简了指令系统，使指令种类更少，但各指令所需要的机器周期数相差不大，而且每个机器周期所完成的操作也比较简单，因此没有必要再采用微程序设计。这样，程序经过编译后指令数是增加了，但执行指令的平均机器周期减小了，使得程序的执行时间大大减小。RISC 指令的简单性编译工作简单化。因为在编译时不必在具有类似效果在许多指令中去进行复杂选择，同时寻址方式少，也不必为选择理想的寻址方式而浪费时间，指令长度固定，指令格式少，这使得优化时更换指令或取消指令变得容易。同时，因为大部分指令能够实现单周期执行，使编译程序比较容易调整指令流。
[答案]A：④ B：① C：② D：③ E：①

试题 3 （1995 年试题 7）

从下列关于 RISC 的叙述中，选出 5 条正确的叙述，并按编码从小到大的次序写在答卷的 A～E 栏内。

①RISC 机器对编译程序的要求比传统的 CISC 低。

② RISC 的 CPU 工艺水平已达到 0.1μm 线宽。

③ RISC 指令招待采用微程序控制方式

④ RISC 的 CPU 采用流水线技术。

⑤ RISC 比传统的 CISC 的 CPU 通用寄存器多。

⑥ RISC 指令格式和寻址方式的种类少。

⑦ RISC 机器用作服务器时性能比传统的 CISC 低。

⑧ RISC 较传统的 CISC 的 CPU 存储器之间的操作功能更强。

⑨ RISC 较传统的 CISC 的 CPU 存储器操作指令内容更丰富，功能更强。

⑩ RISC 只选取一些使用频率高但并不复杂的指令。

[解析]

RISC 即精简指令系统计算机，已在工作站和小型机中广泛采用。

①错误。RISC 机器选取的是使用频率最高的简单指令和很有用但不复杂的指令，从而达到精简指令系统的目的。这种指令系统不及传统 CISC 指令系统丰富，使用起来要求有较高的技巧才能充分发挥 RISC 机器的优点。用户通常用高级语言编程，因而对将高级语言自动翻译成机器指令的编译程序提出了更高的要求，要特别重视编译的优化工作。

②错误。目前超大规模集成电路（VLSI）的工艺水平还达不到 0.1μm 线宽的程度。RISC 的 CPU 作为 VLSI 的一种也不可能要求其工艺水平达到如此细的线宽。它是靠改进其 CPU 内部的系统结构来提高性能的。

③错误。RISC 机器精简了指令系统，每个机器周期所完成的操作也较为简单，因此没有秘要再采用微程序方式来控制指令的执行，且硬布线控制逻辑可提高执行的速度。

④正确。RISC 机器采用流水线技术，使得多条指令的操作可重叠执行，从而使得绝大多数拽令执行的机器周期数仅为 1。

⑤正确。访问存储器要花费较多的时间，RISC 机器为了减少访问次数，通常设有较大的通用寄存器堆，内有许多通用寄存器，并且寄存器间的操作功能也列强，这样许多操作都可在寄存器之间进行。

⑥正确。减少指令格式和寻址方式的种类是指令系统精简的一个重要方面，这样便于发挥流水线的效能，使得许多操作可重叠执行。

⑦错误。实际上，目前用作服务器的大多数工作站和小型机都主要采用了 RISC 的芯片和结构。

⑧正确。见⑤的分析说明。

⑨错误。为了尽可能减少访问内存的次数，大多数 RISC 机器只有很少且非常简单的存储器操作指令。以采用 RISC 技术的 SPARC 处理器芯片为例，它只有两条存储器操作指令，即从内存取数 LOAD 和向存中存数 STORE。

⑩正确。参见对①的分析说明。

[答案]A：④ B：⑤ C：⑥ D：⑧ E：⑩

试题 4 （1993 年试题 11）

从供选择的答案中选出应填入下面叙述中的{ }内的正确答案,把编号写在答卷的对应栏内.

为了大幅度提高处理器的速度，当前处理器中采用了指令及并行处理技术，如超级标量（Superscalar,）它是指 A 。流水线组织是实现指令并行的基本技术，影响流水线连续流动的因素除数数据相关性、转移相关性外，还有 B 和 C ；另外，要发挥流水线的效率，还

必须重点改进 D 。在 RISC 设计中，对转移相关性一般采用 E 方法解决。
供选择答案
A：①并行执行的多种处理安排在一条指令内
⑤ 一个任务分配给多个处理机并行执行
⑥ 采用多个处理部件多条流水线并行执行。
⑦ 增加流水线级数提高并行度
B、 C：①功能部件冲突 ②内存与 CPU 速度不匹配 ③中断系统 ④访内指令
D：①操作系统 ②指令系统 ③编译系统 ④高级语言
E：①猜测法 ②延迟转移 ③指令预取 ④刷新流水线重填
[解析]
为了大幅度提高处理器的速度，当前处理器中都采用了指令级并行处理技术，如超级标题就采用了用多个处理部件多条流水线来并行执行指令，在超级标量处理机中配置了多个处理部件和指令译码，采取了多条流水线，还有多个寄存器端口和总线，可以同时执行多悠个操作，以并行处理来提高计算机的速度。
流水线是实现指令并行的基本技术，其基本思想是将一条指令的全过程分为若干段，如取指、译码、取操作数、运算存结果等，每段由不同的功能部件来执行。当流水线各段工作都饱满时，才能达到最高效率。
计算机中不同指令的执行过程并不是规整统一的，有可能不同的指令在不同的操作中用到同一功能部件，使得流水线吵同指令在同一时间访问同一功能部件，这就是功能部个冲突。此时，必然有指令要停下来，从而影响了流水线的效率。
数据有关，即不同甘共苦的指令要访问同一存储单元的数据，如果下一条指令的操作数地址是上一条指令保存结果的地址，那么取操作数的操作就要在保存结果的操作执行完之后才能进行。在遇到条件转移指令时，当条件转移指令进入流水线后，直到下一地址确定之前，流水线不能继续工作而处于等待状态，这也要影响流水线效率。
中断系统工作也和转移情况类似，当中断发生时，要中止当前程序而转入中断程序，此时流水线也会中断。因此数据相关性、转移相关性、功能部件冲突和中断系统都是影响流水线连续流动的因素。好的编译系统产生的目标指令流可使得流水线尽可能满负荷工作。所以，要充分发挥流水线效率，重点是改进编译系统。
在精简指令系统计算机（RISC）中，若遇到成功的条件转移指令或无条件转移指令，流水线使预取的反映指令的失效，从而浪费了机器周期，影响了流水线的效率。为了提高流水线工作效率，RISC 一般将转移指令与其前面的一条指令对换位置，使成功的转移总是在紧跟的指令被执行之后发生，从而使预取的指令不作废，这就是延迟转移法。
[答案]A：③ B：① C：③ D：③ E：②

试题 5 （1991 年试题 11）

从供选择的答案中选出应填入下面叙述中的{ }内的正确答案,把编号写在答卷的对应栏内.
减少指令执行周期数是 RISC 计算机性能提高的基础,它是通过 A 、指令控制部件 B 佩代码、C 寄存器、 D 寻址方式和限制访问内存来实现的。 E 都是与 RISC 芯片有关的名字。
供选择的答案
A：①超长指令字（VLIW） ②单指令多功能 ③精简指令系统 ④适当扩充指令系统
B：①尽可能多用原有 ②增加 ③软件固化 ④不用
C：①增加 ②减少 ③精选 ④不用
D：①增加 ②减少 ③精选 ④不用
E：①SPARC 和 MIPS ②SPARC 和 SCSI

③MIPS 和 FLOPS ④MIMD 和 EISA
[解析]
精简指令系统计算机 RISC（Reducde Instruction Set Computer）通过精简指令系统，拽令控制部件不用微代码，增加寄存器，减少寻址方式和限制访问内存等方法来减少指令执行周期数，从而提高计算机的性能。因此，减少指令执行周期数是 RISC 计算机性能提高的基础。SPARC 是 Sun 公司生产的 RISC 芯片的名称，MIPS 是生产 RISC 芯片 R3000 的公司的名称，这两种芯片都是著名的 RISC 芯片。因此，SPARC 和 MIPS 都是与 RISC 芯片有关的名字。SCSI 是一种小型计算机标准接口，通常用来连接外存储器。FLOPS 表示每秒钟浮点运算的次数。MIMD 为多指令流多数据流，是一种按弗林分类法分类的计算机系统类型。EISA 是一种扩充的工业标准总线系统结构，它们都和 RISC 芯片无关。
[答案]A：③ B：④ C：① D：③ E：①

2.5 综合性试题

最后解析几道综合性试题，涉及到操作系统、网络、并行处理、性能评价等知识点。

试题 1 （1998 年试题 11）

从供选择的答案中选出分别与下述概念最密切相关的术语组，把相应的编号写在答卷的对应栏内。
1．电子商务 A
2.人机界面 B
3．计算机性能评价 C
4.并行处理 D
5．网络体系结构 E
供选择的答案
A～E：
①OSI、对等层协议、无连接服务
②OSI、基准测试程序、TPC
③OCR、基准测试程序、TPC
④吉普森方法、基准测试程序、TPC
⑤EDI、网上商店、身份验证
⑥向量计算机、MPP、多指令流
⑦EDI、网上商店、OCR
⑧所见即所得、语言识别、OCR
⑨EDI、MPP、多指令流
⑩语音识别、OCR、OSI
[解析]
可依次判断供选择的术语组，看哪些与给定的概念相关。
·OSI：Open System Interconnection,开放系统互连，与网络体系结构相关。
·对等层协议：与网络体系结构相关。
·无连接服务：与网络体系结构相关。
·TPC：衡量流水线处理机的性能，与计算机性能评价有关。
·OCR：光学字符识别技术，与人机界面有关。
·基准测试程序：目前比较好的测试计算机性能方法，与计算机性有评价有关。

·吉普森方法：与计算机性能评价有关。
·EDI：Electronic Data Interchange,电子数交换，无纸贸易，与电子商务有关。
·网上商店：以 Internet 为基础，所有卖买包括洽淡、订货、在线付（收款）、开据电子发票都能过 Internet 完成，与电子商务有关。
·身份验证：电子商务中用于验证网络客户的真实身份。
·向量计算机:与并行处理有关.
·MPP：位平面阵列处理机，与并行处理有关。
·多指令流：同一时刻可执行多条指令，与并行处理有关。
·语音识别：用软件把人的声音识别为正确的命令或文字，与人机界有关。
[答案]A：⑤ B：⑧ C：④ D：⑥ E：①

试题 2 （1994 年试题 9）

在下列 12 组中，每组中有 3 个词。从该 12 组中，选出组内 3 个词都是与提高计算机硬件处理有关的 5 组,将编号依次写在答案的 A～E 栏内。
①数据库 双机系统 存储器交叉存取
②高速缓冲存储器 虚拟计算机 并行处理
③存储器交叉存取 并行处理 先行控制
④流水线 高速缓冲存储器 存储器交叉存取
⑤双机系统 流水线 结构化程序设计
⑥高速缓冲存储器 存储器交叉存取 先行控制
⑦数据库 批处理 流水线
⑧双机系统 并行处理 虚拟计算机
⑨先行控制 并行处理 流水线
⑩流水线 并行处理 高速缓冲存储器
（11）双机系统 流水线 结构化程序设计
（12）虚拟计算机 批处理 存储器交叉存取
【解析】
根据题意，重点分析与提高计算机硬件处理速度有关的术语和词句。
·数据库便于大量数据的存取操作和检索，与提高计算机硬件处理速度无关。
·双机系统能提高可靠性，与提高计算机硬件处理速度无关。
·存储器交叉存取可提高存储器的存取速度。
·高速缓冲存储器减少了对下一级速度较慢存储器的访问，可提高访问下一级存储器的速度。
·虚拟计算机与提高计算机硬件处理速度无关，虚拟存储器只能扩大程序使用空间，也与提高计算机硬件处理速度无关。
·并行处理可提高硬件处理速度。
·先行控制可预取指令，提高了中央处理器的处理速度。
·流水线使得不同指令的不同操作步骤可在宏观上并行工作，提高了处理速度。
·结构化程序设计是一种较好的模块化程序设计方法，可使程序清晰、易于理解和维护，但与提高计算机硬件处理速度无关。
·批处理是操作系统中一种作业调度的方法，可提高计算机硬件的工作利用率，与提高计算机硬件本身的处理速度无关。
只有③、④、③、③和@组内的 3 个词都是与提高计算机硬件处理速度有关的。
【答案】A：③B：④C：③D：③E：③

试题 3 （1992 年试题 9）

从下列叙述中选出 5 条正确的叙述，把编号依次写在答卷的 A～E 栏内。

①存储器的多体交叉是指将连续地址的存储单元交叉分配到多个模块中，使连续地址的读出高速化。

②操作系统的固化是指用微程序实现指令系统中的复杂指令。

③容错计算机主要用于批处理。

④向量处理机是采用流水线技术通过时间重叠来提高向量运算速度的处理机。

⑤VLIW 计算机是一种新型的体系结构，能充分开发细粒度的并行性，指令长度特别长。

⑥激光打印机是一种印字质量高的高速击打式打印机。

⑦中断屏蔽位为程序员提供了控制中断处理顺序的手段。

⑧在浮点数的加减运算中，首先要进行对阶，即将大阶数的阶码化为与小阶数的阶码相等。

⑨故障诊断就是对机器单件进行测试并确定是否存在故障的过程。

⑩并行性包括同时性（Simultaneity）和并发性（Concurrency）双重含义。

【解析】

①正确。多体交叉存储器将连续地址的存储单元依次分配到多个存储体模块中，每次读写时，多个存储体可同时动作，将多个连续地址的存储单元同时读出。

②错误。操作系统固化是指将操作系统，特别是常用的核心部分用固件来实现，用微程序实现指令系统中的复杂指令并不等于操作系统的固化。

③错误。容错是采用冗余的方法来消除故障的影响，使计算机在出现某些故障的情况下仍可继续运行或降级运行。批处理是操作系统调度作业的一种方法，在批处理系统中，各项工作有序地排在一起形成一个作业流，用户和计算机间不再直接交互信息，由计算机系统自动地、顺序地执行作业流。

容错计算机和批处理之间不存在必然的联系，容错计算机可用于批处理系统，也可用于分时系统或实时系统，特别是实时系统由于要求有更高的可靠性，对容错也有更高的要求。

④正确。向量处理机，有时亦称为向量计算机，例如 Cray－1 等，都是采用流水线技术通过时间重叠的方法来提高向量运算速度的。

⑤正确。超长指令字（VLIW，Very Long InstrUction Word）计算机因其指令宇特别长，有的可达数百位而得名。VLIW 计算机通常具有多个异构的处理单元。一条超长指令可分为若干字段，分别控制各个处理单元和其他功能部件，执行不同的操作，能充分开发细粒度的并行性特性。

⑥错误。击打式打印机指的是靠机械冲击力通过色带等打印出字符或图形的打印机；激光打印机则是利用电子照相原理，即激光放电形成静电影像，吸附墨粉，再转印到普通纸上，最后加热定影印出字符或图形，虽然其印字质量高，但不是击打式打印机。

⑦正确。中断屏蔽位通常在程序状态字中，由程序员编程来置位。当中断屏蔽位被置成“l”（或者等价地，在某些计算机中，中断允许位被置成“0”）时，即使相应的中断源存在，也不允许其提出中断申请或不响应其中断申请。通过这种手段，程序员可以控制中断处理的顺序。

⑧错误。在浮点加减法运算中首先要进行对阶，使参加运算的两个操作数具有相同的阶码，然后只需对其尾数进行相应的运算，对阶的过程总是通过尾数算术右移，将小阶数的阶码增加，以向大阶数的阶码靠拢，而不是将大阶数的阶码向小阶数的阶码靠拢。

⑨错误。对机器硬件测试并确定其是否存在故障的过程称为故障检测。判定故障发生在某个子系统、功能块直至器件的过程称为故障定位。故障诊断应包括故障检测和故障定位两个方面。

⑩正确。并行性包括同时性和并发性双重含义，前者指两个或多个事件在同一时刻发生，后

者指两个或多个事件在同一时间间隔内发生，前者比后者要求更为严格。

【答案】A：① B：④ C：⑤ D：① E：(10)

三 网络基础知识试题精解

对高级程序员这个层次的人来说，网络知识显得重要多了，最近几年每年至少有一道专门考查网络知识的试题。网络基础知识一个突出的特点是涉及面广、基本概念和术语多，复习时应注意从面上把握一些重要的概念，同时了解一些最新的网络技术知识。

3.1 主要知识点

掌握计算机网络的功能、分类与组成，网络结构与通信，网络协议模型 OSWM 和 TCP／IP 协议以及 Client／Server 结构，了解 Internet／Intranet、网络安全和网络管理方面的知识。

3.1.1 网络的功能、分类和组成

3.1.1.1 网络的功能

计算机网络具备 3 大基本功能：①通信交往，即计算机之间和计算机用户之间的相互通信与交往；②共享资源，包括硬件资源、软件资源、数据与信息资源；③计算机之间或计算机用户之间的协同工作。

计算机网络的主要用途：①可产生一个性能／价格比更好的系统；②提供具有更好可用性和可靠性的应用环境；③在计算机网络内可以通过合理调度实现计算机之间工作负荷的均衡分配；④由计算机网络所构成的系统可以更方便地进行资源扩充和升级换代；⑤可提供友好方便的用户使用界面和计算机资源的有效管理手段。

3．1．1．2 网络的分类

一般按地理区域范围可分为局域网（LAN）、城域网（MAN）、广域网（WAN）、互联网（Int。et）4 种类型。

局域网的基本特征是：

①网络的所有物理设备分布在半径不超过几公里的有限地理范围之内。

②整个网络由同一个组织或机构所拥有。

③在局域网中可实现相当高的数据传输速率，如传输速率范围达 1000MbiUS。

④网络连接相当规整，有严格的标准可遵循。

城域网指所有物理设备分布在同一城区内，覆盖范围约 10km。

广域网的基本特征是：

①网络中信息的传输距离相对很长，可达几千公里以上，涉及到对远程（非本地）计算机的存取与访问。

②通常分属于多个单位或部门所有，资源子网中的各类资源与通信子网分别由各自管辖与负责。

③长距离通信线路上传输速率相对较低，一般在几十 kbit／s 至 ZMbit／S 的数量级之间。

④网络的相互连接结构通常不规整。

互联网实质上是指两个或多个网络互相连接所形成的网络。因特网是全球最大的、开放

的、由众多网络相互连接而成的计算机网络。

3．l．l．3 网络的基本组成

计算机网络由计算机硬件、软件、通信设备和通信线路（通信介质）、数据和信息资源所组成。一般把网络分成资源子网和通信子网两大部分。局域网由网络硬件、网络软件、网络 信息资源和应用程序 3 个部分组成。局域网的资源于网由网络中的各台计算机及其外部设备组成；通信子网由传输介质、网卡和网络连接设备组成，比较简单。广域网的通信子网则很
复杂，通常要电信部门负责提供。

3.1.2 网络协议与标准

网络协议（有时也称为通信协议）是指在计算机与计算机之间进行通信必须遵循的一些事先约定好的规则。网络协议必须遵循标准化的体系结构，目前主要有 ISO 的 OSI 标准和 TCP／IP 协议组标准。

3.l.2.1 二 ISO 的 OSngM 层次模型

在 OSllRM 层次模型中，把网络协议规定成 7 层模型：

①物理层给出了在一个通信信道的物理媒体上传输原始的二进位数据流时的协议。

②数据链路层给出了把二进位数据流划分成为数据帧，并依照一定规则传送与处理的协议。

③网络层把数据帧划分成更小的“分组”，规定分组的格式，给出使分组经过通信于网正确地从源传送到目的地的协议。网络层提供数据报服务和虚电路服务两类典型的数据分组传送服务方式。

④运输层也称为传输层，根据高层用户的请求建立起有效的网络通信连接。因特网广泛采用的 TCP 协议是一个传输层协议，用来在一个不可靠的互联网上提供可靠的端到端字节流服务协议。

⑤会话层的作用是允许在不同主机上的各种进程之间进行会话。

⑥表示层为应用层提供所传输的信息在表示方面的规则与协议。

⑦应用层为各类不同的网络应用提供使用网络环境的手段，具体规定了用户级别需要的、带有通信任务的信息服务的规则和协议。

3.1.2.2 TCPffP 协议组

TCPllP 协议组，也称因特网协议组，共有 5 层结构。

物理层和数据链路层与 OSllR 的规定相同，主要实现网络数据的物理存取。

互联网层（即网络层）用来方便地连接各种通信子网，负责选择合适的通信节点，使数据能从源主机发往目的地主机。在互联网层有 IP 协议、ICMP 协议、ARP 协议与 RARP 协议。IP 协议是互联网层的核心协议，负责 IP 分组的传送，ICMP(网际控制报文协议)、ARP(地址转换协议)RARP(反向地址转换协议)等协议则提供辅助的控制功能。

传输层的协议提供主机之间进程与进程的有效数据传输。在传输层中有 TCP 协议与 UDP 协议 TCP 协议即传输控制协议，提供了可靠的面向连接的数据流传输服务规则和约定。用户数据报协 UDP 提供了无连接数据报服务的规则。UDP 协议比 TCP 协议有更好的性能和更高的率，它也使用两个端口来标识源与目的地机器内的端点。

应用层提供了各种应用程序使用的协议，因特网上应用广泛的 HTTP、FTP、Telnet 等服务都是 TCP/IP 应用层协议。应用层对应于 ISO/RM 的会话层、表示层、应用层。

3.1.3 局域网技术

3.1.3.1 主要网络专用设备

主要有网卡、集线器(Hub)、重发器、网桥和交换机等设备。网卡及其驱动程序事实上已基本实现了网络协议中低两层的功能。Hub 是物理层协议级的互联设备，它将多个站点互联起来，也允许将多个网段连到同一个 Hub。重发器是一种在物理层上互联网段的小设备，它放大、增强信号并进行转发以保证信号的可靠传输，重发器连接的两个网段，必须是同一种类型的 LAN。网桥也称为桥接器，是一种在数据链路层把网段互相连接起来的设备。在网桥中可以进行两个网段之间的数据链路层的协议转换。交换机也称为交换器，是在 LAN 中互连多个网段，并可进行数据链路层和物理层协议转换的网骆互联设备。

3.1.3.2 主要传输媒体

目前在网络中数据与信息传输的(有线信道)线路主要采用 3 大类传输媒体，即同轴电缆、双绞线与光纤。双绞线是使铜质导线相互之间按一定扭矩绞合在一起的一种传输媒体，按其是否外加屏蔽层而区分为屏蔽双绞线(STP)和非屏蔽双绞线(UTP)。双绞线也可按其电气特性加以分类，网络中最常用的是 3 类线和 5 类线，目前已有 6 类以上的线。同轴电缆在 LAN 中主要采用幼缆(适用于距离短、站点少的情况)和粗缆(适用于较长距离或较多站点的情况)。光纤由单根玻璃纤维纤心、纤心的包层和塑料保护涂层的组成。根据光在光纤中的传播方式，当前有两大类常用的光纤：单模光纤和多模光纤。

3.1.3.3 LAN 的协议标准和主要的媒体访问控制方式

在 LAN 标准中主要关心物理与数据链路层的网络标准。数据链路层又分为媒体访问控制子层 MAC(与不同的媒体有关)和逻辑链路子层 LLC(与媒体及其访问方式无关)。目前 LAN 有十几个协议标准，统称为 ISO 8802 标准(或 IEEE 802 标准)。

当前使用最广泛的 LAN 是以太网。以太网(Ethernet)也称为 CSMA/CD 总线网。其主要特征是所有 LAN 中的站点本质上都连接在一条公共的总线上，以广播方式通信，采用带有冲突检测(CD)的载波侦听多路访问(CSMA)的媒体访问控制原理进行工作。

令牌环网是由一组连接成环形的计算机站点组成的 LAN，在各个站点之间依照在环上的次序传输令牌的帧信息。常规的令牌环网最高可用 16Mbit/s 工作。与以太网相比较，令牌环网的优点是能提供良好的实时性能和定时保证，因此令牌网主要用于对实时性与可靠性要求更高的一些应用中(如工业自动化控制系统等)。

除了以太网和令牌环网外，目前常用的 LAN 构建方式还有快速以太网、FDDI 光纤网和 ATM 交换式 LAN、千兆位以太网。

3.1.4 广域网技术

3.1.4.1 数据通信

计算机数据通信指的是通过通信信道在计算机之间进行数据与信息的收集、传输、交换或重新分布的一个过程。数据通信涉及到以数字形式或者模拟形式发送数据与接收数据这两个主要环节。通常把发送出数据的来源称为信源或数据源，而把接收数据的目的称为信宿或数据宿。

在数据通信基本模型中把作为信源与信宿的计算机(或其他数字装置)称为 DTE(数据终

端设备)，把变换器(如调制器)和反变换器(如解调器)称为 DCE(数据通信设备)。在 DTE 与 DCE 之间应当有标准化的接口，比如 MODEM 与 PC 间的接口常用 COM 串行通信(EIA-RS232C)标准接口，就是物理层协议的一个例子。

数据交换的基本方式有 3 种：电路交换、报文交换和分组交换。这几种方式有时也可能组合起来使用。

3.1.4.2 网络接入技术

(1)面向家庭、小型商务或小规模应用的主要接入技术。

①普通电话公用网的接入。

②ISDN 接入，即窄带的综合业务数字网服务。

③ADSL 接入，即非同步数字用户环路接入技术。

④Cable Modem 接入，即采用电缆调制解调器在有线电视电缆上进行数据调制。

⑤低轨道卫星网接入，这是目前主要的无线接入技术之一。

(2)面向大型单位与组织机构 ISP(指通信子网)的接入技术。

①X.25 公用分组交换网接入。

②帧中继接入采用租用专线方式，是一种高速流水线方式的分组交换技术。

③光纤接入，光纤接入技术可分为光纤环路技术(FITL)和光纤同轴混合技术(HFC)。其中光纤环路技术采用全数字传输方式，可细分为光纤到路边(FTTC)、光纤到大楼(FTTB)和光纤到户(FTTH)等。光纤接入可用于各类高带宽、高质量的应用环境中，比如对于大客户，可以以 DDN 专线、将来的 B-ISDN 或 ATM 服务专线等方式接入。

3.1.4.3 路由器和网关

路由器(Router)是实现网络层互连的设备，在 OSI 标准中称为中间系统。这是一种应用相当广泛的提供网络服务的设备。通常路由器提供了各种速率的多种链路或子网的接口，是一种价格较为昂贵的主动的智能网络节点，一般能参与网络的管理，提供对资源的动态控制，支持网络工程的实现和协助网络维护活动。路由器的本质特征是：提供网络层的互连，具有路由选功能与流量控制能力。

网关(Gateway)也称为信关，实质上是通信服务器。这是一种相当复杂的在应用层进行网络互连的设备，可以用来连接异种网络(包括不同的各类网络操作系统)，实现网络之间协议转换的功能，故有时也称为协议转换器。

3.1.5 网络的安全性

一个计算机网络良好的安全性由保密性、完整性、可用性、真实性、实用性和占有性等 6 个指标组成。网络的安全性增加了信息安全与路由安全等方面的许多问题。ISO 7498-2 网络安全体系确定了多种基本安全服务和多种安全机制，分别在 OSI 的 7 层中实现网络的安全服务。网络的常用安全机制有标识与验证机制、网络访问控制机制、加密机制、信息完整性机制、认证和审计机制。

3.1.5.1 网络的信息安全技术

(1)访问控制与目录管理

资源的合法使用主要依靠网络管理员进行的必要的访问控制与目录管理，以保证网络资源不被非法用户使用或破坏，也保证数据资源不被非法读出或改写。

(2)数据加密

数据加密算法与技术有许多种。最有名的是 IBM 公司提出的 DES(数据加密标准)加密算法，这是一种对称加密的算法。RSA 算法是比较有名的一类非对称加密算法。所谓"非对称"是指把密钥分开，一个公开密钥用于加密，另一个专用密钥用于解密，这样便于密钥的管理。目前在 Internet 上提供了另一种 PGP 加密系统，这种系统把加密技术交给了公众使用。

(3)身份验证与鉴别

网络的信息安全技术的使用不仅对用户身份进行验证与鉴别，也可以对信息的真实可靠性进行验证与鉴别，用来解决冒充、抵赖、伪造或篡改等问题。除了标识用户加上口令的机制外，最常用的是"数字签名"技术。数字签名可以采用秘密密钥或公开密钥技术实现。签名算法可用 RSA、DSS 数字签名标准等。

(4)两个 TCP/IP 安全协议

Kerberos 和 SSL/SHTTP 都是完整的安全协议，作为保证通信信息安全的协议，应提供下列功能：双方确认认证、数据源的不可否认性、数据接收方的不可否认性、信息完整性保护、以密码保密信息内容以及防重传性(指一个信息包被获取后不能再被重新传输)。

3.1.5.2 防火墙技术

实现防火墙技术的新产品主要有两大类：一类是网络级防火墙，另一类是应用级防火墙。目前一种趋势是把这两种技术结合起来。

(1)网络级防火墙，也称为包过滤型防火墙。事实上这是一种具有特殊功能的路由器，采用报文动态过滤技术，能够动态地检查流过的 TCP/IP 报文或分组头，根据企业所定义的规则，决定禁止或者允许某些报文通过。允许通过的报文将按照路由表设定的路径进行信息转发，相应的防火墙软件工作在传输层与网络层中。

(2)应用级防火墙，也称应用网关型防火墙。大多采用代理服务机制，即采用一个网关来管理应用服务，在其上安装对应于每种服务的特殊代码(代理服务程序)，在此网关上控制与监督各类应用层服务的网络连接。常用的应用级防火墙大致有 3 种类型，分别适合于不同规模的企业内部网：双穴主机网关、屏蔽主机网关和屏蔽子网网关。它们的共同点是需要有一台主机(称为堡垒主机)来负责通信登记、信息转发和控制服务提供等任务。

3.1.6 Internet/Intranet

3.1.6.1 Internet

Internet 目前最主要的服务方式是采用浏览器以 Web 方式入网，以获得大部分的服务项目。主要提供的服务有：远程登录(Telnet)、文件传输(FTP)、电子邮件(E-mail)、网络新闻(News)、查找文件(Archie)、查找人员、以菜单方式浏览信息(Gopher)、按内容自动查找(WAIS)和全球范围的超媒体信息浏览服务(WWW)。

Internet 采用的是 TCP/IP 体系结构网络协议，加入 Internet 的任何网络必须支持 TCP/IP 协议，即在互联网层采用 IP 协议，在传输层采用 TCP(或 UDP)协议。因特网的应用层协议发展较快，目前拥有 Telnet、FTP、SMTP(简单邮件传输协议)、DNS(域名服务协议)、NSP(名字服务协议)和 HTTP(超文本传输协议)等一大批与 Internet 应用服务相关的协议，还有网络管理(如 SNMP)和网络安全等方面的协议。

IP 地址和域名是 Internet 的重要概念。连接到 Internet 上的每台计算机都必须有一个唯

一地址。Internet 的地址是用数字来表示的，称为 IP 地址。IP 地址又分为 5 类：A 类、B 类、C 类、D 类和 E 类。大量使用的仅为 A、B、C 这 3 类。A 类地址中第 1 个字节表示网络地址，最高位必须是"0"，而后 3 个字节表示该网内计算机的地址；B 类地址中前两个字节表示网络地址，最高位必须是"10"，后 2 个字节表示网内计算机的地址；C 类地址则是前 3 个字节表示网络地址，最高位必须是"110"，后 1 个字节表示网内计算机地址。A 类地址用于非常巨大的计算机网络，B 类地址次之，C 类地址则用于小型网络。

接入 Internet 的某台计算机要和另一台计算机通信就必须确切地知道其 IP 地址。为便于记忆，采用域名来识别 Internet 上的计算机。

3.1.6.2 Internet

现代的企业网络将以 Internet 作为基础。Internet 的中文名称为内联网。它是把 Internet 技术应用于企业局域网的产物，也是汲取了 Internet 和企业局域网两者的长处而有机地组成的新型企业网络。

Internet 技术的主要特征表现在：采用 TCP/IP 通信协议作为企业网络实现网络通信的基础；采用 Web 技术和 HTTP、SMTP、FTP、SNMP、DNS 等一系列公开协议标准作为企业网络构建时的基本技术；主要着眼于满足企业内部的使用，有控制地向外发布主页信息以及向企业外部提供有限的信息服务；在企业 LAN 的基础上增扩了 Web Server、Mail Server、FTP Server 和 DNS Server 等主要软硬件设施，构成完整的 Internet 网络；采用了有效的网络安全设施和网络管理平台，以保证 Internet 的运行效率和企业的根本利益。

3.1.6.3 Extranet

Extranet 的中文名称为外联网，是企业外部网的意思，是 Internet 概念的进一步延伸与扩展，即把 Internet 构建的技术应用到到了企业与企业之间的网络互联网上，成为企业之间合作的纽带或桥梁。Extranet 把采用 Web 技术构建的企业信息系统的适用范围进一步扩大到了一些特定的外部企业。

构建 Extranet 所采用的技术与 Internet 基本一样，即以 Web 技术、通信互联技术和数据库技术为核心。但是需要考虑外部企业与外部用户的访问需求及相关的安全保证。

3.1.7 客户机/服务器模式

客户机/服务器(Clien/Server)模式，简称 C/S 模式，是网络应用的重要方式。新的网络操作系统和多用户数据库系统基本上都支持 C/S 模式。

两层模式。从数据库管理系统的应用来看，在 LAN 上采用的 C/S 模式，至少拥有一台数据库服务器(DBMS Server)，为各台工作站存取公共数据提供后援支持。把应用任务中的程序执行内容划分成两部分：与数据库存取有关的部分由数据库服务器承担，与应用的人机界面处理、输入/输出或一部分应用的逻辑功能等有关的内容由客户端、工作站承担。

3 层方方式的 C/S 模式，即客户机--功能服务器--数据库服务器。把应用系统的软件相应地分为 3 层：客户机实体内驻留用户界面层软件，负责用户与应用之间进行对话的任务；功能服务器实体内存放有业务逻辑层软件，用来响应客户机的请求，完成相应的业务处理或复杂计算任务；数据库服务器实体内驻留有数据库服务层软件，用来执行功能层发送来的数据库操作，任务完成后逐层地返回给客户机上的用户。

3.1.8 网络计算与电子商务

网络计算是一种高效率和低成本的计算模型，这种模型适合于充分利用 Internet 的功能，

提供了一类一体化的解决方案。网络计算中所依托的基本技术，除了原有的 TCP/IP 网络协议体系结构与数据库技术外，目前主要有 Web 技术、HTTP 协议标准、CGI 公共网关接口、HTML、XML、Java 应用、瘦客户机技术和 Browser/Server 结构模式。

基于 B/S 模式的信息系统通常采用 3 层或更多层的结构：浏览器--Web 服务器--数据库服务器。以 Web 服务器作为系统的核心，用户端通过 Browser 向 Web 服务器提出查询请求(HTTP 协议方式)，Web 服务器根据需要再和数据库服务器发出数据请求。数据库服务器则根据检索与查询条件返回相应的数据结果给 Web 服务器，最终 Web 把结果翻译成 HTML 或各类脚本(Scripts)语言的相应格式发回至 Browser，用户通过 Browser 浏览所需结果。

电子商务(简称 eB)通常指的是商务贸易活动中各个环节的电子化，其内容可覆盖与商务活动有关的所有方面。eB 是一种典型的网络计算模式的应用，大体上可分为 3 大类，即政府贸易管理的电子化、企业与企业之间的电子商务、企业与个人消费者之间的交易(电子购物)。从技术角度来看，eB 将以 Internet/Intranet/Extranet 建设为基础，采用网络计算模式开展商务活动，网络传输与信息的安全性技术有着关键性的重要作用。

3.1.9 网络管理基本概念

网络管理有 5 大功能，即配置管理、失效与故障管理、性能管理、计费管理和安全管理。TCP/IP 协议组的 SNMP(简单网络管理协议)是网络管理协议的工业标准。目前的网络管理产品大多都支持 SNMP。

企业网主要采用网络管理平台和在该平台上开发出的相应网络管理应用软件进行全企业网范围内的有效网络管理，这些平台与软件都支持开放式的 SNMP 网络管理协议标准。网络管理平台是实现开放式集成化的网络集中统一管理的一项主要技术，提供了一种软件开发支持环境，在此环境中允许实现网络管理功能的集成。在企业网中，网络管理平台一般采用具有一定性能与规模的通用性网络管理工作站主机，其主要任务包括：搜索与查找网络拓扑，构建、显示与维护企业网的网络拓扑图，构建有关信息的中心数据库，保护、维护与更新该数据库的内容，对网络中出现的各类事件进行监视、记录、分析、处理和过滤，提供友善的网络管理界面和使用方式等。

企业网网络管理平台的代表产品有 IBM 公司的 N、etview fo AIX 或 TME10、CA 公司的 Unicentre TNG|HP 公司的 Open View、Sun 公司的 Net Manager 和 Cabletron 公司的 Spectrum 等。网络设备的主要供应厂商也提供相应的网络管理软件，比如 Cisco 公司的 Works，3Com 公司的 Transcend/Ondemand，Bay Networks 公司的 Qptivity。

3.2 试题分析

网络基础试题主要考查基本知识，难度不大，但涉及面广。从历年试题统计(见表 4-1)来看，自 1994 年以来，每年都有专门考查网络基础知识的试题，主要考查各类网络协议和标准，其中许多试题都涉及到 OSI(开放网络互连)参考模型，与 Internet 有关的的内容也是反复考查的知识点，可以说，Internet 技术、网络协议和标准是考查重点。至今没有网络安全性、客户机/服务器结构、网络计算方面的试题，希望能引起注意。

表 4-1　历年高级程序员级网络基础试题统计

试题	考查知识点
1990 年试题 10	串行通信标准
1992 年试题 7	微机局域网操作系统
1994 年试题 10	以太网
1995 年试题 5	协议概念
1995 年试题 9	OSI 模型
1996 年试题 4	Internet 常识
1997 年试题 3	网络管理
1997 年试题 11	TCP/IP 协议与 Internet
1998 年试题 10	涉及电子商务和网络体系结构
1998 年试题 11	网络互连设备与 OSI 模型
1999 年试题 13	宽带网络接入技术
2000 年试题 13	网络协议，涉及网络管理、移动上网、拨号上网、路由选择

试题 1（2000 年试题 13）

从供选择的答案中，选出应填入下面叙述中{ }内的最确切的解答，把相应编号写在答卷的对应栏内。如：

A：用于网络管理；

B：用于移动电话无线上网；

C：可用于家庭拨号上网；

D：一种面向比特的数据链路通信规程；

E：一种路由选择协议。

供选择的答案

A～C：①SNMP ②PPP ③RIP ④WAP

D、E：①OSPF ②HTTP ③HDLC ④RARP

【解析】

本题考查常用的网络协议知识，涉及最新的移动无线上网技术。

这里简单补充一下有关知识。

SNMP 是基于 TCP/IP 的简单网络管理协议。

PPP 是用于拨号接入，实现点对点通信的远程访问协议。

RIP(路由信息协议)是最常用的内部路由协议，多用于 UNIX 系统中。

OSPF(开放式最短路径优先)是一种为 TCP/IP 开发的连接状态协议，也是一种内部路由协议，多用于路由器中。

WAP 是用于移动电话(手机)无线上网的协议。

HTTP 是超文本传输协议。

HDLC 是国际标准化组织(ISO)制定的高层数据链路控制协议。

RARP(反向地址转换协议)用于动态实现 IP 地址向物理地址的转换。

【答案】A：① B：④ C：② D：③ E：①

试题 2（1999 年试题 13）

从供选择的答案中，选出应填入下面叙述中{ }内的最确切的解答，把相应编号写在答卷的对应栏内。

向端用户提供尽可能宽带的网络接入是引起人们广泛关注的技术。A 只能提供 128kbit/s 的接入数据速率， B 则是通过电话双绞线向端用户提供更高信息传输带宽的一种接入技术，而采用 C 和电缆调制解调器(Cable Modem)也可获得和后者同样数量级的接入带宽。第 3 代无线通信的 D 可提供高达 2Mbit/s 的接入数据速率。光纤到户，即 E ，则是将来的一种

发展方向。

供选择的答案

A、B：①B-ISDN ②N-ISDN ③CDMA ④ADSL

C、D：①HFC ②GSM ③CDMA ④HDSL

E：①FDDI ②FTTH ③FTTC ④FTTB

【解析】

本题考查网络接入技术知识，涉及的内容比较新、比较多。这里用较大的篇幅详细介绍试题中涉及的若干知识点。

B-ISDN(Broad Integrated Service Digital Network)，中文名称是宽带综合业务数字网。B-ISDN 所涉及的关键技术主要有 ATM 宽带交换机、同步光纤网/同步数字系列和光纤同步网。B-ISDN 以光纤为传输媒介，这一方面保证了所提供的业务质量，同时也减少了网络运行中的诊断、纠错、重发等环节，从而提高了网络的传输速率，B-ISDN 的用户/网络接口的速度可高达 150Mbit/s 或 600Mbit/s。另外，B-ISDN 可以提供多种高质量视频信息，如增强型可视电话、高清晰度电视等。B-ISDN 还可以将电信与广播、电视合为一体，即将广播电视业务也纳入到电信业务之中。

根据不同形式宽带通信及其应用，B-ISDN 有两种主要的业务类别：交互型业务和分配业务。交互型业务又分为 3 种业务：会话型业务、电子信箱业务和检索型业务。分配业务又由两种业务来表示：客户不能进行单独演示控制的分配型业务和客户能够进行单独演示控制的分配型业务。

B-ISDN 向客户提供的业务将有以下特点：①高度的综合性；②图像通信占有重要的地位；③大量使用多媒体业务；④客户从网络得到的业务将是大量的、多种多样的；⑤终端工作速率可变；⑥终端将一机多用。

N-ISDN(Narrow Integrated Service Digital Network)，中文名称是窄带综合业务数字网，又称"一线通。""一线通"采用数字传输和数字交换技术，利用现有的电话网将电话、传真、数据、图像等多种业务综合在一起进行传输和处理，实现了端到端的数字连接，可实现会议电视、远程教学、远程医疗和多点采集。

"一线通"在上网的同时还能利用另一个 B 通道进行电话交流或收发传真。若将两个 B 通道捆绑成一个通道使用，速度将达 128kbit/s，

N-ISDN 主要有基本速率扫口和基群速度接口两种接口。基本速率接口主要面向现有电话网的普通用户、上网用户以及小型单位，提供 2B+D(144kbit/s)的基本速率，其中 B 信道为 64kbit/s，D 信道为 16kbit/s。基群速率接口主要面向企业用户和集团用户，提供会议电视等高速通信业务，速率为 30B+D(2Mbit/s)，其中 B 信道为 64kbit/s，D 信道为 16kbit/s(注：B 信道为传送信息的信道，D 信道为传送信令的信道。)

CDMA(码分多地址移动通信技术)与 FDMA(频分多地址移动通信技术)和 TDMA(时分多地址移动通信技术)相比，CDMA 具有许多独特的优点，其中一部分是扩频通信系统所固有的，另一部分则是由软切换和功率控制等技术带来的。CDMA 移动通信网是由扩频、多址接入、蜂窝组网和频率再用等几种技术结合而成，含有频域、时域和码域三维信号处理的一种协作，因此它具有抗干扰性好，抗多径衰落，安全保密性高，同频率可在多个小区内重复使用，所要求的载干比(C/I)小于 1，容量和质量之间可做权衡取舍等属性。系统容量大，理论上 CDMA 移动网比模拟网大 20 倍。实际要比模拟网大 10 倍，比 GSM 要大 4～5 倍。这些属性使 CDMA 比其他系统有更大的优势。

CDMA 移动通信网的关键技术有：①功率控制技术；②PN 码技术；③RAKE 接收技术；④软切换技术；⑤话音编码技术。

xDSL 是 DSL(Digital Subscriber Line)的统称，意即数字用户线路，是以铜电话线为传输介质

的传输技术组合。DSL 技术在传统的电话网络(POTS)的用户环路上支持对称和非对称传称模式，解决了经常发生的在网络服务供应商和最终用户间的"最后 1 公里"的传输瓶颈问题。由于电话用户环路已经大量铺设，如何充分利用现有的铜缆资源，通过铜质双绞线实现高速接入就成为业界的研究重点，因此 DSL 技术很快就得到重视，并在一些国家和地区得到大量应用，其中"x"代表着不同种类的数字用户线路技术。各种数字用户线路技术的不同之处，主要表现在信号的传输速率和距离以及对称和非对称的区别上。DSL 技术主要分为对称和非对称两大类。

第一类：对称 DSL 技术

(1)HDSL--High-bit-rate DSL(高速度 DSL)

HDSL 是 xDSL 技术中最成熟的一种，已经得到了较为广泛的应用。其特点是：①利用两对双绞线传输；②支持 Nx64kbit/s 和种速率，最高可达 E1 速率；③HDSL 是 T1/E1 的一种替代技术，主要用于数字交换机的连接、高带宽视频会议、远程教学、蜂窝电话基站连接、专用网络建立等。

与传统的 T1/E1 技术相比，HDSL 具有以下优点：①价格便宜；②容易安装(T1/E1 要求每隔 0.9～1.8km 就安装一个放大器，而 HDSL 可在 3.6km 的距离上传输而不用放大器。)

(2)SDSL--Symmetric DSL(对称 DSL)

①利用单对双绞线；②支持多种速率到 T1/E1；③用户可根据数据流量，选择最经济合适的速率，最高可达 E1 速率，比用 HDSL 节小一对铜线；④在 0.4mm 双绞线上的传输距离可达 3km 以上。

(3)MVL--Multiple Virtual Line(多虚拟数字用户线)

MVL 是 Paradyne 公司开发的低成本 DSL 传输技术。

①利用一对双绞线；②安装简便，价格低廉；③功耗低，可以进行高密度安装；④用与 ISDN 技术相同的频率段，对同一电缆中的其他号干扰非常小；⑤支持语音传输，在用户端无需语音分离器；⑥支持同一条线路上同时连接多至 8 个 MVL 用户设备，动态分配带宽；⑦上/下行地速率可达 768kbit/s；⑧传输距离可达 6km。

第二类：非对称 DSL 技术

ADSL-Asymmetric DSL(非对称 DSL)能够在现有的铜双绞线，即普通电话线上提供高达 8Mbit/s 的高速下行速率，远高于 ISDN 速率；而上行速率为 1Mbit/s，传输距离达 3km～5km。其优势在于可以充分利用现有的铜缆网络(电话线网络)，在线路两端加装 ADSL 设备，即可为用户提供高宽带服务，由于不需要重新布线，降低了成本，进而减少用户上网的费用。

ADSL 的主要特点：①具有很高的传输速率，ADSL 能在现有普通电话线上以很高的速率传输数据，下行达到 8Mbit/s，上行达到 1Mbit/s；②上因持网和打电话互不干扰；③提供多种先进服务；④独享带宽，ADSL 的每个用户都独享 8Mbit/s 带宽；⑤费用低廉；⑥安装方便快捷。

另外还有 VDSL(Very-high-data-rate DSL)等 DSL 技术，这些技术可在较短的距离上提供极高的传输速率，但应用还不是很多。

所谓光接入网(OAN)就是采用光纤传输技术的接入网，泛指本地交换机或远端模块与用户之间采用光纤通信或部分采用光纤通信的系统。通常，OAN 指采用基带数字传输技术，并以传

输双向交互式业务为目的接入传输系统，将来应能用数字或模拟技术升级传输带宽广播式和交互式业务。

ONU 的作用是为光接入网提供直接的或远端的用户侧接口。按照 ONU 在接入网中所处的位置不同，可以将 OAN 划分为几种基本不同的应用类型，即光纤到路边(FTTC)、光纤到楼(FTTB)以及光纤到家(FTTH)和光纤到办公室(FTTO)。

(1)光纤到路边(FTTC)
在 FTTC 结构中，ONU 设置在路边的入孔或电线杆上的分线盒处，此时从 ONU 到各个用户之间的部分仍为双绞线铜缆。若要传送宽带图像业务，则这一部分可能会需要同轴电缆。FTTC 结构主要适用于点到点或点多点的树形--分支拓扑结构，用户多为居民住宅用户和小企事业用户，典型用户数在 128 个以下，经济用户数正逐渐降低到 8～32 个乃至 4 个左右。
(2)光纤到楼(FTTB)
FTTB 也可以看作是 FTTC 的一种变形，不同处在于将 ONU 直接放到楼内(通常为居民住宅公寓或小企事业单位办公楼)，再经多对双绞线，将业务分送给各个用户。FTTB 是一种点到多点结构，通常不用于点到点结构。FTTB 的光纤化程度比 FTTC 更进一步，光纤已铺设到楼，因而更适于高密度用户区，也更接近于长远发展目标。预计会获得越来越广泛的应用，特别是应用于那些新建工业区或居民楼以及带宽传输系统共处一地的场合。
(3)光纤到家(FTTH)和光纤到办公室(FTTO)
在原来的 FTTC 结构中，如果将设置在路边的 ONU 换成无源光分路器，然后将 ONU 移到用户家，即为 FTTH 结构。如果将 ONU 放在大企事业用户(公司、大学、研究所、政府机关等)终端设备处，并能提供一定范围的灵活的业务，则构成所谓的光纤到办公室(FTTO)结构。由于大企事业单位所需业务量大，因而 FTTO 结构发展很快。考虑到 FTTO 也是一种纯光纤连接网络，因而可以归入 FTTH 一类的结构。然而，两者的应用场合不同，结构特点也不同。FTTO 主要用于大企事业用户，业务量需求大，因而在结构上适于采用点到点或环形结构。而 FTTH 用于居民住宅用户，业务量需求大，因而在结构上适于采用点到点或环形结构。而 FTTH 用于居民住宅用户，业务量需求很小，因而经济的结构必须是点到多方式。FTTH 和 FTTO 主要特点可以总结如下：由于整个接入网是全透明光网络，因而对传输制式(例如 PDH 或 SDH，数字或模拟等)、带宽、波稀薄和传输技术没有任何限制，适于引入新业务，是一种最理想的业务透明网络，是接入网发展的长远目标。
HFC(Hybrid Fiber Coax，即光纤同轴混合)是在有线电视网(CATV)的基础上发展起来的一种宽带接入技术，其网络结构为树型或总线型。从前端至小区光节点采用光纤传输，从光节点到用户终端采用同轴电缆，小区光节点所管辖的用户数一般在 500 左右(理想情况)，整个光节点内部网可提供约 1GHZ 的带宽，由光节点内的所有用户共享。在 HFC 前端、语音、数据、视频等多媒体信息经调制后与电视射频信号一起伟输到用户端，经过分离器提取多媒体信息，并恢复成数据包形式，通过类似 IEEE 802.3 的共享介质接入机制 MLAP(MAC Level Access Protocol)进入用户终端设备。此外，前端的带宽控制器可通过编程方式对用户 Cable Modem 进行频率分配控制。
Cable Modem 是 HFC 中的关键设备，按照其上下行信道传输速率是否对称分为两类：一是上下信道对称，双向通信速率最高可达 100Mbit/s，适用于高速 LAN 互联；另一种是上下信道不对称，上行速率最高为 768kbit/s，下行速率最高可达 30Mbit/s，适用于高速 Internet 接入等。
HFC 的特点是除了能提供原有的视频信号外，还能提供双向语音、数据、视频等宽带多媒体信息业务，现阶段具有一定的发展前景，但属于过渡型技术。
【答案】A：② B：④ C：① D：③ E：②

试题 3（1998 年试题 10）

本题是一道综合题，涉及电子商务和网络体系结构，参见第三章 3.5 节试题 1。

试题 4（1998 年试题 11）

从供选择的答案中，选出应填入下面叙述中{ }内的最确切的解答，把相应编号写在答卷的

对应栏内。
有多种设备可以实现不同网段或网络之间的互连，互连设备通常可按工作在 OSI 模型中的层次来划分。在物理层实现互连的称为 A ；在数据链路层实现互连的称为 B ；在网络层实现互连的称为 C ；在运输层及以上高实现互连的设备称为网关或 D 。E 也是一种用来构造局域网的常用设备，通常可以用双绞线把服务器与 PC 客户机等连入 E 。
供选择的答案
A～E：①集线器 ②协议转换器 ③网桥 ④路由器
⑤网关 ⑥转发器
【解析】
网络互连需要通过一个中间设备或中间系统，术语称之为中继系统(Relay System)，根据中继系统在网络中所处的层次，可以有以下几种中继系统：
物理层中继系统，即转发器(Repeater)；
数据链路层中继系统，即网桥或桥接器(Bridge)；
网络层中继系统，即路由器(Router)；
网桥和路由器的混合系统，即桥路器(Brouter)；
网络层以上的中继系统，即称为网关(Gateway)。网关也称为网间连接器、信关或联网机。用网关连接两个不兼容的系统要在高层进行协议转换，因此，网关也称为协议转换器。
双绞线以太网问题问题和集线器(Hub)一块使用的，集线器也是一种用来构造局域网的常用设备，通常可以用双绞线把服务器与 PC 客户机等连入集线器。
【答案】A：⑥ B：③ C：④ D：② E：①

试题 5（1997 年试题 3）

从供选择的答案中，选出应填入下面叙述中{ }内的最确切的解答，把相应编号写在答卷的对应栏内。
当网络用户通过网络与另一台主机 X 通信，发现响应太慢时，可运行 A 程序，把一个分组发向主机 X，通过查看所返回的分组首部的 B ，发现问题的所在，并采取相应的措施。
对于一个大中型网络，需要有一个网络管理系统进行管理。当前流行的各类网管平台软件都支持 C 协议。驻留在 D 上的网管平台软件可通过该协议软件调阅被管的网络结点内的管理信息库中的内容。有若干常用的网络平台软件，但 E 不是网管平台软件。
供选择的答案
A：①Browser ②Bitware ③Ping ④Handshaking
B：①地址 ②时间戳 ③标识码 ④校验码
C：①MAP ②SNMP ③MHS ④FTAM
D：①数据库服务器 ②通信服务器 ③主路由器 ④网络管理工作站
E：①Net Manager ②Open View ③Sun View ④Net View
【解析】
该题主要测试考生对网络基本概念和常用方法的掌握程度。
在使用 TCP/IP 协议的网络中，Ping 是一个常用的网络测试程序，它发送一个 ICMP(Internet Control Message Protocol)的请求应答包给一个主机网关，以期对方能发回一个 ICMP 的应答包。发送方可以通过检查返回包首部的时间戳来获得应答时间的长短信息。因此当网络用户通过网络与另一台主机 X 通信，发现响应太慢时，可运行 Ping 程序，把一个分组发向主机 X，通过查看所返回的分组首部的时间戳，发现问题的所在，并采取相应的措施。
IP 提供的是不可靠非面向连接的数据报服务。为了使网上的计算机能够检测报告差错，或提供有关意外情况的信息，设计者在 IP 协议中提供了一种特殊作用的报文制，即 ICMP。

它位于 IP 数据报的数据部分，同 IP 数据报的数据信息一样在网上传输，但其目的地不是目的计算机上的一个用户进程，而该计算机上的 IP 软件。当一个有错误的 ICMP 报文到达时，IP 软件模块就处理本身的问题，不再将其提供给高层。

Browser 是在网络上进行信息浏览的软件--浏览器，目前常用的有 Netscape 和 Internet Explorer。Bitware 是一个利用调制解调器(Modem)收发传真的软件。Handshaking(握手)是网络上进行信息传输时传输双方为达到同步所做的操作的通用说法。

目前常用网络平台软件有 Sun 公司的 Net Manager，HP 公司的 Open View，IBM 公司的 Net View。Sun View 不是网络平台软件，它是 Sun 公司的操作系统 SunOS 上的视窗系统。一个大中型网络，需要有一个网络管理系统进行管理。网络管理最通用的协议是 SNMP(Simple Network Management Protocol)，即简单网络管理协议。当前流行的各类网管平台软件都支持这一协议，它主要用于在网络中不同站点之间交换与网络管理有关的信息，通过网络管理软件可监视和控制网络的配置及运行情况，测试、记录和统计网络中各处的数据流量和差错情况等。当前流行的各类网管平台软件都支持 SNMP 协议。驻留在网络管理工作站上的网管平台软件可通过该协议软件调阅被管的网络结点内的管理信息库中的内容。有若干常用的网络平台软件，全 Sun View 不是网管平台软件。

MHS(Message Handling System)是网络上传送标准电子邮件的报文处理系统。FTAM(File Transfer Access And Management)是指在网络上不同主机间的文件传送访问和管理。

MAP(Manufacturing Automation Protocol)是指制造自动化协议，是由美国通用汽车公司提出的一种用于生产自动化的局域网协议，它为众多来自不同厂家的各种设备集成提供一个标准的、开放的通信网络环境。

网络上存在不同功能的服务器：数据库服务器提供对数据库数据的管理，并为数据库应用程序提供数据访问服务；通信服务器一般提供很多 I/O 端口，可以同时为许多客户提供通信服务；网络上的路由器负责不同网络、不同协议之间数据包的转发。这些设备与网络管理平台软件没什么关系。网络管理工作站提供对所负责的网络的监控，所以驻留在网络管理工作站上的网管平台软件可通过该软件调阅被管的网络结点内的管理信息库中的内容。

【答案】A：③ B：② C：② D：④ E：③

试题 6（1997 年试题 11）

从供选择的答案中，选出应填入下面叙述中{ }内的最确切的解答，把相应编号写在答卷的对应栏内。

计算机的发展已进入了网络计算的新时代。Internet 是目前世界范围内最大的互联网。如此多的各种计算机之所以能通过 Internet 相互通信，是因为它们遵循了一套共同的 Internet 协议。这套协议的核心是 A ，在其上建立的无连接的运输层协议是 B ，万维网 WWW 上超文本传输遵循 C ，电子邮件传输遵循 D ，Ethernet 与 Internet 连接时要用到 E 。

A～E：①TCP ②IP ③EIDFACT ④HDLC

⑤ARP ⑥UDP ⑦FTP ⑧ICMP

⑨SMTP ⑩HTTP

【解析】

Internet 是用一种路由器(专用计算机)将网络互联组成的。互联的计算机需要软件才能通信，通信双方必须遵循的规则称为协议。Internet 中使用的一个关键协议是网际协议 IP(Internet Protocol)，IP 详细定义了计算机之间进行通信应遵循的规则，例如定义了分组的格式，以及路由器如何将每一个分组递交到目的地等。

Internet 中的分组称为 IP 数据报。为了使路由选择和数据报的递交成为可能，连接到 Internet 上的每台计算机都必须被指定一个唯一的地址，该地址称为 IP 地址。在 IP 上层建立的传输

层有两个并列的协议：ICP 和 UDP，其中 TCP(Transport Control Protocol)是面向连接的，UDP(User Datagram Protocol，用户数据报协议)是无连接的。一般情况下，TCP 和 UDP 并存于 Internet 中，TCP 提供高可靠性服务，其协议非常复杂，执行效率低，UDP 提供高效率服务，其协议相对简单得多。

WWW 是 World Wide Web 的缩写，是把信息检索技术和超文本(Hyper Text)技术相结合而形成的环球信息系统。超文本是一种描述信息的方法，其文本中的被选词可以被扩展，以提供该词所关联的其他信息，即把这些词"链接"到其他文档上，这些文档可能是以文本、图片或其他形式提供的。超文本技术是指在一文档中的某些词暗藏着与其他文档的联系，通过这些词(如用鼠标单击这些词)就能迅速找到相关文档，并可把它调来阅读或作其他处理。在 WWW 中，与某一文档相关联的文档可以放置在世界各地的其他计算机内。WWW 的成功在于制定了一套标准的容易掌握的超文本标记语言 HTML(Hyper Text Markup Language)、统一资源定位器 URL (Uniform Resource Locator)和超文本传输协议 HTTP(Hyper Text Transport Protocol)。WWW 提供一种高级浏览服务，采用客户机/服务器(Client/Server)模式工作。Internet 上的一些计算机运行着 WWW 服务器程序，它们是信息的提供者，在用户计算机上运行着 WWW 客户程序，帮助用完成信息咨询。超文本传输协议 HTTP 是 WWW 客户机和服务器在网际上响应用户请求并传输信息的协议。当用户激活一个"链接"后，服务器使用 HTTP 通过送回约定好格式的文件来作出响应，客户机通过一个浏览器来显示响应信息。

TCP/IP 协议提供两个电子邮件传输协议：MTP(Mai Transfer Protocol，邮件传输协议)和 SMTP(Simple Mail Transfer Protocol，简单邮件传输协议)。在 Internet 中，大部分电子邮件由 SMTP 发送，其特点就是简单，它只定义邮件如何在传输系统中通过发方和收方之间的 TCP 连接进行传输而不规定其他操作。

在 Internet 中，IP 层以上各层都使用 IP 地址，而 IP 地址和计算机的物理地址是不同的，在物理网络内部，仍使用各自原来的物理地址，这样，在 Internet 中存在两种地址，二者之间要建立映射关系，包括从 IP 地址到物理地址的映射和从物理地址到 IP 地址的映射。Ethernet 网络具有广播能力，与 Internet 连接时，遵循 ARP 协议(Address Resolution Protocol)。ARP 的原理如下：假设在某广播型网络上，主机 A 要得到主机 B 的物理地址，首先主机 A 广播一个 ARP 请求报文，请求 Internet 地址 IB 的主机回答其物理地址 PB。网上所有主机(包括 B 机)都将收到该请求，但只有 B 机能够识别出自己的 IB 地址，并向 A 发回应答，回答 B 机的物理地址 PB，以后就可直接在 A 机和 B 机之间传输数据了。

【答案】A：② B：⑥ C：⑩ D：⑨ E：⑤

试题 7（1996 年试题 4）

从供选择的答案中，选出应填入下面叙述中{ }内的最确切的解答，把相应编号写在答卷的对应栏内。

Internet 提供的服务有通信、远程登录、浏览和检索等。A 直接用于人际通信， B 用于远程登录。 C 不是浏览软件。在浏览软件中， D 不支持 HTML，E 是目前微机上最常使用的浏览器。

供选择的答案

A、B：①WWW ②E-mai ③URL ④TCP/IP

⑤Telnet ⑥Lycos ⑦HTML ⑧PPP

C～E：①Hotjava ②Netscape ③Mosaic ④Lycos ⑤Gopher

【解析】

Internet 即国际互连网络，现一般译为因特网。Internet 是目前世界上最大、用户最多的互联网络。TCP/IP 是传输控制协议/互连网协议，Internet 使用 TCP/IP 协议，把不同的局域网和

广域网通过路由器互相连接，形成一个全球性大网。
Internet 提供各种服务职通过信、远程登录、新闻组、电子邮件、文件传输、信息的测览与检索等。电子邮件(E-mail)用网络用户间的通信；FTP 以客户机/服务器方式为用户提供文件舆服务，用户可以从文件服务器上下载自己感兴趣的文件，也可以将自己的文件传送到服务器上 ;远程登录 Telnet 使得用户可通过运行 Telnet 把自己的计算机作为服务器的一个操作终端，调用服务器的共享资源。
Internet 上的海量信息，使得用户查找自己感兴趣的信息变得费时费力，因而好的浏览器会成为用户上网的好帮手，节约上网时间，提高上网效率。Gopher 是一咱交互式、菜单驱动的 Internet 浏览器。WWW(World Wide Web 译为万维网，它使用超文本舆协议 HTTP(Hyper Text Transport Protocol)向用户提供服务，在 Internet 上有很多 WWW 服务器，提供 WWW 服务。要访问这些服务器，用户可在客户机上运行 WWW 浏览器，该程序可以解释和显示从 WWW 上找到的超文本文档。目前的 WWW 浏览器有 Mosaic、Netscape Navigator、Hotjava 等，Netscape Navigator 是目前微机上较流行的 WWW 浏览器。Hotjav 可运行于 WindowsNT、Windows95 之上，支持用 java 编制的可执行信息，上述浏览软件中，只有 Gopher 不支持 HTML。
HTML(Hyper Text Markup Language)超文本标记语方是一种描述语言，用它编制 WWW 上的超文本文档，支持超文本链接，可以从一个文档指向另一个文档，或是从文档的一部分指向另一部分。PPP(Point-to-Point Protocol)是一个点到点的信息舆协议，是用户计算机通过 Modem 与 Internet 相连的一种方式。URL(Universal Resource Locator)统一资源定位器，用于确定 Internet 上的地址。Lycos 是 Internet 上一个信息检索工具，不是浏览器。
【试题】A：②B：⑤C：④D：④E：②

试题 8（1995 年试题 5)

从供选择的答案中，选出应填入下面叙述中{ }内的正确答案，把编号写在答卷的对应栏内。
协议是一组 A ，它有助于 B 之间的相互理解和正确进行通信。协议中有 3 个鲞关键因素。其中 C 定义数据的表示形式；D 则能使数据管理所需的信息得到正确理解，E 则规定了通信应答信号之间的间隔和先后关系。
供选择的答案：
A、B：①软件 ②外部设备 ③通信实体 ④时钟 ⑤约定的规则 ⑥寄存器组 ⑦存储器 ⑧CPU
C、D、E：①媒体 ②语义 ③文本 ④语言 ⑤时序 ⑥编码 ⑦语法⑧波特率⑨文件
【解析】
协议就是两个计算机之间通信时对传送信息内容的理解、信息表示形式以及各种情况下的应答信号都必须遵循的共同约定。在 ARPA 网中将协议按功能分成若干层次，分层标准和各层中采用的具体协议的综合就构成了网络的体系结构。
为了减少协议设计的复杂性，大多数网络都按 OSI 的 7 层结构来组织，每一层都有协议，但不管是何种网络，哪一层的协议，作为协议必须具备 3 个要素，即语法、语义和时序。语法是一种形式化的表示，它定义了传输数据的表示形式；语义则定义了数据的实际含义，从而使数据管理所需的信息得到正确理解 ;时序则规定了通信实体之间应答信号的相互间隔和顺序关系。
【答案】A：⑤ B：③ C：⑦ D：② E：⑤

试题 9（1995 年试题 9)

从供选择的答案中，选出应填入下面关于 OSI 叙述中{ }内的正确答案，把编号写在答卷的

对应栏内。
国际标准化的开放系统互连(OSI)参考模型共分 7 层。其中，处理系统之间用户信息的语法表达形式问题的是 A 层；规定通信双方相互连接的机械、电气、功能和规程特性的是 B 层；向用户提供各种直接服务，如文件传送、电子邮件、虚拟终端等的是 C 层；通过校验和反馈重发等方法将原始不可靠的物理连接改造成无差错的数据通道的是 D 层；负责通信子网中从源到目标路径选择的是 E 层。
供选择的答案：
A～E 层：①物理层 ②数据链路层 ③网络层 ④运输层
⑤会话层 ⑥表示层 ⑦应用层
【解析】
在国际标准化组织公布的开放系统互连基本参考模型中，详细规定了该模型的 7 个层次，并对每一层在网络传输中所各自负责的具体任务做了详尽的描述。
OSI 模型共分 7 层，自上而下依次为应用层、表示层、会话层、传输层、网络层、数据链路层和物理层。下层实体为上层实体提供服务，上层实体为下层实体提供接口。具体请参见相关的知识点综述。
【答案】A：⑥ B：① C：⑦ D：② E：③

试题 10（1994 年试题 10）

从供选择的答案中，选出应填入{ }内的正确答案，把编号写在答卷的对应栏内。
以太网遵循 IEEE802.3 标准，用粗缆组网时每段不能大于 A 来。超过上述长度时，要分段，段间用 B 相连。同时，整个网的总长度不能大于 C 米。若总长度超过上述长度，则需分成两个网，网之间用 D 相连。这是在 ISO 的 OSI 的模型中 E 层的连接。
供选择的答案
A～C：①50 ②100 ③185 ④500
⑤1000 ⑥2500
B、D：①网络适配 ②重发器 ③调制解调器 ④网桥
E：①物理 ②数据链路 ③网络 ④运输
【解析】
以太网(Ethernet)遵循美国电气电子工程师协会 IEEE 制定的 802.3 标准，目前已成为国际标准化组织 ISO 的国际标准 ISO 802.3，是一种带冲突检测的载波侦听多路访问(CSMA/CD)技术，其网络物理介质可以采用双绞线、细同轴电缆、粗同轴电缆或光纤。10Base5 表示网络是采用同轴电缆的标准构建而成；10Base2 表示细同轴电缆；10BaseT 表示双绞线；FDDI 表示光纤。
受发送器的功率影响以及接收器的分辨能力及媒体上信号传输衰减和畸变的限制，10Base5 标准规定每段同轴电缆的长度不能超过 500m。超过上述长度时，段与段之间要用重发器(Repeater)相连。Repeater 实际上是把从一段中接收到的信号放大后，再向另一段发送出去，有时也译为中继器。
重发器不能无限制地使用。由于以太网冲检测的需要，全网内端到端最大的信号传播延迟时间不能超过一定的限制值，这使得整个网的总长度不能大于 2500m。若总长度超过了 2500m，要分成两个网，网之间再用网桥(Bridge)来相连，这是在 ISO 的开放系统互连(OSI)七层模型中数据链路层的相连，用重发器相连则是在物理层的连接。两个网络在网络层连接通常由路由器(Router)来完成，在运输层以上的高层连接的网际互连设备则称为网关(Gateway)。网络适配器是将终端接入网络的设备，如常见的各种网卡；调制解调器(MODEM)则是将离散的数字信号转换成连续的模拟信号以便在模拟信道上传输的设备。

【答案】A：④ B：② C：⑥ D：④ E：②

试题 11（1992 年试题 7）

从供选择的答案中选出应填入下面关于网络操作系统叙述中{ }内的正确答案，把编号写在答卷的对应栏内。

微机局域网操作系统能支持网络中的服务器与工作站的通信连网。通常此类操作系统可由 A 、B 和 C 所组成。A 是专门提供网络服务的。设置的网络管理软件，它直接与网络服务器硬盘和网卡 D 连接，对用户访问服务顺的权限和数据加密进行管理，对多个用户访问服务器共享资源进行管理。C 用于实现多个网络的互连服务。此类操作系统中的协议模块与 OSI 协议建立起对应关系，比如： E 对应于会话层。

供选择的答案

A～E：①路由器软件 ②NETBIOS 仿真程序 ③系统管理服务程序

④网络文件服务器程序 ⑤驱动程序 ⑥工作站 Shell 程序

【解析】

微机局域网操作系统能支持网络中的服务器与工作站的通信连网，通常此类操作系统由网络文件服务器程序、工作站 Shell 程序和路由器软件所组成。

网络文件服务器程序是专为提供网络服务设置的网络管理软件，它直接与网络服务器硬盘和网卡驱动程序连接，对用户访问服务器的权限和数据加密进行管理，对多个用户同时访问服务器共享资源进行管理。

工作站 Shell 程序驻留在各结点上，为本地的应用提供各类网络服务，路由器软件用于实现多个网络的互连服务。此类操作系统中的协议模块与 OSI 建立起对应关系，比如，NETBIOS 仿真程序对应于会话层。

【答案】A：④ B：⑥ C：① D：⑤ E：②

试题 12（1990 年试题 10）

从供选择的答案中选择应填入下面叙述中的{ }内的正确答案，把编号写在答卷的对应栏内。

RS-232-C 是目前常见的一种接口标准，它是由 A 提出制定的。该标准在 OSI 模型中属于 B 层协议标准，通过 RS-232-C 来连接两个设备最少要连接 C 条线。这个标准的设计数据速率是处理 D bit/s。在 Dbit/s 条件下，采用 RS-422 标准来代替 RS-232-C，连接设备间的距离可扩展到约为原有的 E 倍。

供选择的答案

A：①CCITT ②EIA ③IFIP ④IEEE

B：①物理 ②数据链路 ③网络 ④运输 ⑤会话

C、E：①2 ②3 ③4 ④7

⑤10 ⑥25 ⑦80 ⑧100

D：①4800 ②9600 ③19200 ④20000

⑤6400

【解析】

RS-232-C 是由电子工业协会(EIA，Electronic Industries Association)制定的数据终端设备和数据电路端接设备连接的物理接口标准，属于国际标准化组织 ISO 的开放系统互连(OSI)模型中的最低层，即物理层的协议标准。它规定了接口的机械、电气和功能特性。

RS-232-C 规定的机械特性是 25 针的插头/座，减去一些未定义的针外，实际上只定义了 20 根针的功能。用来连接两个设备至少要连接 3 根钱线，即信号地、发送数据和接收数据线。在采用 RS-232-C 连接计算机与终端的场合就只使用这 3 根线。

RS-232-C 的设计数据速率是 2000bit/s，连接设备间的距离也有规定。为了实现更高的数据速率和更远的距离连接，EIA 又制定了另一个 RS-422 标准。该标准的电气特性与 RS-232-C 不同，不用公共地，采用双线平衡传输的方式，在同样数据速率条件下，可达到较远的传输距离。在数据速率是 2000bit/s 条件下，连接设备间的距离可扩展到原有 RS-232-C 标准的约 80 倍。

【答案】A：②： B：① C：② D：④ E：⑦

四 专业英语试题精解

根据考试大纲的要求，高级程序员应具有大学毕业程度的英语词汇量，能正确阅读和理解计算机领域的英文文献，相当于大学四级英语水平。

解答专业英语试题的关键在于熟悉相关的计算机专业英语。试题以考查词汇为主，兼考语法。复习时应多读一些计算机方面的英语时文，积累一些计算机专业词汇。解题时，一般先考虑语义，后考虑语法。

1990～2000 年英语试题涉及的主题分布情况见表 5-1。总的来说，大多涉及流行的软件技术，时代感很强。

表5-1　历年高级程序员级专业英语试题统计

试题	所涉及的主题
1990年试题 12、13	程序设计语言（programming languages）、集成软件（integrated software）
1991年试题 12、13	二进制数（binary numerals）、计算机文化（computer literacy）、
1992年试题 12、13	RISC 与 CISC、AI（人工职能）
1993年试题 12、13	移动式计算机（Mobile computers）、C 语言编码风格
1994年试题 11、12	面向对象的数据库管理系统（Object-oriented DBMSs）、Windows NT
1995年试题 11、12	微内核技术（microkernel technology）、窗口程序开发（Windows development）
1996年试题 12、13	局域网技术（LAN technology）、Java 语言
1997年试题 12、13	关系数据库模型（relational database model）、多媒体（multimedia）
1998年试题 12、13	拼写检查（spell checker）、局域网（LAN）
1999年试题 14、15	信息产业（information sector）、网络管理工具（network management tools）
2000年试题 14、15	VOIP、防火墙（firewall）

试题 1（2000 年试题 14）

从供选择的答案中，选出应填入下面叙述中{ }内的最确切的解答，把相应编号写在答卷的对应栏内。

Network managers have long { A } practical voice-over-IP(VOIP) solutions. VOIP{ B } ease network management and decreases costs by converging at a company's telephony and data infrastructures into one network. And a VOIP solution implemented at a company's headquarters with far-reaching branch offices can { C }tremendous amounts of { D } in long distance phone bill, provided that solution delivers POTS-like voice { E } over the Internet.

供选择的答案

A：①await ②awaited ③awaiting ④awaits

B：①promise ②promised ③promises ④promising

C：①get ②put ③save ④waste

D：①cash ②money ③space ④time

E：①frequency ②length ③quality ④quantity

【解析】

题中 A 处考查语法，这里是现在完成时，动词形式是过去分词，应选择"awaited"。

B 处也是个语法问题，这里是一般现在时，主语是单数形式，因而谓语动词的形式也是单数，应选择"promises"。

C 处从语义上看是"节省"的意思，只有"save"一次符合要求。

D 处指费用、金钱，只有"money"一词适合。

E 处指语音质量，应选择"quality"。

参考译文：网络管理员一直期待着实用的 VOIP(IP 话音业务)解决方案。通过将公司的电话和数据基础设施集成于同一网络，VOIP 简化网络管理，降低成本。如果 VOIP 解决方案通过因特网提供类似 POTS 的声音质量，将其用于拥有无程分支机构的公司总部，可节省大量的长途话费。

【答案】A：② B：③ C：③ D：② E：③

试题 2（2000 年试题 15）

从供选择的答案中，选出应填入下面叙述中{ }内的最确切的解答，把相应编号写在答卷的对应栏内。

Basically, a firewall is a { A } process or a set of integrated processes that runs on a router or { B } to control the flow of networked application traffic { C } through it. Typically, firewalls are placed on the entry point to a { D }network such as the Internet. They could be considered traffic cops. The firewall's { E }is to ensure to ensure that all communication between an organization's network and the Internet conform to the organization's security policies.

供选择的答案

A：①dependent ②isolated ③related ④standalone

B：①browser ②client ③repeater ④server

C：①pass ②passed ③passing ④passes

D：①local ②national ③private ④public

E：①mole ②pole ③role ④rule

【解析】

题中 A 处显然是"独立的、单独的"的意思，应选择"standalone"。

从上下文看，B 处是指服务器，应选择"server"。

C 处则是个语法问题，现在分词作定语，应选择③。

从后面的"Internet"判断，D 处意义应当是"公共的"，只有"public"一词符合。

E 处意是"角色"的意思，选择"role"。

参考译文：基本上，防火墙是在路由器或服务器上运行的一个独立的或一组集成的处理器，用来控制通过网络应用程序流经它的信息流。典型的防火墙位于进入像因特网这样的公共网络的入口处。可以将它们看作是交通警察。防火墙的角色就是确保组织的网络与因特网之间的所有通信符合组织的安全政策。

【答案】A：④ B：④ C：③ D：④ E：③

试题 3（1999 年试题 14）

从供选择的答案中，选出应填入下面叙述中{ }内的最确切的解答，把相应编号写在答卷的对应栏内。

All of these applications will enhance the { A } of life and spur economic growth. Over half of the U.S. work force is now in jobs that are information { B } . The telecommunication and information sector of the U.S. economy now { C }for 12 percent of the Gross Domestic Product, groing much faster than any other sector of the economy. Last year the { D }in this sector exceeded 700 billion dollars. The U.S. exported over 48 billion doooars of telecommunication equipment { E }.

A：①quantity ②quality ③mass ④amount

B：①based ②bassed ③bases ④base

C：①look ②looks ③account ④accounts

D：①revenues ②expenses ③outputs ④loss
E：①among ②alone ③simple ④single
【解析】
题中 A 处显然指生活质量，应选择②。
B 处则是个语法问题，是被动语态。
C 处从语义上看是"占多少比例"，"account for"是习惯用法，注意一下语法，谓语动词是单数。
D 处指财政收入，只有"revenues"一词符合。
E 处是"单独"、"独自"的意思，语法上是副词作状态，选"alone"。
参考译文：所有这些应用软件会提高人们的生活质量，刺激经济增长。现在美国半数以上的劳动力所从事的工作依赖于信息。在美国经济中，电信和信息门类占国内生产总值的 12%，而且增长速度也比其他经济门类都快。去年该行业收入超过 7000 亿美元。美国电信设备的出口额超过 480 亿元。
【答案】A：② B：① C：④ D：① E：②

试题 4（1999 年试题 15）

从供选择的答案中，选出应填入下面叙述中{ }内的最确切的解答，把相应编号写在答卷的对应栏内。
The growth of switching has { A } a new generation of network management tools that help { B }cope with the challenges, { C }them to correlate device alarm in order to { D }pinpoint root causes, or to monitor service levels without depending on IP subnetting schemes. These new tools have become essential to coping with the primary { E }effect of any treatment that significantly increases network flexibility: added complexity.
供选择的答案
A：①create ②creates ③created ④creating
B：①technicians ②workers ③salesmen ④professors
C：①enjoying ②enabling ③engaging ④enriching
D：①better ②well ③welly ④goods
E：①back ②side ③front ④lateral
【解析】
本题是关于网络管理技术的，难度大一些。
A 处主要考语法，完成时动词形式。
B 处考词义，只有"technicians"(技术员)最合适。
C 处从语义和语法上看应是"enabling"，意思是"使……能"。
D 处副词修饰动词不定式，而且隐含比较级，只有"better"合适。
E 处是"副作用"的意思，显然用"side"。side 作形容词，表示"旁边的、侧面的、副的、枝的"，side effect 则指副作用。
参考译文：交换技术的发展产生了新一代网络管理工具。这些工具有助于技术人员应付挑战，使它们能与设备报警相联以更好地查明根本原因，或者不依赖于 IP 子网配置来监控服务级别。这些工具对解决因显著增加网络适应性(额外的复杂性)带来的主要副作用的问题是必需的。
【答案】A：③ B：① C：② D：① E：②

试题 5（1998 年试题 12）

从供选择的答案中，选出应填入下面叙述中{ }内的最确切的解答，把相应编号写在答卷的对应栏内。

Many word processing programs include spell checker. It checks the spelling of every word in a { A } by looking up each word in its dictionary. If the word does not appear in the dictionary, the user is { B }to a possible misspelling and possible corrections are often { C }. Spell checker does not recognize unusual people names or specialized terms, but it will often allow you to to create your own personal dictionary of specialized words you often use. Spell checker is a valuable aids to proofreading, but it can not catch the { D }of one correctly spelled word for another (such as form for from). Thus it does not { E }a document is free of spelling errors.

A;①document ②equipment ③program ④statement

B、C ①alerted ②alternate ③guessed ④guided ⑤suggested ⑥surprised

D; ①addition ②condition ③notation ④substitution

E; ①committee ②correct ③grarantee ④prove

【解析】

参考译文：许多字处理都包括了拼写检查的功能，拼写检查通过查自己的字典来检查一个文件中的每一个单词。如果字典里没有这个单词，用户就会被警告可能拼写错误，同时也经常提供一个纠正该错误的建议。拼写检查程序不能识别不常用的人名或专用术语，但是它允许你生成自己的个人字典，把自己常用的词语加进去。拼写检查程序对文字校对来说是一个有用的工具，但是它不能检查到一个单词被另一个拼写正确的单词所代替这样的错误，因此它不能保证一个文件中没有拼写错误。

【答案】A：① B：① C：⑤ D：④ E：③

试题 6（1998 年试题 13）

从供选择的答案中，选出应填入下面叙述中{ }内的最确切的解答，把相应编号写在答卷的对应栏内。

A local-area network(LAN) is a communications network that { A } a variety of devices and provides a { B }for information exchange among those devices. The scope of the LAN is small typically a single building or a cluster of builings. The LAN is usually owned by the same organlization that owns the { C } devices.The internal data rate of LAN is at least several Mbit/s. The basic of LAN communciation is broadcasting. At each station, there is a transmitter/receiver that commuinciates over a { D }shared by other stations. A transmission from any one station is { E }to an received by all other stations.

供选择的答案

A、B、D、E：①broadcast ②cable ③control ④interconnects

⑤internet ⑥meander ⑦means ⑧medium

⑨modem ⑩output 11. switch 12. relay

C：①adopted ②attached ③selected ④unified

【解析】

参考译文：局域网是将许多设备互连在一起并提供一种在这些设备间交换信息的方法的通信网。局域网的范围通常比较小，如一个建筑物或一群建筑物。局域网通常为拥有那些连接在网上的设备的同一组织所拥有。局域网的内部传输率至少为几兆字节每秒。局域网的基本通讯方式为广播方式。每个站都有一个转发/接收器，通过一个与其他站共享的媒介来通讯。从任何一个站发送的消息都被广播发送到所有的站，被所有的站接收。

【答案】A：④ B：⑦ C：② D：⑧ E：①

试题 7（1997 年试题 11）

从供选择的答案中，选出应填入下面叙述中{ }内的最确切的解答，把相应编号写在答卷的对应栏内。

The relational database model requires the data be { A } through programs that don't rely on the position of the data in the database. This in in direct { B } to the other database mode, where the program has to follow a series of pointers to the data it seeks.{ A } program { C } a relational database simply asks for the data it seeks; the DBMS performs the necessary searches and { D }the information. The { E }on how the search is done are specific to the DBMS and vary from product to product.

供选择的答案

A：①accessed ②moved ③read ④wrote

B：①conduct ②contract ③contrast ④construct

C：①consulting ②containing ③querying ④queuing

D：①erases ②provides ③proves ④values

E：①details ②documents ③tails ④tenants

【解析】

参考译文：在关系型数据库模型中，要求通过程序访问数据时，不必依赖于数据在数据库中的位置。这种方式与其他数据库模式中采用的方式的不同之处在于：其他数据库模式中，程序需要通过访问一系列指针才能查到所要访问的数据。在关系型数据库中，程序查询数据库的方式只是直接指向所要查找的数据，数据库管理系统会实施必要的查询工作，然后将查询结果提供给程序。具体如何进行查询，与数据库管理系统有关，并且不同的数据库产品之间也有所不同。

【答案】A：① B：③ C：③ D：② E：①

试题 8（1997 年试题 12）

从供选择的答案中，选出应填入下面叙述中{ }内的最确切的解答，把相应编号写在答卷的对应栏内。

When most people refer to multimedia, they generally mean the combination of tow

or more continuous media, usually with some user { A } . In practice, the two media are normally audio and video, that is, { B }plus moving { C }.

It should be obvious by now that transmitting multimedia material in uncompressed form is completely out of { D }, the only hope is that massive compression is possible. Fortunately, a algorithms that make multimedia transmisson { E }.

供选择的答案

A～C：①display ②games ③help ④interaction

⑤pictures ⑥sound ⑦web

D、E：①impossible ②fearful ③feasible ④program

⑤questio ⑥thing

【答案】A：④ B：⑥ C：⑤ D：⑤ E：③

试题 9（1996 年试题 13）

从供选择的答案中，选出应填入下面叙述 {} 的最确切的解答，把相应编号写在答卷的对应栏内。

{ A } whether to go with a shared or switched fast LAN technology once you start to maxout your

shared Ethernet of token-ring LAN is tough call. Users and analysts involved in { B }high-speed LANs have learned the benefits of each. However there are trade-offs to each, and { C } which fast LAN technology to choose is critical for { D }network performance problems and { E }costly purchasing mistakes.

供选择的答案

A～E：①avoiding ②clipping ③choosing ④deciding ⑤dissolving
⑥ensuring ⑦evolving ⑧implementing ⑨knowing ⑩solving

【解析】

参考译文：你一旦想扩大你自己的以太网或令牌环 LAN 的共享能力，就面临着一种选择：是用共享快速 LAN 技术还是交换快速 LAN 技术?那些已决定高速 LAN 的用户们和分析家们都知道，这两种技术各有其优点也都有其不足。明确选择哪一种技术是解决网络性能问题和避免购买浪费的关键。

【答案】A：④ B：⑧ C：⑨ D：⑩ E：①

试题 10（1996 年试题 14）

从供选择的答案中，选出应填入下面叙述中{ }内的最解切的解答，把相应编号写在答卷的对应栏内。

The most accurate and most boring way to { A } Java is that is a new computer programming language developed by Sun Microsystems that creates { B }independent programs the that can be distributed and run remotely. To run Java programs, a computer must have a Java interpreter. Currently { C }Java programs are small"applets" that are { C }as part of web pages.

When you use a Java { D }browser to view a Web page that in cludes a Java applet, the broser loads the applet onto your computer through your modem or network. Then the Java interpreter runs the applet, which could include animation or sound, on your computer rather than transmitting the code bit by bit over the Internet. A few thousand bytes of Java code can turn into a powerful program on your computer.

So an applet could in clude { E }with Java interpreter.

供选择的答案

A：①command ②comment ③describe ④discover
B：①condition ②platform ③programmer ④workstation
C、D：①available ②capable ③possible ④probable
⑤stable ⑥valuable
E：①animation or sound ②animation and sound
③animation and be run ④animation and be loaded

【解析】

参考译文：关于 Java 一词最准确且最乏味的描述是，它是一种新的计算机程序设计语言，是由 Sun Microsystems 公司开发的，它用来生成与平台无关的可以分布在远程机器上运行的程序。要想运行 Java 程序，计算机上必须装有 Java 解释器。当前可用的 Java 程序是一些小(应用程序)片断，被用作 Web 页的一部分。

当你用一个支持 Java 的浏览器来浏览一个包含 Java 小应用程序的 Web 页时，浏览器将这个小应用程序通过调制器或网络传到你的计算机上，然后 Java 解释器运行这个小应用程序，其中可能镶嵌动画和声音，这些操作都是在你的计算机上完成的，而不是通过 Internet 一位一位地传输代码过来。几千字节的 Java 代码可能在你的计算机上变成一个强有力的程序。

所以一个 Java 小应用程序可以包含动画，需要 Java 解释器来执行它。
【答案】A：③ B：② C：① D：② E：③

试题 11（1995 年试题 11）

从供选择的答案中，选出应填入下面英文语句叙述中{ }内的正确答案，把编号写在答卷的对应栏内。

For users, microkernel technology promise { A }, compact, and sophisticated operating systems that are typically { B }across a range of hardware platforms. These operating systems will be customizable to let users run multiple operatin system and application "per-Sonalities" on top of a single microkernel { C } .Microkernel-based systems can provide this flexbility because the core operating

system functionc are { D } from the large number of utilities, features and extension that are layered on top of them. As a result, updation and maintaining operating system is easier, since developers don't have to modify the { E } every time they must add a new feature.

供选择的答案：

A：①slow ②fast ③quickly ④speed

B：①moveable ②made ③portable ④use

C：①bases ②foundations ③systems ④foundation

D：①separate ②connected ③compared ④selected

E：①core ②center ③heat ④utility

【解析】

参考译文： 对用户来说，微内核技术有希望实现能在许多硬件平台上移植的快速、紧凑和先进的操作系统。这些操作系统可以定制化，以使用户可在单个微内核基础上运行多个操作系统和应用程序。

基于微内核的系统之所以能够提供此灵活性是因为核心的操作系统功能与大量用程序、特征和扩展功能是分离的。它们分层加于核心功能上，其结果是使更新和维护操作系统更加容易了。因为开发人员不必在每次加入一个新特色时都修改核心。

【答案】A：② B：③ C：④ D：① E：①

试题 12（1995 年试题 12）

从供选择的答案中，选出应填入下面英语文句叙述中{ }内的正确答案，把编号写在答卷的对应栏内。

Application development increasingly means Windows development, and the popularity of visual development tools has { A } in tandem with Windows itself. These tools create beautiful windowing { B } , and their fast development cycles and easy learning curves make them a good { C } for many types of PC development projects. Today's developers are leveraging these tools and the abundance of cheap, powerful PCs to shift the balance of power to the desking.

As the world moves inexorably toward Windows and other { D }user interfaces, developers can choose from an abundance of { E } oriented tools. Popular examples in clude Microsoft Corp's Visual Basic, Powersoft Corp's PowerBuilder, Gupta Technology Corp's SQL Windows, and so on.

供选择的答案：

A：①rise ②risen ③rised ④rising

B：①interfaces ②pictures ③graphs ④books

C：①choose ②selections ③choice ④select
D：①graphics ②graph ③graphitic ④graphical
E：①visually ②lively ③quickly ④specially
【解析】
参考译文：应用程序的开发越来越多地意味着 Windows 的开发。直观化的开发工具的流行紧随着 Windows 本身的普及而发展。这些工具能生成漂亮的窗口式接口，其快速开发周期和易于学习的过程使它们成为多种 PC 开发项目的良好选择。今天的开发人员正利用这些工具和大量廉价而功能强有和 PC 机把力量的平衡移到台式机上。
随着世界坚定地朝着 Windows 和其他图形用户接口转移，开发人员可以在大量的面向直观化的工具中进行选择。流行的例子包括 Microsoft 公司的 Visual Basic、Powersoft 公司的 Power Builder, Gupta 技术公司的 SQL Windows 等。
【答案】A：② B：① C：③ D：④ E：①

试题 13（1994 年试题 11）

从供选择的答案中，选出应填入下面英语文句叙述中{ }内的正确答案，把编号写在答案的对应栏内。
Object-oriented DBMSs integrated a variety of { A } data types-such as business procedures, graphics, pictures, vioce and annotated text.
Object orientation also makes a { B }to application development efficiency. It makes the data, functions, attributes, and relationships an integral part of the { C } .
In this way, objects can be reused an replicated.
Some leading RDBMS vendors support the concept ofintegrating object management capabilities with their current line of relational products. That capability enable users to { D }the development cycle, since integrity logic and business rules no longer need to be programmed { E } each application.
供选择的答案
A～E：①tool ②in ③idea ④contribution
⑤joke ⑥short ⑦object ⑧theoretical
⑨extent ⑩shorten 11.real-world 12.into
【解析】
参考译文：面向对象的数据库管理系统将各种现实世界的数据类型--商业事务过程、图形、图片、语音和注释文本等加以集成。
面向对象对应用开发效率也有贡献。它使数据、函数、属性和关系成为对象的完整的部分，这样，对象就可以被重新使用和复制。
某些领先的关系数据库管理系统的销售商支持将对象管理能力与他们目前经营的关系产品系列相结合的概念。这种能力可以使用户缩短开发周期，因为完整的逻辑和事务规则不再需要为每个应用程序编程。
【答案】A：11 B：④ C：⑦ D：⑩ E：12

试题 14（1994 年 12）

从供选择的答案中，选出应填入下面英语文句叙述中{ }内的正确答案，把编号写在答卷的对应栏内。
In { A }software, one question of near-universal interest is how rapidly Windows NT, which began shipping last summer, will be { B } in the marketplace and for what uses. For the most part,

observers are { C } about the operating system's long term prospects, but expect it to remain on { D } rather than desktops during 1994, saying that it won't become a mainstream product until { E } PCs typically have 16 to 24MB of RAM, which is unlikely to happen next year.

供选择的答案

【答案】A～E：①applications ②accepted ③clients ④enthusiastic
⑤Servers ⑥desktop ⑦systems ⑧replaced

【解析】

参考译文：在系统软件中，一个普遍关心的问题是去年夏天开始出售的 Windows NT 将会如何快速地被市场所接受以及起什么作用。大部分观察家们对该操作系统的长远前景表示出热情，但预料在 1994 年间它仍将用在服务器上而非台式机上。他们说，在典型的台式 PC 机具有 16MB 至 24MB 内存之前，它不会成为主流产品，而这件事明年是不大可能发生的。

【答案】A：⑦ B：② C：④ D：⑤ E：⑥

试题 15（1993 年试题 12）

从供选择的答案中，选出应填入下面英语文句中{ }内的正确答案，把编号写在答卷的对应栏内。

Mobile computers- which { A } laptops, notebooks, subnotebooks and handhelds - { B } so ubiquitous in such a short time, no sunrise to hear who say:"It will define the leading edge { C } the next five years or so." The most remarkable { D }mobile computers is the amount of data storage and memory packed { E }their tiny boxes. These devices not only handle windows easily but also run storage-hungry programs.

供选择的答案

A、B：①included ②include ③have included ④have become
⑤had become

C～E：①from ②for ③about ④at
⑤on ⑥into

【解析】

参考译文：移动式计算机包括膝上型、笔记本型、小笔记本型和掌上型，在如此短的时间内变得如此之普遍，不必吃惊它将在下一个 5 年或更长时期名列前茅。移动式计算机最显著的特点是装入其小盒子的数据存储的量。这些设备不仅能容易地处理窗口作业而且能运行大存储量的程序。

【答案】A：② B：④ C：② D：③ E：⑥

试题 16（1993 年试题 13）

从供选择的答案中，选出应填入下面英语语文句中{ }内的正确答案，把编号写在答卷的对应栏内。

One of the guidelines in writing the C code for a software tool is as follows: write code that is as clear and as simple as { A }. The C language can be difficult to read if you combine all { B }features in a single statement. Break complicated { C }into several easy to understand statements for the { D }of readability. This style helps to make your programs more { E } and error-free.

供选择的答案

A～E：①readable ②reusable ③Possible ④semantic
⑤syntactic ⑥constructions ⑦safe ⑧impossible

⑨sake ⑩structure

【解析】

参考译文：为软件工具写 C 程序源码的一个规则如下：写的程序尽可能清晰、简单。如果你把所有句法特征组合在单个语句中，C 语言将会很难读。为了提高可读性应打破复杂结构，将其分为几个易于理解的语句。这种风格有助于使你的程序更加可读，而且不出错误。

【答案】A：③ B：⑤ C：⑥ D：⑨ E：①

试题 17（1992 年试题 12）

从供选择的答案中，选出应填入下列英语文句中{ }内的正确答案，把编号写在答卷的对应栏内。

In recent years, one of the more poplar topics panel discussions at computer conferences and trade { A } has been the"RISC versus CISC"debate.

RISC processors feature a small number of instructions that each executes in { B } machine cycle. CISC processors use complex instructions that can take several cycles to execute.

The RISC Versus CISC debate won't be decided by panel discussion 1 it will be won in the marketplace. And the deciding factor may have little to do with { C } of instructions and registers) and more to do with parallelism.

Since their conception, RISC processors have been evolving toward microparallelism, in corporating parallel-processing features { D } the processor. RISC processors feature pipelining, whereby many instructions can be decoded while one instruction executes.

RISC processor, however, are moving to toward pipelines for each unit of the processor.

CISC processors also employ pipelining. They have many interger instructions that execute in one cycle, but the varying execution times of CISC instructions { E } the effectiveness of parallelis.

供选择的答案

A～E：①union ②two ③numbers ④between ⑤limit

⑥contents ⑦shows ⑧one ⑨within ⑩enhance

【解析】

参考译文：近年来，在计算机学术会议和贸易展览会上最流行的会场讨论课题之一是"RISC 对 CISC"的争论。

RISC 处理器以少量指令为特征，且每条指令在一个机器周期中执行。CISC 处理器使用复杂的指令，它们可能花费若干周期来执行。

RISC 对 CISC 的争议将不由会场讨论来作结论，它将在市场中定胜负。并且，决定的因素也许很少取决于指令和寄存器的数目，而更多取决于并行性。

自从 RISC 概念提出后，RISC 处理器正向着微并行性发展，它在处理器内部结合了并行处理的特性。RISC 处理器以流水线为特点，在流水线上当一条指令执行时，许多指令能被解码。况且，RISC 处理器正向着每个处理器单元多流水线发展。

CISC 处理器也使用流水线。它们有许多在一个周期中执行的整数指令，但是 CISC 指令变化的执行时间限制了并行性的效率。

【答案】A：⑦ B：⑧ C：③ D：⑨ E：⑤

试题 18（1992 年试题 13）

从供选择的答案，选出应填入下列英语文句中{ }内的正确答案，把编号写在答卷的对应栏内。

While most recemt attemtion in the AI field has been focused on expert system software, AI { A }

has also seen dramatic advances. Activity, In the past years was characterized by new low-cost, powerful LISP machines, the introduction of AI workstations, LISP compilers becoming available for all major professional and engineering workstations, and the personal computer emerging as a { B }tool for expert system development. The next few years will see this technology evolves further.

Because the{ C } of an AI computer represents a sizeable investment, companies should carefully { D } all options that are available as well as have a good idea of what the next generation of systems will offer in order to { E }the optimum system. This publication provides the Information necessary to gain this understanding.

供选择的答案

A～E：①choice ②read ③importan ④software
⑤hardware ⑥significant ⑦emergence ⑧survey
⑨purchase ⑩select

【解析】

参考译文：当 AI 领域最近的注意力集中于专家系统软件时，AI 硬件也显示出巨大的进展。过去一些年的活动有下述特征：新的低价格和强功能的 LISP 机、引进 AI 工作站、LISP 编译器可用于所有主要的专业或工程工作站以及个人计算机作为专家系统开发的值得注意的工具出现。最近几年将会看到这个技术的进一步演变。

因为购买一个 AI 计算机意味着一笔相当在的投资，公司为了选择一个最佳系统应仔细地考查所有可用的选择，并对下一代系统将提供什么有一个好的想法。本文对了解这些东西提供了必要的信息。

【答案】A：⑤ B：⑥ C：⑨ D：⑧ E：⑩

试题 19（1991 年试题 12）

从供选择的答案中选出应填入下列英语文句中{ }内的正确答案，把编号写在答卷的对应栏内。

It is traditional when dealing with languages of all sorts to try to separate co
nerns with{ A } , the subject of syntax, from concerns with B , the field of semantics, Consider the simple "language" of binary numerals. Some examples of binary mumerals are

0　　1
101　　0101
11001　　101111

A communication in this language evidently consists of a finite sequence of characters '0', '1' This is just syntax however, and says C about what such a communication is intended to mean.

Mumbers are "D" mathematical concepts, whereas the digit strings that strings that appear on paper are numerals, that is to say, E representations or descriptions of numbers.

供选择的答案

A～C：①symbol ②form ③meaning ④context
⑤nothing ⑥concept

D、E：①abstract ②concrete ③simple ④ordinary
⑤symbolic ⑥logic

【解析】

参考译文：处理各种语言时，传统的做法试图将有关形式的语法问题和有关含义的语义方面区分开来。考虑简单的二进制数的"语言"。下面是二进制数的例子

【答案】 A：② B：③ C：⑤ D：① E：⑤

试题 20（1991 年试题 13）

从供选择的答案中选出应填入下列英语文句中{ }内的正确答案，把编号写在答卷的对应栏内。

With the widespread use of the personal computer, many authorities in the field of { A } have pointed out the need for computer literacy.

Unfortunately, there is no { B } agreement as to what the term "computer literacy"

means. Some feel that computer literacy means knowing how to make the computer "computer literacy" means. Some feel that computer literacy means knowing how to make the computer "compute"; that is, knowing how to program computers in one or more programming languages.

Others feel that knowing how to program is merely a small segment of computer literacy. These people { C } the major emphasis in schools should be on teaching how to effectively use the many software packages that are available.

Still others suggest that computer literacy education is not required. They suggest tha computers are being so rapidly integrated into our society that using a computer will be as { D }as using a telephone or a video tape recorder, and that special education will not be necessary.

{ E } of one's definition of computer literacy, it is recognized by most that learning to use a computer is indeed an important skill in modern society.

供选择的答案

A：①culture ②science ③education ④industry

B、D：①equal ②universal ③different ④difficult

⑤common ⑥big

C：①claim ②deny ③define ④call

D：①Importance ②Instead ③Because ④Regardless

【解析】

参考译文：随着个人计算机的广泛使用，许多教育领域的权威人士已经指出了对计算机文化的需要。

不幸的是对于"计算机文化"术语的含义尚无一致的意见。某些人觉得计算机文化意味着知道如何用计算机来"计算"；也就是说，知道如何以一种或多种编程语言为计算机编程。

另一些人认为如何编程只是计算机文化的一小部分。这些人主张学校中的教学重点应是教会学生如何有效地使用大量已有的软件包。

另外一些人建议计算机文化教育是不需要的。他们建议计算机正如此快速地进入我们社会中，以致于使用计算机将像使用电话或录像机一样普通，特设的教育是不必要的。

抛开人们关于计算机文化的定义，大多数人承认学会使用计算机的确是现代社会中的一项重要技能。

试题 21（1990 年试题 11）

将下列英语译成中文(只可使用词典)：

Programming languages

Ten years ago the proliferation of programming languages caused many people to foresee the development of a computer-age babel where, in total ignorance of every other language, each programmer would learn only his own chosen language. That unhappy situaton has not occurred

for several reasons. First, effective efforts have been made to standardize particular languages such as Fortran and Cobol. It should be pointed out that pragmatic rather than scientific considerations motivated this standardization movement. However, the second reason that Babel has been averted is that computer scientists have begun to apply the scientific method to organize the classification, comparison, and appreciation of various programming languages.

Due to the efforts of McCarthy(1962), Landin(1964), Strachey(1996), Wegner(1968), and others who provided insight into operational models of computation. We can now evaluate programming languages in terms of an unifying view of computation structures. Semantics and the expressive power resulting from modularity can now be studied in terms of the data structures and the accessing paths to them established during the execution of the control statements of the language.

【答案】

参考译文：程序设计语言

十年前程序设计语言的激增使得许多预言在 Babel 时代，计算机空想通天塔的发展情形(Babel 是基督教《圣经》中的城市名，诺亚的后代拟在此建通天塔，上帝怒其狂妄，使建塔人突操不同语言，塔因此终未建成。这里，用来比喻使用多种程序设计语言的空想)，在那里每个程序员完全可以忽略其他任何的语言，仅学习他自己选择的语言。由于种种原因，这种不幸的状况未发生。首先，有效的努力对特定的语言如 Fortran 和 Cobol 等的标准化起了作用。应指出，实用的而不是科学的考虑推动了这个标准化的进程。然而，Babel 空想得以避免的第二个原因是计算机科学家已开始应用科学的方法来组织各种程序设计语言的分类、比较和鉴别。

通过 McCarthy(1962)、Landin(1964)、Strachey(1996)、Wegner(1968)和其他对计算的操作模型深入考查的人们的努力，现在我们正以统一的计算结构观点来评价程序设计语言，可以通过数据结构以及在执行语言语句时建立的访问路径，来研究语言和模块化导出的表达能力。

试题 22 (1990 年试题 12)

将下列英语译成中文(只可使用词典)：

Integrated Software

Convenience and saved time, work, and effort are the promises of integrated software.

The antithesis of stand-alone packages, integrated software delivers a collection of applications based upon a common user interface and sharable data.

In its most common form, the integraged product includes a word processor, a spreadsheet, and some form of database. Many packages add telecommunications, presentaion graphics, and outline modules. Comprehensive products throw in desktop accessories such as calculators, calendars, DOS shells, and other utilities.

Even when stand-alone products are from the same vendor, it can be frustrating trying to move information between applications or simply trying to remember which key to press to call up the menu. That is why integrated packages appeal to many users, particularly novices. Using an integrated product saves you the headache of trying to move data in a Brand X word processor to a Brand Y spreadsheet. And because the integrated package is a single product from a single vendor, training, support, and upgrades also are made simpler.

【答案】

参考译文：集成软件

方便、省时、省工县有效是集成软件的应具备的性质。

同独立软件包相比，集成软件提供一组基于公共用户界面和共享数据的应用程序。
在其大多数常用的形式中，集成的产品包括字处理、电子报表和某种形式的数据库。许多软件包加进了远程通信、表示图形以及轮廓模块。复杂的产品增添了如计算机器、日历等办公辅助软件、DOS 外壳及其他实用程序。
即使独立的产品出自同样的供应商，试图在应用程序间传递信息或者只是简单地记住按哪个键来调出菜单都可能行不通。这就是集成软件包为什么对许多用户特别是新用户有吸引力的原因。使用集成的产品能省去试图将牌号 X 字处理程序中的数据传到牌号 Y 的报表程序这类令人头痛的问题。同时，因为集成软件包是来自单一供应商的单一产品，培训、支持和升级也变得更简单。

五 软件设计试题精解

根据考试大纲的要求，高级程序员不仅要具备高水平的程序编制能力，而且要熟练掌握软件设计的方法和技术，具备一定的软件设计能力。软件设计能力由下午试题考查，目前是 3 道题中选答 2 道。试题的形式一般是根据有关的软件分析设计图(常见的有数据流图、程序流程图、系统流程图和 ER 图)和其他说明资料，考生按要求回答问题或填空。试题内容涉及到流程图设计、1 软件界面设计、数据库设计和软件测试等多个方面，要求考生熟练地掌握软件分析和软件设计的常用方法和技术。

5.1 2000 年度软件设计试题解析

2000 年度下午试卷共有 3 道软件设计试题，都涉及到流程图，有两个突出的特点：

第一，试题 1 的问题 2 要求写出操作的内容，试题 2 的问题 2 和问题 3 要求写出子程序的功能和操作内容。这两个问题的形式在历年试题中比较少见，但这也正反映了对高级程序员软件分析能力的要求是一种科学的考查方式。

第二，在试题 3 的问题 1 中，要求找出哪张图中的哪些文件不必画出。这种形式的问题在历年试题中也比较少见，而且对考生的要求也比较高，要求考生仔细研究每一张图。在这种问题中，解题的指导原则与解答其他数据流图试题相同。只要把握好指导原则，问题的解答还是有规律可循的。

除了上述两个方面，其他关于数据项的定义、判断条件的填写以及流程图处理具体个例的考查方式在历年试题中比较多见，就不再分析了。

试题 1 (2000 年试题 1)

阅读以下说明和流程图，如图 6-1 所示，回答问题 1 和问题 2，将解答写在答卷的对应栏内。

［说明］

本流程图实现由成绩文件生成学生成绩一览表。

某中学某年级的学生成绩数据(分数)登录在成绩文件 F0 中，其记录格式如下：

学号	姓名	课题 1 成绩	课程 2 成绩	……	课程 6 成绩

由该成绩文件生成如下表所示的学生成绩一览表，并按学号升序排列。表中的名次是指该生相应课程在年级中的名次。

学号	姓名	课程 1		课程 2		……		课程 6	
		成绩	名次	成绩	名次	……	……	成绩	名次

在如图 6-1 所示的流程图中，顺序文件 F0 是学生成绩文件。F0 文件经处理 1 处理后产生顺序文件 F，然后经过处理 2 至处理 4 对文件 F 进行处理和更新。在处理 5 中，仅对文件 F 的记录进行编排，输出学生成绩一览表，但不进行排序和增加名次等处理。

图 6-1　2000 年试题 1 流程图

[问题 1]

流程图中文件 F 的记录格式设定为如下形式：

学号	姓名	课程代号	①	②

其中的①和②应定义的为何种数据项?

［问题 2］

简述处理 2、处理 3 和处理 4 作何种处理，若有排序处理则需指明排序的键和序(升序或降序)。

【解析】

本题有一定的难度。在给出了文件 F0 和最后输出的学生成绩一览表后，整个流程图的主体部分(处理 2、处理 3 和处理 4)的内容都需要根据试题说明和流程图的结构进行安排。对于此类试题，需要比较系统的输入数据和输出数据，以确定从输入数据转换成符合输出要求的数据需要经过何种处理，然后再合理安排这些处理的顺序。

系统输的成绩文件 F0 是一个顺序文件，在学号、姓名之后顺序存放了 6 门课程的成绩。文件 F0 经过处理 1 处理后，产生顺序文件 F，然后经过处理 2、处理 3 和处理 4 的处理和更新。在处理 5 中，对文件 F 的记录进行学生成绩一览表的编排输出，不进行排序和增加名次的处理。关于文件 F，试题说明中给出了部发记录格式，除学号、姓名和课程代码外，还有两个未定义的数据项，需要根据实际情况给出。

研究学生成绩一览表的格式，我们可以看到，学号按升序排列，课程也有一个排列次序，而且增加了某学生某门课题成绩的名次。这就要求对每门课程的成绩进行排序，然后按照排序的结果给出名次。这样，在处理 4 处理文件 F 的过程中，需要按学号排序，同时还要按课程代码排序。即使这不是处理 4 的全部内容，也是处理 4 的一部分内容。

如果需要从文件 F 编排输出学生成绩一览表，会发现在文件 F 的记录格式中还有两个未定义的数据项，比较文件 F 的记录格式和学生成绩一览表，发现学生成线一览表中的成绩和名次两个数据项在文件 F 的记录格式和学生成绩一览表，发现学生成绩一览表中的成绩和

名次两个数据项在文件 F 的记录格式中没有反映，先假定这里的两个未定义数据项就是成绩和名次。
或许有的考生要问，如果在处理过程中不产生名次，在处理 5 中只是按排序的顺序来输出是否可以呢?也就是说，在处理 5 中，文件 F 是按课程排序的，同时每门课程又按成绩排序。这是不可能的，由于每个学生各门课程成绩的名次各不相同，如果没有名次这个数据项，是无 法同时保证学号和成绩的有序性的。如果不能够同时保证学号和成绩的有序性，处理 5 编排输出效率将会大大降低。
基于以上的分析，必须明确名次这个数据项应该怎样加入。如果要产生名次，需要对成绩进行排序，这是常识。由于文件 F 的记录格式中提供了数据项名次，可以设想，在每门课程按成绩排序后，再按课程代码和成绩向文件 F 中写入名次是非常容易的。至此，从处理 1 后的文件 F 到满足上述需求的数据已经不需要额外的操作了。
结合以上的分析，试着在头脑中模拟一遍程序的运行过程，以确认上述分析的正确性。下面的工作是将以上的分析用准确的语言描述出来，并进行合理的划分，把它们安排到处理 2、处理 3 和处理 4 中去。
这里按照流程图的顺序来分析，而不是像上面那样逆流程图结构进行分析。在处理 1 后，需要对文件 F 进行排序，包括按课程代码排序(升序或降序排列都可以)和按成绩(升序或降序都可以)排序。排序后，需要在文件 F 中按照排序结果写入名次，然后，将文件 F 再按学号(必须是升序)和课程代码(升序和降序都可以)排序。
将以上所描述的 3 个部分内容分配到处理 2 至处理 4 中，再通读流程图，以确认在该题的解答过程中没有遗漏重要的信息内容。
【答案】
［问题 1］ ①成绩 ②名次
或①名次 ②成绩
［问题 2］
处理 2：以课程代码(升序)和成绩(降序或升序)为键对文件 F 排序。
处理 3：对每个课程代码，决定学生名次，并写入文件 F 的相应字段。
处理 4：以学号(升序)和课程代码(升序)为键对文件 F 排序。
注：在处理 2 和处理 4 中，课程代码的排序方式也可以同时为降序。

试题 2 (2000 年试题 2)

阅读以下说明和流程图，回答问题 1 至问题 4，将解答写在答卷的对应栏内。
【说明】
本流程力量(如图 6-2 所示)是将中缀表示的算术表达式转换成后缀表示，如中缀表达式
(A-(B*C+D)*E)/(F+G)
的后缀表示为：
ABC*D+E*-FG+/。

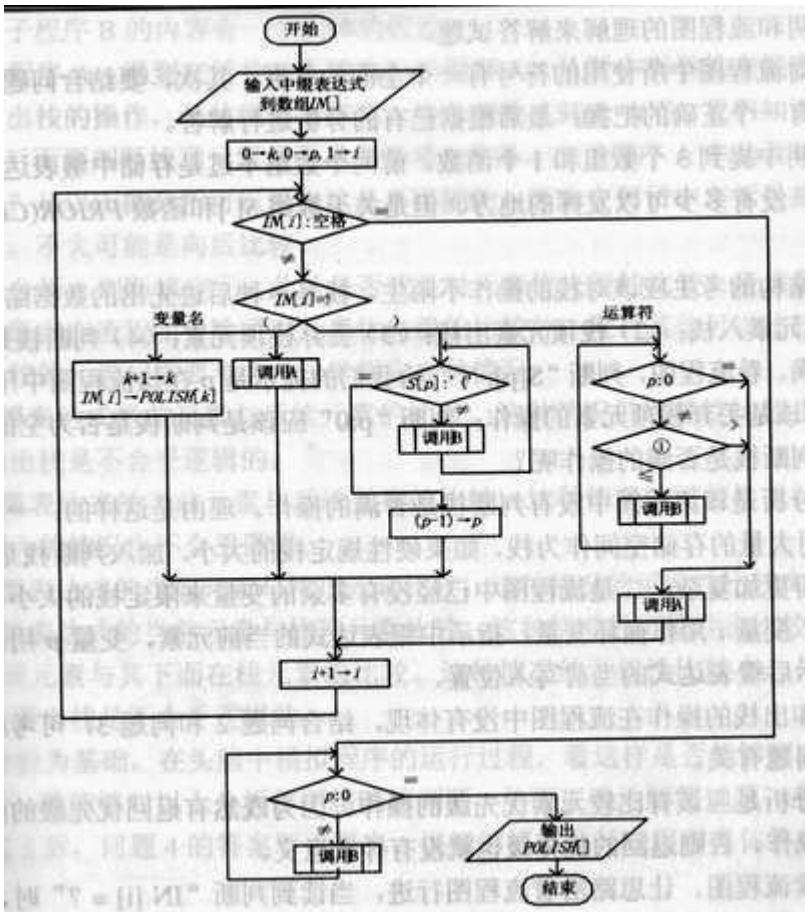


图 6-2　2000 年试题 2 流程图

为了方便，假定变量名为单个英文字母，运算符只有+、-、*和/(均为双目运算符，左结合)，并假定所提供的算术表达式非空且语法是正确的。另外，中缀表示形式中无空格符，但整个算术表达式以空格符结束。流程图中使用的符号的意义如下：

数组 IN［］存储中缀表达式；

数组 POLISH［］存储其后缀表示；

数组 S［］是一个后进先出栈；

函数 PRIOR(CHAR)返回符号 CHAR 的优先级，各符号的优先级如下表所示。

CHAR	PRIOR(CHAR)
*/	4
+-	3
(	2
)	1

［问题 1］

填充流程图中①的判断条件。

［问题 2］

写出子程序 A 的功能，并顺序写出实现该功能的操作。

［问题 3］

写出子程序 B 的功能，并顺序写出实现该功能的操作。

［问题 4］

中缀表达式

(A+B-C*D)*(E-F)/G)经该流程图处理后的输出是什么?

【解析】
本题是关于中缀表达式转换为后缀表达式的试题。如果考生的知识面比较宽，应该了解该种问题是有常用算法的。本题的流程图就是常用算法的一种描述，如果掌握了该算法，解答将是非常容易的。下面先用自然语言叙述该算法。
转换时，从中缀表达式的左端开始，向右逐个扫描处理。表达式包含 4 类元素：(1)变量名；(2)左括号'('；(3)，右括号')'；(4)运算符号。处理时需要用到后进先出的数据结构--栈。具体处理规则如下：
(1)如果当前元素为变量名，则直接将该元素写入后缀表达式。
(2)如果当前元素是左括号，则将该元素入栈。
(3)如果当前元素是右括号，则栈顶元素顺序出栈，写入后缀表达式，直到栈顶元素为左括号，就将该栈顶元素丢弃。
(4)如果当前元素是运算符号，则分两种情况：(a)当栈不空时，比较当前元素与栈顶元素的优先级(优先级从高到低分别为：*/、+-、(、))，当栈顶元素优先级当前元素的优先级时，当前元素入栈，否则栈顶元素顺序出栈，写出后缀表达式，直到栈空；(b)当栈空时，当前元素入栈。
(5)在处理到中缀表达式结尾时，将栈内元素顺序出栈，写入后缀表达式。
但是，如果考生对上述算法不甚了解甚至一无所知呢?下面假定考生不知道上述算法，仅凭对试题说明和流程图的理解来解答试题。
首先，要对流程图中所使用的符号有一个全面的掌握；其次，要结合问题，对流程图中各操作的含义有一个正确的把握；最后根据已有的分析进行解答。
在试题说明中提到 3 个数组和 1 个函数，前两个数组不过是存储中缀表达工和后缀表达式的数组而已，没有多少可以发挥的地方。但是关于数据组 S［］和函数 PRIOR(CHAR)，需要更为深入的理解。
熟悉数据结构的考生应该对栈的操作不陌生。栈是一种后进先出的数据结构，对栈共有 5 种操作：(1)元素入栈；(2)栈顶元素出栈；(3)丢弃栈顶元素；(4)判断栈是否为空；(5)判断栈是否已满。看流程图，判断"S［p］；'('"给我们的提示 p 在该流程图中用作栈顶指针；操作 p-1→p 应该是丢弃栈顶元素的操作；判断"P:0"应该是判断栈是否为空的操作。那么，进栈、出栈和判断栈是否满的操作呢?
第 1 步的分析是该流程图中没有判断栈是否满的操作。理由是这样的：一是问题规模较小，不需要占用大量的存储空间作为栈，如果硬性规定栈的大小，加入判断栈是否满的操作，会使流程图变得更加复杂；二是流程图中已经没有多余的变量来限定栈的大小，因为从流程图中可以看出，变量 i 用作循环环变量，指示中缀表达式的当前元素，变量 p 用作栈顶指针，变量 k 用作指示后缀表达式的当前写入位置。
但是进栈和出栈的操作在流程图中没有体现，结合问题 2 和问题 3，可考虑这两个操作是否与这两个问题有关。
第 2 步的分析是应该有比较元素优先级的操作。因为既然有返回优先级的函数，必定有比较优先级的操作，否则返回的优先级也就没有什么意义。
现在来阅读流程图。让思路沿着流程图行进，当读到判断"IN［i］=?"时，流程图分成了 4 个分支。
(1)当当前元素 IN［i］为变量名时，操作"IN［i］→POLISH［k］"显然是将变量名直接写入后缀表达式的操作。
(2)当当前元素 IN［i］为左括时，执行子程序 A。
(3)当当前素元 IN［i］为右括时，如果栈顶元素为括号，则丢弃栈顶元素；否则调用子程序 B，直到栈顶元素为左括号，将栈元素丢弃。

(4)当当前元素 IN［i］为运算符时，分为两种情况：(a)如果栈空，则元素入栈；(b)如果栈不空，则执行判断填空①，如果结果为大于号，则执行子程序 A，否则，执行子程序 B，直到栈空或判断填定①的结果为大于号时再执行子程序 A。

处理完一个元素后，循环变量 i 递加，继续处理下一个元素。处理完中缀表达式后，如果栈非空，就执行子程序 B，直到栈空，然后输出后缀表达式。

读完流程图后，就会注意到，子程序 B 总是与栈空联系在一起的。在栈的操作中，导致栈空的操作有两种，一是丢弃栈顶元素，二是栈顶元素出栈。子程序 B 中几乎是不可能包含丢弃栈顶元素的操作的，比较中缀表达式和后缀表达式，可以看出在转换过程中消失的元素是左括号和右括号，而这部分操作在处理右括号时就做了(右括号不入栈，左括号被丢弃)。因此可以断定，在子程序 B 中包含的是栈顶元素出栈的操作。如果这里包含出栈的操作，那么出栈后的元素应该送到哪里去?显然，如果是运算符，应该写入 POLISH［］中。在子程序 B 中会有对元素优先级的比较吗?结合以上的分析，从最后一次使用子程序 B 的情况看，不会有比较元素优先级的操作，因为中缀表达式已经处理完毕，现在需要的是将栈内元素全部出栈。这样对子程序 B 的内容有一个大体的假设。

现在看子程序 A，遇到左括号时需要执行子程序 A。从题中列举的中缀式表达式看，该子程序不能包含出栈的操作，具体理由不解释。结合判断"S［p］：'('"，我们知道这里包含入栈的操作，因为后面要判断栈顶元素是不是左括号。此外，在子程序 A 中是否可以包含对元素优先级的判断?从以上的分析可以看出这是不可能的，因为在例子中左括号是第一个元素，没有比较对象，不大可能是向后比较。

基于以上分析，判断填空①，此处是否就是对元素优先级的比较?应该是的，因为结合试题说明可以断定在流程图中除了对元素优先级的比较外，没有其他可以比较大小的对象。但具体参加与比较的元素却需要考虑，这里假定５种情况：

(1)中缀表达式当前元素与后继元素的比较。这种情况的可能性比较小，在流程图中后续元素影响出栈是不合乎逻辑的。

(2)后缀表达式的当前元素与后续元素的比较。这种情况的可能性比较小，因为当前元素影响写入的情况也不合乎逻辑。

(3)中缀表达式的当前元素与栈顶元素比较。这种情况的可能性比较大。

(4)后缀表达式的当前元素与栈顶元素比较。这种情况的可能性也比较大。

(5)栈顶元素与其下面在栈元素的比较。这种情况的可能性比较小，与栈内其他元素比较以决定是否出栈是不合乎逻辑的。

以上述分析为基础，在头脑中模拟程序的运行过程，看这样是否能够解决问题。通过这样的模拟运行，就能够对以上分析作更精确的判断，从而正确地解答试题。在解答了问题 1、问题 2 和问题 3 后，问题 4 的答案即可得出。详细的模拟运行过程和具体答案，这里应当再重复了。

如果说这里的解题过程有什么经验可循，就是透彻的理解和周密的思维，另外分析问题的速度也是一个重要方面。以上解析达 2000 多字，其实在考试中一定要在 30 分钟之内完成这些分析，否则就要占用其他试题的解题时间了。

【答案】

［问题 1］PRIOR(IN［i］:PRIOR(S［P］)

［问题 2］

功能：将当前符号 IN［i］入栈。

操作：p+1→p

IN［i］→P［p］

［问题 3］

功能：出栈(将栈顶元素送往数组 POLISH［］)
操作：k+1→p
S［p］→POLISH［］
P-1→p
［问题 4］AB+CD*-EF-*G

试题 3（2000 年试题 3）

阅读以下说明和流程图(如图 6-3～图 6-6 所示)，回答问题 1 和问题 2，将解答写在答卷的对应栏内。
［说明］
某供销系统接受顾客的订货单。当库存中某配件的数量小于订购量或库存量低于一定数量时，向供应商发出采货单；当某配件的库存量大于或等于订购量时，或者收到供应商的送货单并更新了库存后，向顾客发出提货单。该系统还可随时向总经理提供销售和库存情况表。
该供销系统的分层数据流图中部分数据流和文件的组成如下：
文件
配件库存=配件号+配件名+规格+数量+允许的最低率库存量
数据流
订货单=配件号+配件名+规格+数量+顾客名+地址
提货单=订货单+金额
采货单=配件号+配件名+规格+数量+供应商名+地址
送货单=配件号+配件名+规格+数量+金额
假定顶层图是正确的，"供应商"文件已由其他系统生成。
［问题 1］
指出哪张图中的哪些文件可不必画出。
［问题 2］
指出哪些图中遗漏了哪些数据流。回答时用如下形式之一：
(1)××图中遗漏了××加工(或文件)流向××加工(或文件)的××数据流；
(2)××图中××加工遗漏了××输入(或输出)数据流。
【解析】
流程图试题的解答较其他类型的试题要直观一些，但要求考生必须特别仔细，百密一疏也会导致严重的失分。
首先阅读试题说明，明确供销系统的基本结构。基本结构如项层图(图 6-3)所示，供销系统与顾客、供应商和总经理之间交换数据。供销系统接受顾客的定货单，反馈不合法的定货单，并在适当的时机向顾客发出提贷单；供销系统向供应产发出采购单，接受送货单，并反馈不合法的送货单；在以上购销操作的基础上，该系统接受总经理的并返回销售及库存情况表。

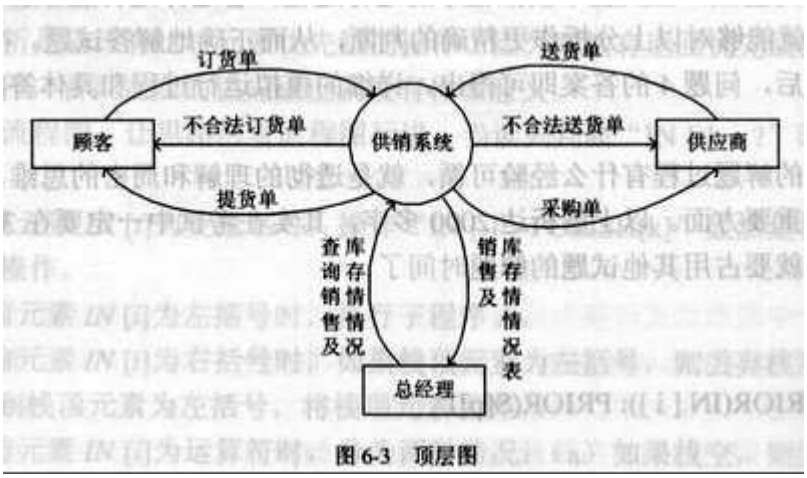


图 6-3　顶层图

问题 1 要求指出哪张图中的哪些文件是不必画出的。这是一个难度较高的要求。首先，它要求仔细研究每一张图，确认每张图中每个文件是否必须画出；其次所谓不必画出，其含义是该文件画出也不会导致该流程图发生错误，但不画出该文件将使整个流程图的层次结构更科学、更清晰。一般来说，如果一个元素(文件如数据流等)只用于一个加工，那么它就不必在其上层流程图中画出。但试题说明中只假定了顶层图的正确性，对于其他流程图，我们首先要确定它们是否完整、是否正确。

对照试题说明，顶层图简单明了，甚至没有涉及文件，因此我们可以跳过项层图不予考虑，直接研究 0 层图(图 6-4)。0 层图中涉及到 3 个文件，即配件库存、缺货订单和采购清单。这里缺货订单和配件库丰都在销售和采购两个加工中使用，只有采购清单只用于采购加工的似乎应该判断采购清单是不必画的，但由于试题不保证项层图以外各加工子图的正确性，难道就此确定在 0 层图采购清单文件不必画出的?不，在各个加工子图中，有可能发生文件、数据流、加工的遗漏或错画，影响我们在此处的判断。因此，只有通过仔细研究以上 3 个文件在各个加工子图中应用印证我们的判断。

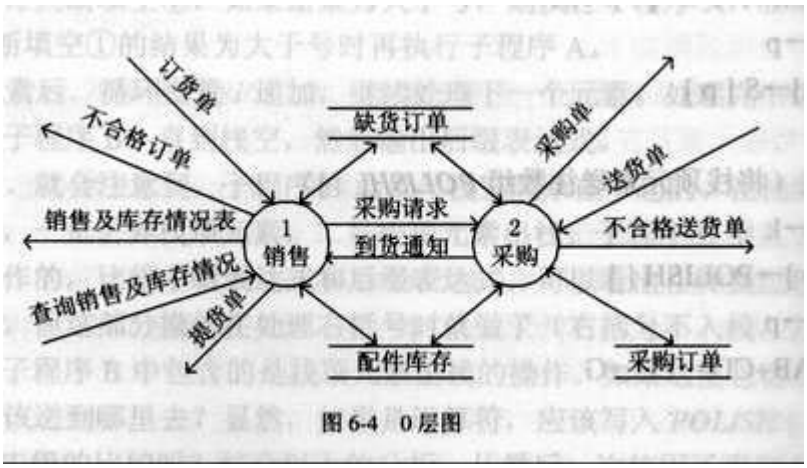


图 6-4　0 层图

首先看配件库存文件。在加工 1 子图(图 6-5)中，配件库存文件经过加工 1.4 更新库存而修改，然后还要为加工 1.1 和加工 1.2 提供数据支持，在加工 1.1 中，通过检查订货单的订购的配件在配件库存中是否有记录来确定该配件是否属于经销范围，然后以此来确定订货单是否合法。其次，还要在加工 1.2 中根据比较合法订货单的数量与库存数量，以此确定是否需要发出采购请求。再交，在发出提货单后，如果库存量低于允许的最低库存量，也要发出采购注。可以看出，在该加工图中，未发现对配件库存文件的不适当的使用。在加工 2 子图(图 6-6)

中，在加工 2.4 核对送货单后要更新库存文件然后加工 2.3 计算增量提供数据支持，计算出需要采购的配件的数量。应该说这里对配件库存 文件的使用了是正确的。因此我们可以确定配件库存文件在加工 1 和加工 2 中都有应用，画在 0 层图中是合适的。

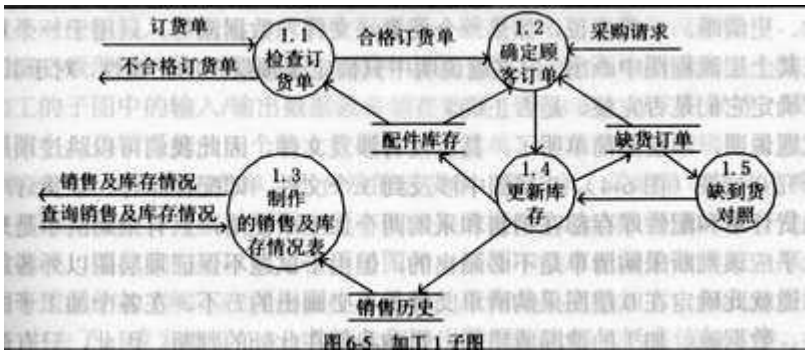


图 6-5　加工 1 子图

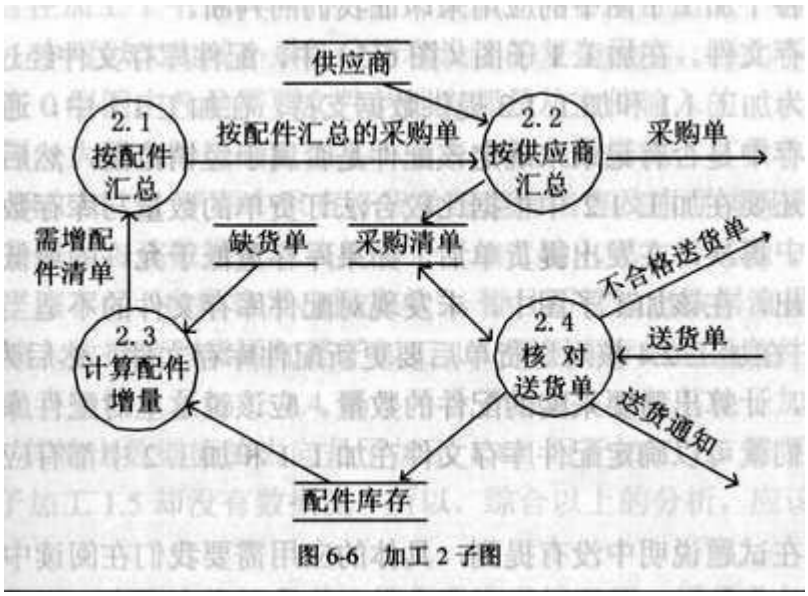


图 6-6　加工 2 子图

缺货订单文件在试题说明中没有提到，具体的应用需要我们的阅读中体会。0 层图中显示该文件被加工 1 和加工 2 使用，且在两个加工中也都有所有应用。现在的任务就是确定两个加工中对该文件的使用是正确的。在加工 1 中缺货订单的数据来自加工 1.2 和加工 1.4，在加工 1.2 中，接受合法的订货单后，如果订货单上数量大于配件库存文件的数量，将产生缺货订单文件。在加工 1.4 中，在有货订单送达后，需要交出提货单，然后更新库存如果库存量低于允许的最低库存量，应该将缺货信息反映到缺货订单中。在加工 2 中，缺货订单为加工 2.3 提供部分数据支持，在计算配件增理以明确需要增加的配件的清单时，需要参照缺货订单。结合以上对使用缺货订单文件的分析，可以认为该文件在加工 1 和加工 2 中的应用都是正确的，因此将该文件画在 0 层图中是有必要的。

再来看采购清单。0 层图显示该文件只应用于加工 2。采购清单如果就应用在加工 1 中，最有可能的地方是加工 1.2 及其后的采购请求数据流，但在这里采购清单是有必要的吗?从加工 2 中可以看出，采购清单是按供应商对采购单进行汇总来产生的，对采购单进行汇总，显然已经属于加工 2 的工作，不应在加工 1 中予以反映。由此可以断定在加工 1 中不应该有采购单文件。所以在 0 层图中画采购清单是不合适的。

以上考察了 0 层图、加工 1 子图和加工 2 子图中缺货订单、配件库存及采购清单的使用，下面研究加工 1 中的销售历史文件。如果没有该文件支持，加工 1.3 是无法制作销售及库存情况表的；而且该文件由加工 1.4 更新库存产生，其输入数据流和输出数据流均在加工 1 中，

在加工 1 中使用该文件是合适的。

问题 2 要求指出哪些图中遗漏了哪些元素。一般来说，这类题目的解答首先要考虑各层次图的数据平衡，其次要考虑加工的输入数据流和输出数据流要平衡，即保证加工的输出数据流都有其对应的输入的数据流。所谓数据平衡，就是在多层次数据流程图中，父图和子图之宰的数据流必须保持一致，比如说在父图中某加工有 2 个输入数据流和一个输出数据流，那么在该加工的子图中的输入/输出数据流必须在数目上和内容上与父图保持一致。

在顶层图中，供销系统的 3 个输入数据流(订货单、送货单与销售及库存情况)和 5 个输出数据流(不合法的订货单、不合法的送货单、提货单、采购单与销售及库存情况表)在 0 层图中都得到反映。考察 0 层图中所有的输入数据流和输出数据流，没有发现超顶层图中是否会有遗漏的加工呢?在顶层图中总经理的查询是单立的加工，但在 0 层图中却给包括到加工 1 中去了，成为加工 1.3，其输入或输出数据流也都包括在该加工中。考察加工 1 子图，加工 1.3 包含在加工 1 中是科学的，因为该加工需要来自加工 1 数据支持，且事务简单，包含在加工 1 中可以大大减小系统分析和设计的复杂程度。因此不能说 0 层图遗漏了加工。至于文件，在 0 层图中凡是需要文件的地方都是从文件输入的，未发生遗漏现象。

仔细研究 0 层图，大体上确定加工 1 有 3 个输入数据流(订货单、到货通知和查询销售及库存情况)，便在加工 1 子图中只有订货单与查询销售及库存情况两个数据流，显然遗漏了输入数据到货通知，该数据流应该从哪个子加工输入呢?看试题中的说明"收到供应商的送货单并更新了库存后，向顾客发出了提货单"。暂且不管输出数据流提货单，根据试题说明，至少可以判定输入数据流到货通知是子加工 1.4 更新库存的前提条件，没有到货通知，就无法更新库存。是直接输入加工 1.4 吗?再看子加工 1.4 与其他子加工的关系。子加工 1.5 是缺到货对照，其输出数据流的去向是子加工 1.4。而子加工 1.5 需要到货通知的支持，但从子加工 1.4 到子加工 1.5 却没有数据流。所以，综合以上分析，应该在子加工 1.5 处输入数据流到货通知。

考察 0 层图，加工 1 有不合格订货单、销售及库存情况表、提货单与采购请求 4 个输出数据流。但在加工 1 子图中，却不有输出数据流提货单，这显然就不符合数据平衡原则的。但该数据流应从何处输出呢?仍旧是上面引用的试题说明，在更新库存后向顾客发出提货单。显然这里合适的位置是子加工 1.4。

在 0 层图中加工 2 有 3 个输出数据流，分别是采购单、不合格送货单与到货通知。在加工 2 子图中这 3 个输出数据流都有反映，因此可以确定在加工 2 子图中没有遗漏输出数据流。在 0 层中，加工 2 有两个输入数据流(送货单与采购请求)，显然加工 2 子图中遗漏了输入数据流采购请求，问题是该数据流从哪个子加工输入。

子加工 2.4 的任务是核对送货单，显然采购请求不应该在这里输入；那么在子加工 2.1、子加工 2.2 和子加工 2.3 中,究竟应该在何处输入采购请求数据流呢?如果在子加工 2.2 中输入，则由子加工 2.1 到子加工 2.2 数据流按配件清单必将是不完全的，因为这样只计算了缺货订单上需要的配件数量。所以，只能从子加工 2.3 输入采购请求。

由于上面已经确认 0 层图没有遗漏的数据流,我们就不必担心在 0 层图中遗漏了加工 1 和加工 2 中的某个数据流。

接着看加工 1 子图和加工 2 子图中的各个子加工,看它们的输入数据流和输出数据流是否平衡。子加工 1.1、1.2、1.4、1.5 和子加工 2.1、2.2、2.3、2.4 都没有问题，只有子加工 1.3，其输出是销售及库存情况表，但在图中该子加工只从文件销售历史中输入数据，销售历史中显然不包含反映库存情况的数据，因此我们可以断定，加工 1 子图中遗漏了从文件配件库存到子加工 1.3 的数据流。

最后再仔细研究整个数据流程图，未发现数据不平衡现象。

【答案】

［问题 1］
0 层中的"采购清单"不必画出。
［问题 2］
加工 1 子图中遗漏了"配件库存"文件到 1.3 加工的数据流。
加工 1 子图中 1.4 加工遗漏了"提货单"输出数据流。
加工 1 子图中 1.5 加工遗漏了"到货通知"输入数据流。
加工 2 子图中 2.3 加工遗漏了"采购请求"输入数据流。

5.2 1999 年度软件设计试题解析

1999 年度下午试卷共有 3 道软件设计试题，选答 2 道，比较容易。试题 1 的 3 个问题是历年试题中比较常见的类型 ，没有出现描述某个处理流程、修改流程图以适应新的要求等令人生畏的问题。与历年试题比较，由于问题的规模较小，试题 2 也相对容易。试题 3 给出了一个翻译流程的状态转换矩阵和语义动作矩阵，要求考生根据句法图、流程图和矩阵中已有内容填写两个矩阵。这种试题的解答的关键是要在理解句法图和语法图的基础上理解各种状态的含义，由此可以确定各种状态下相应的语义动作。在理解过程中，要注意利用矩阵中已有部分来推测未知部分。

试题 1（1999 年试题 1）
阅读以下说明和流程图 6-7，回答问题 1 至问题 3，将解答写在答卷的对应栏内。
［说明］
本流程图描述了某仓库物品入出库管理的处理流程。每张入库单都由两们操作员分别录入，经处理 1 或处理 3 输入系统合作性检查，并将合法的入库单或出库单记入入库单文件或出库单文件。然后通过处理 2 或处理 4 实时更新库存文件。处理 5 每周执行一次，它依次检查库中的每一种物品，当某物品的库存小于该物品的最低库存量时，制订采购计划，输出订购单。处理 6 和处理 7 每月执行一次，处理 6 将入库单文件和出库单文件合并成月入库文件，并根据统计的要求对其进行排序。处理 7 进行统计，产生月报表，并把该月合并后的月入出库文件添加到月入出库后备文件中，以备日后查找。最后清除入库单文件、出库单文件和月入出库文件。
系统中某些文件和报表的格式如下：
库存文件记录。物品编号+名称+规格+库存量+最低库存量+最高库存量(其中"最高库存量"指该物品允许存放在库中的最大值。)
入库单文件记录。日期+物品编号+数量
出库单文件记录。日期+物品编号+数量
月报表格式

物品编号	日期	入库数	出库数
××××	××	××	××
××	××	××	××
	××	××	××
		……	
	当月小计	×××	×××
××××	××	××	××
	××	××	××
		……	

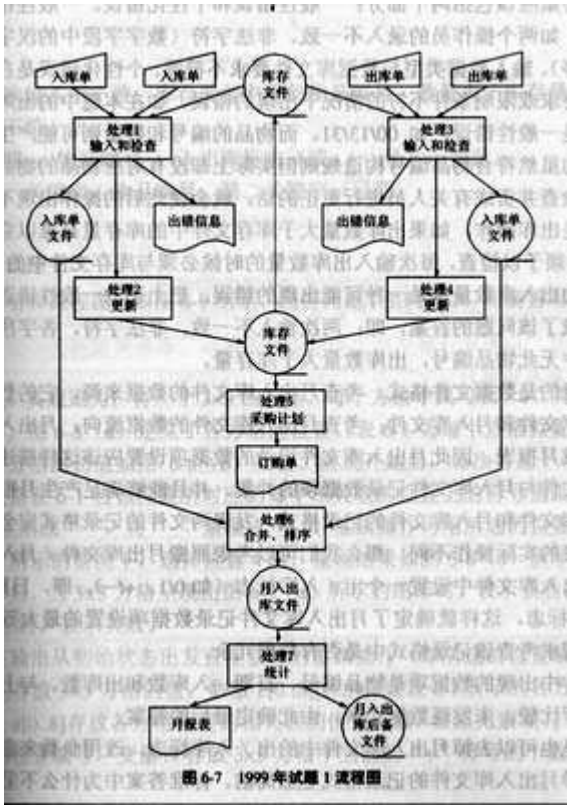


图 6-7　1999 年试题 1 流程图

［问题 1］

指出处理 3 能检查出库单中的哪些错误。

［问题 2］

指出月入出库文件的记录格式。

［问题 3］

指出处理 6 排序的第一和第二关键字。

【解析】

该题涉及的是某仓库的物口出入管理系统的处理流程图。仔细阅读试题说明和流程图，会发现需要特别注意处理 3 和处理 6

问题 1 涉及的是数据输入与合法性检查，这里需要回答在处理 3 中能够检查出哪些错误。关于数据合法检查的问题，几乎在历年的软件分析试题中都会出现。解答此类题目也有一定的套路，即答案应该包括两个部分：一般性错误和个性化错误。一般性错误即通常都会出现的录入错误，如两个操作员的录入不一致、非法字符(数字字段中的汉字字符、非数学标记符号的字符等)、输入数据类型与数据库文件要求不符等。个性化错误是在输入过程中与具体的数据文件要求或限制文件不符的情况下出现的错误，如在本题中的出库单文件中关于日期可能产生的是一般性的错误，如 00/13/31。而物品的编号和数量则可能产生个性化错误，如由于笔误造成的虽然符合物品编号构造规则但实际上却没有对应物品的物品编号。如果不在处理 3 中予以检查并责成有关人中员进行更正的话，就会使此后的操作出现不可预测的错误。另外，由于这是出库操作，如果出库数量大于库存文件中的库存量是难以完成一次出库操作的，在处理中必须予以检查，每次输入出库数量的时候必须与库存文件中的库存量进行比较。同时，非正数的出入库数量也是一种可能出现的错误。把上述的一般性错误和个性化错误合并在一起就形成了该问题的答案，即：两次输入不一致，非法字符，各字段的数据类型不一致，库存文件中无此物品编号，出库数量大于库存量。

问题 2 涉及的是数据文件格式。考查月出入库文件的数据来源，它的数据来源是经过合并排

序的月出库文件和月入库文件；考查月出入库文件的数据流向，月出入库文件经过处理 7 的统计后形成月报表。因此月出入库文件记录的数据项设置应该这样描述：其最大范围不应超出月出库文件与月入库文件记录数据项的并集，并且能够满足产生月报表的数据需求。

考查月出库条件和月入库条件的记录格式，发现两文件的记录格式完全相同，所不同的是每条记录代表的实际操作不同。那么我们考虑照月出库文件/月入库条件的记录格式，然后在月出入库条件中设置一个出/入库标志(如 0/1，+/-)，即，日期+物品编号+数量+出入库标志，这样就确定了月出入库文件记录数据项设置的最大范围。再根据月报表文件的数据需求考查该记录格式是否数据冗余。

月报表文件中出现的数据项是物品编号、日期、入库数和出库数。与上述的月出入库记录格式进行比较，未发现冗余，由此确定最后答案。

其实在这里也可以去掉月出入库文件中的出/入库标志，改用数来表示出库，正数来表示入库，这样月出入库文件的记录格式更加简洁。标准答案中为什么不采用这种方案，请读者自己思考。

其实问题 2 的解答相当容易，在这里用这么多文字进行解析，关键是要说明解答此类试题的科学思维方示，即通过数据来源确定记录中数据设置的最大集合，通过数据流出方向消除其数据冗余。对于复杂的试题来说，难度主要集中在消除数据冗余这一部分，希望读者在复习备考过程中加强这方面的练习。

问题 3 考查的排序关键项问题。首先应该明确，排序的目的是为了便此后的处理，提高系统的效率，但只有选择正确的排序关键项才能达此目的，否则只能起反作用。这也是解答此类题目的关键所在。

处理 6 中的排序产生月出入库文件。如果不继续往下看，我们能够意识到可以以日期和物品意识到可以以日期和物品编号为排序关键项，便没有更多的证据来确谁是第一关键项，谁是第二关键项。

考查处理 7，处理 7 使用的月出入库文件。如果不继续往下看，我们能够意识到可以以日期为物品编号为排序关键项，但没有更多的证据来确定谁是第一关键项，谁是第二关键项。

考查处理 7，处理 7 使用的月出入库文件就是处理 6 合并排序的结果，那么关键项的高以置必须有利于处理 7，而处理 7 产生的两个结果中，从月报表格式可以明显看出是先以物品编号为主，然后再按日期罗列出/入库操作。这样，我们就可以确定，如果处理 6 中的排序不把物品编号作为第一关键项的话就是一种失败，产生月报表时的效率必然很低。确定了第一关键项，第二关键项也就很明显了。所以，在处理 6 中必须以物品编号为第一关键项，以日期为第二关键项。

【答案】

［问题 1］两次输入不一致；库存文件无此物品编号；数量大于库存量；非法字符；各字估的数据类型不符。

［问题 2］日期+物品编号+入出库标志(或+/-号)+数量

［问题 3］第一关键字：物品编号；第二关键字：日期。

试题 2（1999 年试题 2）

阅读以下说明和流程图，如图 6-9 所示，回答问题 1 至问题 3，将解答写在答卷的对应栏内。

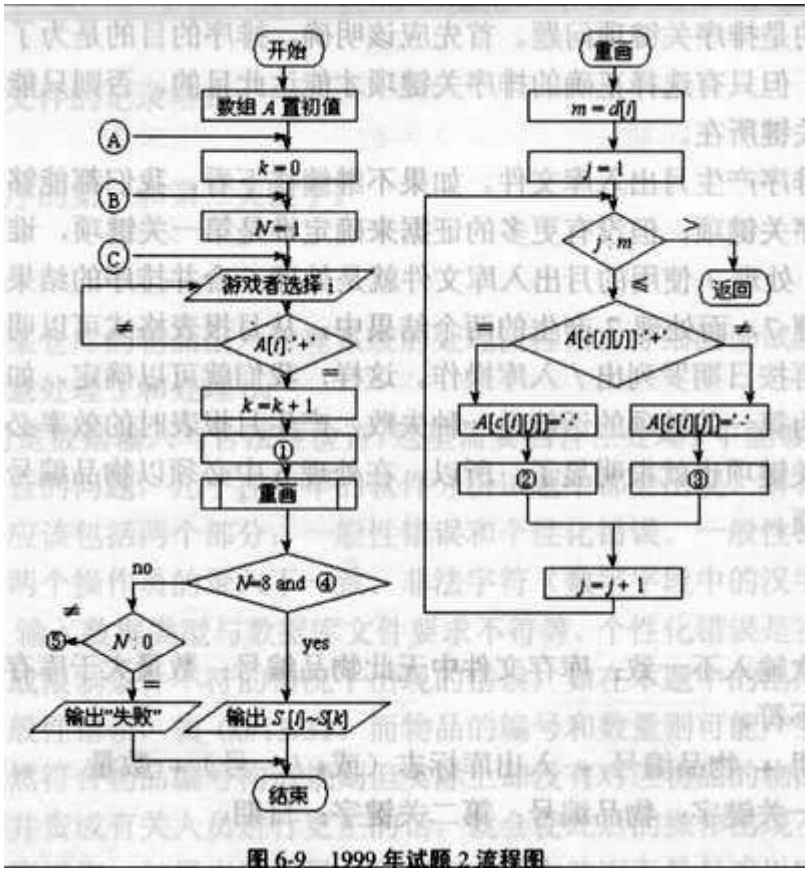


图 6-9　1999 年试题 2 流程图

［说明］

有一种游戏，其规则为：有一个 3×3 的方格，每个方格中只可画"+"符号或"-"符号，表示该方格的值。图 6-8(a)定义了各方格的位置，表 6-1 为每个方格位置定义了与其相关联的位置集，各方格的初值如图 6-8(b)所示。游戏开始后，每次可选一个值为"+"的方格位置，然后根据表 6-1 将该位置所对应的和线个相关联的位置上的符号重画成其不同的符号即将"+"重画成"-"，将"-"重画成"+"。重画操作可用所选位置编号来描述。例如如图 6-8(b)所示的情况下，选择位置 4 时，重画结果如图 6-8(c)所示，经过边续的若干次操作后，当 3×3 方格呈现出图 6-8(d)所示的图形时，表示获胜；当呈现出图 6-8(e)所示的图形表示失败。

表 6-1　方格位置及其相关位置集的对照表

方格位置	相关位置集
0	0134
1	012
2	1245
3	036
4	13457
5	258
6	3467
7	678
8	4578

流程图旨在输出从初始状态出发直至获胜的重画操作(即所选的位置编号)序列。图中假定数组 A［0..8］存放 3×3 方格值，数组 c〔0..8〕〔1..5〕存放表 6-1 所示的各方格位置的相关联的位置集，数组 d〔0..8〕存放各方格位置的相关联的位置个数，数组元素 S〔1〕~S〔k〕存放各次重画操作所对应的位置编号，变量 N 存放 3×3 方格中当前的"+"符号的个数。

012　　---　　-+-　　+++　　---

345　　-+-　　+-+　　+-+　　---

678　　---　　-+-　　+++　　---
(a)　　(b)　　(c)　　(d)　　(e)

〔问题 1〕　填充图中的①~④。

〔问题 2〕　图中的⑤应与 A、B 和 C 中的哪一点连接。

〔问题 3〕　如果每次由游戏者选择方格改由程序自动枚举选择，那么，为从初态出发求出所有可能的获胜重画操作序列，在哪些情况下需要进行回溯处理。

【解析】

这是一道典型的流程图试题，主要考查考生完善和评价流程图的能力，解答此类试题的常用方法是：先弄清楚流程图中各个变量 的含义以及处理过程中各种状态的标志，再分析流程图的宏观结构，然后结合试题说明理解算法说明在流程图中是同哪部分流程图来实现的

首先，通过阅读试题说明，我们将各个变量的含义罗列出来：

(1)数组 A〔0..8〕存放 3×3 方格的值，其值为"+"或者"-"；

(2)数组 c〔0..8〕〔1..5〕存放各方格位置的相关联的位置集；

(3)数组 d〔0..8〕存放各方格位置的相关联的位置个数；

(4)数组元素 S〔1〕~S〔k〕存放各次重画操作对应的位置编号；

(5)变量 N 存放 9 个方格中当前"+"符号的个数。

关于流程图中各个变量的含义，我们结合对流程图的理解来明确其含义。同时，仔细观察成功时各方格的值，并结合变量设置及流程图中关于成功/失败的判断，可以推测流程图中以"+"个数为 8 且位置 4 的方格值为"-"来判断成功，而以"+"个数为 0 来判定失败。

现在让我们带着问题 1 和问题 2 来理解流程图。首先，直到语句框 N＝1，都是处理流程图中的赋初值操作，N＝1 是很好理解的，因为在方格的初值中只有一个"+"，但这里的 k=0 是什么意思?往后看，语句框 k=k+1 没有提供多少关于 k 的有效信息，而输出 S〔1〕~S〔k〕却让我们意识到 k 标志序列中已加入方格的个数，同时，S〔k〕也就表示第 k 个被选中方格的位置。在判断每次选择的都是一个值为"+"的方格之后，流程图进入实际处理过程。而 i 也就是当前被选中的方格的位置，那么在重画部分的 d〔i〕也就是当前被选中方格的相关位置的个数，于是 A〔c〔i〕〔j〕〕也就是当前被选中方格的第 j 个相关位置的值了。

基于以上的理解，由于 S〔k〕存储的第 k 个被选中方格的位置，因此在选定了方格后应该将其位置记入其位置记入对应的数组元素中。考虑到语句框①之后的操作是重画，这里应该完成上述操作因为这是一个循环结构(虽然我们还没有确定填空⑤到底连接到哪里)，并且考虑到 k 和 i 在流程图中的含义，所以我们认为这里应当是 S〔k〕=i。

现在可以跳过重画部分不管，解决填空④和⑤。在经过④的判断后，在 yes 分支中处理了成功怀况下的输出位置序列操作，那么这里的判断应该是判断是否成功。结合上我们成功时方格值分布情况的理解，在填空④中填写 A〔4〕='-'。

那么填空⑤呢?⑤虽然指示了在既未成功又未失败的情况下跳转的目的地，由此构成了一个完整的循环，也就是继续选择。

显然，跳转到 A、B 两处都不能实现继续选择的功能。跳转 A 处几乎是一切从头重来，但数组 A 的值是不确定的，而且变量 N 也不能正确反映"+"的个数，每次循环后总是只有一个"+"，而实际上不应是这样的，除非碰巧在第一次成功，否则很难成功。跳转到 B 处，则可以保证已经选择了方格的个数是正确的，但变量 N 仍不能正确反映"+"的个数。由此可见，跳转 A、B 两处都是不正确的，只能跳转到 C 处。

现在我们来看重画部分的流程图。结合对变量的理解，我们可以认定，填空②和③是对一个相关位置的方格时行重画以后，又进行某些操作。每次重画都涉及"+"数量的增加或减少，这也是在流程图中必须处理的。综合考虑流程图的结构，应在②和③中分别填写 N=N-1 和 N=N+1 来完成上述处理。至此问题 1 和问题 2 都解决了。

问题 3 是关于在使用程序自动枚举选择的情况下何时进行回溯处理的问题。首先，在一次成功或失败时需要回溯处理，这是很明显的；其次，由于每次都必须选定一个值为"+"的方格，如果出现了 9 个方格中只有一个"+"，必须导致失败的情况，那么再次出现这种情况 时应该自动回溯处理，可能的情况如下所示。　另外也不能忽视，在一种枚举选择完毕后必须进行回溯处理，否则就不能够实现穷尽所有成功情况的重画操作。

012	+--	-+-	--+	---	---	---	---	---	---
345	---	---	---	+--	-+-	--+	---	---	---
678	---	---	---	---	---	---	+--	-+-	--+

【答案】

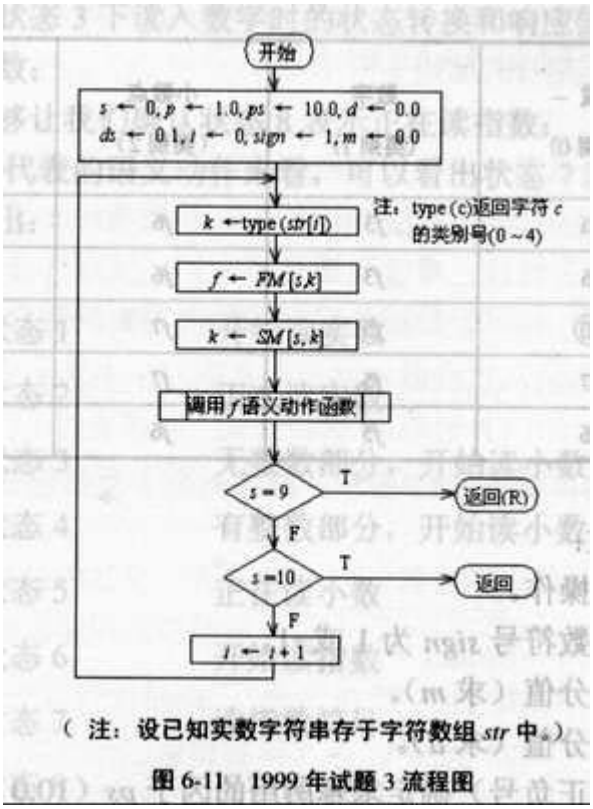
［问题 1］①S〔k〕=i②N=N-1③N=N+1④A〔4〕='-'

［问题 2］　⑤C 相连。

［问题 3］　①获胜。②失败。③出现了以前出现过的图形。④一种图形的枚举选择完。

试题 3（1999 年试题 3）

阅读以下说明和流程图 6-11，回答问题 1 和问题 2，将解答写在答卷的对应栏内。



（注：设已知实数字符串存于字符数组 *str* 中。）

图 6-11　1999 年试题 3 流程图

［说明］

本流程图采用状态矩阵方法将已知字符序列翻译成实数(其句法图如图 6-10 所示)。本题的状态矩阵分成两部分，状态转换矩阵 SM(见表 6-2)和语义动作矩阵 FM(见表 6-3)，它们分别存放每个状态遇到某字符时应执行的语义动作以及执行动作后应转移到的新状态。

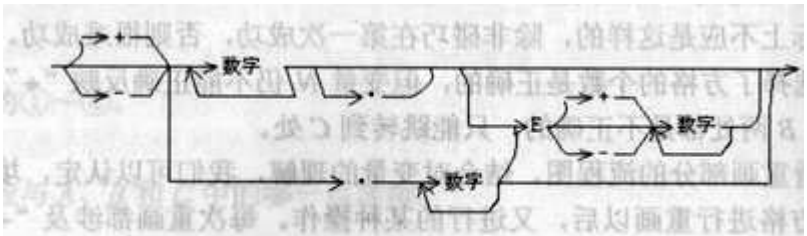


图 6-10　某语言的实数句法图

表 6-2　状态转换矩阵 *SM*

状态 \ 字符类	+ 或 -（类别 0）	数字（类别 1）	小数点（类别 2）	字符 E（类别 3）	其他字符（类别 4）
0	1	2	3	10	10
1	10	①	②	10	10
2	9	2	4	6	9
3	10	5	10	10	10
4	9	③	9	④	9
5	9	5	9	6	9
6	7	8	10	10	10
7	10	⑤	10	10	10
8	9	8	9	9	9

表 6-3　语义动作矩阵 *FM*

状态 \ 字符类	+ 或 -（类别 0）	数字（类别 1）	小数点（类别 2）	字符 E（类别 3）	其他字符（类别 4）
0	f1	f2	f0	f7	f7
1	f7	⑥	f0	f7	f7
2	f6	⑦	f0	f0	f6
3	f7	⑧	f7	f7	f7

本流程图从 0 状态出发逐个读入字符，在执行了 FM 中相应的语义动作后，转移到 SM 中指出相应新状态，重复这一过程，直至到达 9 状态或 10 状态。9 状态表示已正确地把该字符序列翻译成实数(注意，此时已多读进实数后的下一个字符)，10 状态表示出错。

问题 3 考查的是关键项问题，首先应该明确，排序的目的是为了便于此后的处理，提高系统的效率，但只有选择正确的排序关键项才能达到此目的，否则只能起反作用。这也是解答此类题目的关键所在。

现有以下语义动作函数：

f0，空函数，不做任何操作。

f1，按当前字符确定实数符号 sign 为 1 或-1。

f2，翻译实数的整数部分值(求 m)。

f3，翻译实数的小数部分值(求 d)。

f4，按当前字符(幂的正负号)确定求幂所用的因子 ps(10.0 或 0.1)

f5，翻译实数的幂(求 p)

f6，求实数的值 R=sign*(m+d)*p。

f7，输出错误信息。

［问题 1］将状态转换矩阵 SM 中①～⑤处的正确内容填入答卷的对应栏内。

［问题 2］将语义动作矩阵 FM 中⑥～⑩处的正确内容填入答卷的对应栏内。

【解析】

本题的解答关键是确定各种状态的含义，而理解这些状态的可靠信息源则是句法图、状态转换矩阵 SM 和语义动作矩阵 FM。

首先，考查在状态 0 下遇到各种情况时的状态转换及相应的语义动作 。在状态 0 下，如果读入一个"+"或者"-"，要转换为状态 1，并且执行语义动作 f1，按当前字符确定实数符合号 singn 为 1 或者-1；如果读入一个数字，则要转换为状态 2，并且执行语义动作 f2，翻译实数的整数部分值；如果读入一个小数点，则要转移为状态 3，执行语义动作 f0，即不做任何操作；在读入字符 E(指数标志)和其他字符时出错。

如果在头脑中模拟一台计算机来读实数的话，不难判断：

状态 1 表示开始读一个实数；

状态 2 表示正在读一个实数；

状态 3 表示无整数部分，开始读小数。

以同样的方法考虑在状态 2 下读入数字、读入小数点、读入 E 时的状态转换和相应的语义动作，能够得到以下判断：

状态 2 表示正在读一个实数，这在上面已经得出同样的结论；

状态 4 表示在有整数部分的情况下，开始读小数；

状态 6 表示开始读指数。

以同样的方法考虑在状态 3 下读入数字时的状态转换和响应的语义动作，可以认定：

状态 5 表示正在读小数；

状态 6 时读入数字能够让我们确认状态 8 表示正在读指数；

从状态 7 的位置及 f4 代表的语义动作来看，可以看出状态 7 表示指数符号。

把各种状态的含义列出：

状态 1 开始读实数

状态 2 正在读实数

状态 3 无整数部分，开始读小数

状态 4 有整数部分，开始读小数

状态 5 正在读小数

状态 6 开始读指数

状态 7 读指数符号

状态 8 正在读指数

基于以上分析，可以进行试题的解答。

①是在状态 1 下读入数字时的状态转换，自然是转换为状态 2，表示正在读实数；

②是在状态 1 下读入小数点，自然是转换为状态 3，表示无实数部分，开始读小数；

③是在状态 4 下读入数字，自然是转换为状态 5，表示正在读小数；

④是在状态 4 下读入 E，自然是转换为状态 6，表示开始读指数；

⑤是在状态 7 下读入数字，自然是转换为状态 8，表示正在读指数；

⑥是在状态 1 下读入数字需要执行的语义动作，我们可以确定这里需要填写 f2，即翻译实数的整数部分。

⑦是在状态 2 下读入数字，需要执行的语义动仍旧应该 f2；

⑧是在状态 3 下读入数字，其相应的语义动作就应该是 f3，即翻译实数的小数部分值；

⑨是在状态 5 下读入 E 时的相应语义动作，参考状态 2 时读入 E 的语义动作，这里填写 f0；

⑩是在状态 6 下读入"+"或者"-"时的相应的语义动作，理解 f4 所代表的语义动作，这里必须填写 f4。

【答案】

［问题 1］①2 ②3 ③5 ④6 ⑤8

［问题 2］⑥f2 ⑦f2 ⑧f3 ⑨f0 ⑩f4

5.3 1998 年度软件设计试题解析

共 3 道软件设计试题，试题 1 考查数据处理流程，是一道软件分析试题。试题 2 考察软件测试，由于其内容的特殊性(知识内容集中出现)，只要考生对覆盖标准的理解正确，通过建立真值表或其他方法来解答题目，难度应该是不大的。试题 3 是流程图填空及修改，考查的是字符串处理方面的问题，相对于数组处理等其他内容来说直观性和可操作性强。

试题 1(1998 年试题 1)

阅读以下说明和流程图，如图 6-12 所示，回答问题 1 至问题 3，将解答写在答卷的对应栏内。

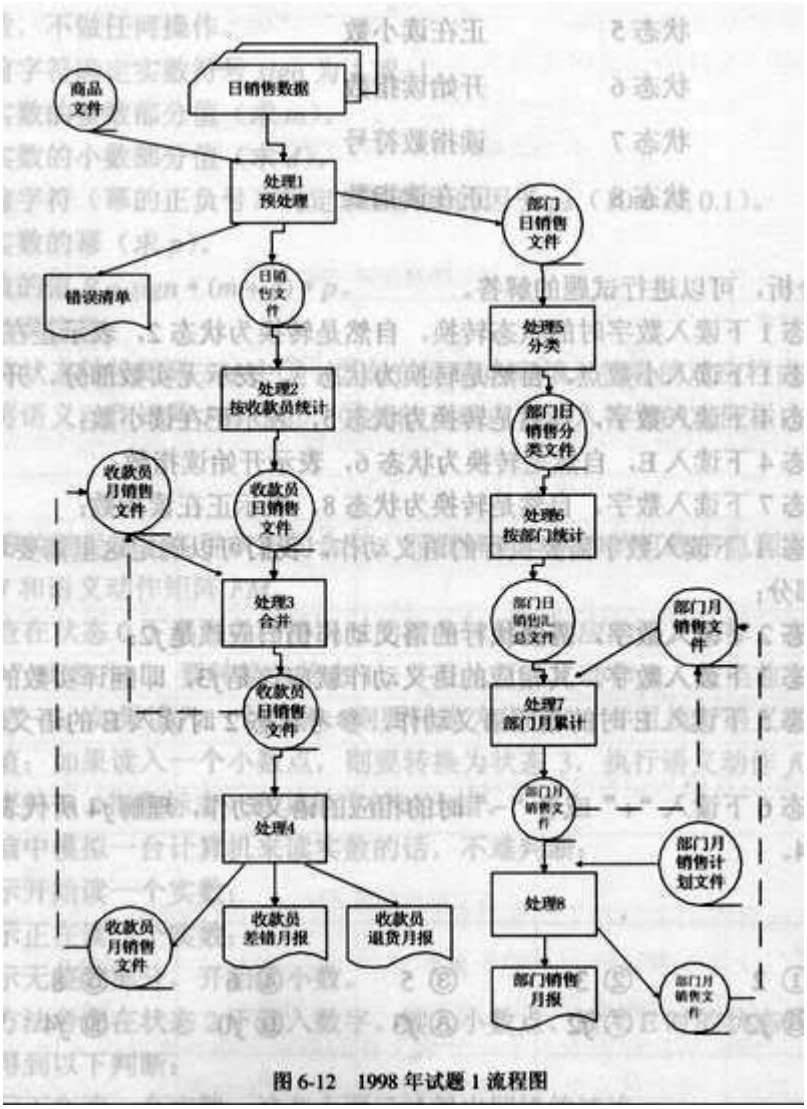


图 6-12　1998 年试题 1 流程图

［说明］

本流程图描述某超市销售数据的部分处理流程。超市中有若干台收款机和若干名收款员。这里，把一个收款员开始使用一台收款机到离开这台收款机称为该书款员的一次作业。作业开始时，收款员先在收款机上输入收款员和作业前金额。作业前金额是为了销售时的找零而在作业前预先放入钱箱的金额数。作业结束时，收款员要打开钱柜，取走全部现金，并把这些现金的金额数(称为作业后金额)输入收款机。

当作业前金额+本次作业售货总金额-本次作业退货总金额≠作业后金额时，表示这次作业存在金额差错。

本流程图已简化，并作以下假定：该超市只有现金交易(不用信用卡和礼券)；一个收款员因某种原因(如吃饭)在一天中可以有多个作业；销售方式只有售货和退货两种。

整个超市分成若干部门（如食品部、服装部），系统按部门统计一个月中各类货物的销售数量和金额，最后根据月销售计划文件分析各部门完成销售计划的情况。系统还统计每个收款员的差错情况和退货情况。

图中处理 4 和处理 8 每月的最后一天执行一次(营业结束后)，其他处理每天执行一次。

图中部分数据、文件的记录格式如下：

日销售文件记录，

(作业开始标识+收款机号+收款员号+作业前金额)

|(售货标记|退货标记)+货号+数量+金额)

|(作业结束标记+收款机号+收款员号+作业后金额)

日销售数据，收款机号+收款员号+作业前金额{(售货标记|退货标记)+货号+数量+单价+金额}+作业后金额

部门日销售文件记录，部门号+(售货标记|退货标记)+货号+数量+金额

部门月销售计划文件记录，部门号+月计划金额

收款员差错月报，月份+收款员号+差错作业数+差错总金额

收款员退货月报，月份+收款员号+退货次数+退货总金额

其中{w}表示 w 重复出现多次；a|b 表示 a 或 b；a+b 表示 a 与 b。

［问题 1］

分别写出收款员日销售文件、商品文件、部门日销售汇总文件至少应包含哪些数据项。

［问题 2］

处理 1 能检查出日销售数据中的哪些错误。

［问题 3］

处理 4 对收款员月销售文件作何种操作。

【解析】

软件分析试题解答的关键在于切实理解软件需求。我们可以看到，该系统要完成 3 项任务，即(1)按部门统计各类货物的月销售量和金额，(2)分析各部门完成销售计划情况，(3)统计每个收款员的差错情况和退货情况。

对于问题 1，需要从软件需求入手，分析文件的数据来源和通过该文件与其他文件共同作用需要产生的数据，确定该文件应该包含的数据项；对于问题 2，要考查处理 1 所处理的数据对象，然后分析该数据对象可能出现的错误；对于问题 3，关键是要看处理 4 的任务和处理前后收款员月销售文件数据的作用。由于本题涉及面广，限于篇幅，本书只能结合 3 个问题的解答来解析本题。

可以看出，商品文件和日销售数据通过处理 1 生成错误清单、日销售文件和部门日销售文件。首先，商品文件应该记录有关商品的一些数据，应该有一个标识商品的记录主键，而在系统中又多处用到货号这个数据项作为处理的依据。因此，选用在商品文件中包含货号，并以之作为记录主键是合理的。在部门日销售文件中可以看到部门号这个数据项，但在日销售数据中却没有该数据项。很简单，在商品文件中应该有部门号这个数据项，否则部门日销售文件中的部门号就没有来源。包含了货号和部门号的商品文件应该基本可以满足处理 1 的数据需求，但这里包含一种潜在的危险性，即日销售数据中的价格存在错误。如果在商品文件中不包含一种潜在的危险性，即日销售数据中的价格存在错误。如果在商品文件中不包含一个关于价格的数据项，这种错误将无法排除，所以商品文件至少应该包含货号、部门号和价格

3 个数据项。

处理 1 的处理对象就是商品文件和日销售数据。根据数据对象的特点来分析可能在以下几个方面出现错误。(1)在日销售数据中存在金额、作业前金额和作业后金额 3 个数据项，保证数据的完整性来考查，对于数字型的数据一般要检查其合法性，即该数据是否在允许的取值范围之内，如作业前金额和作业后金额都不可能是负数，金额的取值是固定的，即应该是数量与单价的乘积。(2)不排除日销售数据中存在非法字符的可能性，因为各种误操作都是可能的。(3)上面已经提到，在商品文件中包含一个价格的数据项，以便检查日销售数据中与货号的正确对应。可能产生进入日销售数据文件的错误也就是这些，通过以上分析不难得出可能发生的错误。处理 1 应该能够检查出所有的错误，所以正确的解答应该包含以上 3 个方面的内容。

继续向下阅读处理流程，发现收款员日销售文件和收款员月销售文件进行了合并，说明收款员月销售文件的数据来源于收款员销售文件进行了合并，说明收款员月销售文件的数据来源于收款员日销售文件，即收款员月销售文件包含的数据项在收款员日销售文件中也应该包含，而且我们看到收款员日销售文件只用到这一次，所以基本上可以判定它们所包含的数据项应该是相同的。基于以上的分析，我们通过分析处理 4 对收款员月销售文件进行处理所产生的数据，来分析收款员日销售文件应该包含的数据项。处理 4 产产收款员月销售文件、收款员差错月报和收款员退货月报。处理 4 的输入和输出中都包括收款员月销售文件，因此在进行分析时该文件可以忽略不计，把收款员差错月报和收款员退货月报进行合并，就可以得到收款员日销售文件的结构，但有些数据项应该改名称，如差错总金额和退货总金额在这里显然是不合适的，可以改为差错金额的退货金额。

处理 4 对收款员月销售文件进行何种操作?因为该处理每月末执行一次，收款员月销售文件在通过处理 4 产生收款员差错月报、收款员退货月服之后已完成了任务，应该清空数据准备在下一个月份里记录收款员销售情况了，所以，在处理 4 中，收款员差错月服和收款员退货月报产生之后，应该删除其中所有记录，即进行初始化。

现在本题只有部门日销售汇总文件应包含哪些数据项没有解决，但部门日销售汇总文件通过处理 7 之后产生的部门月销售文件包含哪些数据项没有明确提示，只有部门日销售文件和部门月销售计划文件可供参考。再回过头仔细阅读本系统需要完成的 3 项任务，需要按部门统计各类货物的月销售量和金额，还需要分析各部门完成销售计划情况。显然，在部门月销售文件中应该包含月销售金额这个数据项，以便与月销售计划中的月计划金额进行比较，该数据项必须从部门日销售汇总文件中累积得到。系统要利用部门月销售文件统计各类货物的月销售量和金额，显然这两项数据的来源也是部门日销售汇总文件，而作为对货物分类统计的依据，数据项货号是必不可少的。除此之外，作为部门日销售汇总文件，应该以部门号作为记录主键。以上从对系统的数据支持角度来分析部门日销售汇总文件应该包括的数据项，如果提到的数据项没有合法来源,那么系统就无法工作。部门日销售汇总文件的数据来源于部门销售文件，这证明分析是符合系统需求的。

【答案】

［问题 1］

收款员日销售文件：（每项 1 分）

收款员号+差错作业数+差错金额+退货次数+退还金额

商品文件：（每项 1 分）

货号+部门号+单价

部门日销售汇总文件：（每项 0.5 分，去除累计和的小数分）

部门号+货号+销售数量+销售总金额

［问题 2］

①存在非法字
②数量×单价≠金额
③货号（或单价）与商品文件中的货号（或单价）不一致
［问题 3］删除其中的所有记录或初始化。

试题 2　（1998 年试题 2）

阅读以下说明和流程图 6-13，回答问题，将解答写在答卷的对应栏内。

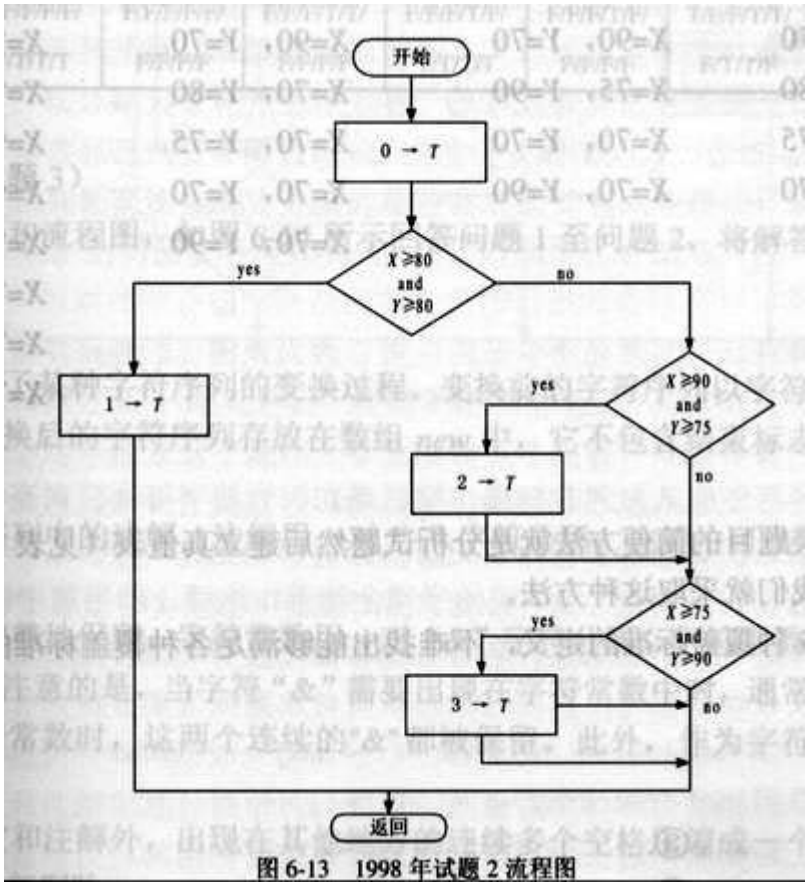


图 6-13　1998 年试题 2 流程图

［说明］
本流程图描述了某子程序处理流程，现要求用白盒测试法为孩子程设计测试数据。
［问题］
根据判定覆盖、条件覆盖、判定一条件覆盖、条件组合覆盖（即多重条件覆盖）和路径覆盖等 5 覆盖标准，从供选择的答案中分别找出满足相应覆盖标准的最小测试数据组（用①～ ⑿回答）。
供选择的答案

① X＝90，Y＝90 X＝70，Y＝70	② X＝90，Y＝70 X＝70 Y＝90	③X＝90，Y＝90 X＝90, Y＝75 X＝75，Y＝90	④X＝90，Y＝75 X＝75，Y＝90 X＝70，Y＝70
⑤ X＝90，Y＝90 X＝90，Y＝75 X＝75，Y＝90 X＝70，Y＝70	⑥ X＝80，Y＝80 X＝90，Y＝70 X＝70，Y＝90 X＝70，Y＝70	⑦X＝80，Y＝80 X＝90，Y＝75 x＝90，Y＝90 X＝75，Y＝90 X＝70，Y＝90	⑧X＝80，Y＝80 X＝90，Y＝70 X＝70，Y＝90 X＝70，Y＝70 X＝75，Y＝75

⑨ X＝80，Y＝80
X＝90，Y＝75
X＝90，Y＝70
X＝70，Y＝80
X＝70，Y＝75
X＝70，Y＝70

⑩ X＝90，Y＝90
X＝90，Y＝75
X＝90，Y＝70
X＝75，Y＝90
X＝70，Y＝70
X＝70，Y＝90

(11)X＝80，Y＝80
X＝90，Y＝75
X＝90，Y＝70
X＝70，Y＝80
X＝70，Y＝75
X＝70，Y＝70

(12)X＝80，Y＝80
X＝80，Y＝70
X＝70，Y＝80
X＝70，Y＝70
X＝90，Y＝75
X＝90，Y＝70
X＝70，Y＝90
X＝70，Y＝75
X＝75，Y＝90
X=75， Y=80
X＝70，Y＝90

【解析】

解答软件测试类题目的简便方法就是分析试题然后建立真值表（见表 6-4），通过真值表来解答试题，一在我们就采取这种方法。

表 6-4　真值表

	X>=80	Y>=80	X>=80 and Y>=80	X>=90	Y>=75	X>=90 and Y>=75	X>=75	Y>=90	X>=75 and Y>=90
①	T/F	T/F	T/F	T/F	T/F	T/F	T/F	T/F	T/F
②	T/F	F/T	F/F	T/F	F/T	F/F	T/F	F/T	F/F
③	T/T/F	T/F/T	T/F/F	T/T/F	T/T/T	T/T/F	T/T/T	T/F/T	T/F/T
④	T/F/F	F/T/F	F/T/F	T/F/F	T/T/F	T/F/F	T/T/F	F/T/F	F/T/F
⑤	T/T/F/F	T/F/T/T	T/F/F/T	T/T/F/F	T/T/T/F	T/T/F/F	T/T/T/F	T/F/T/F	T/F/T/F
⑥	T/T/F/F	T/F/T/F	T/F/F/F	F/T/F/F	T/F/T/F	F/F/F/F	T/T/F/F	F/F/T/F	F/F/F/F
⑦	T/T/T/F/F	T/F/T/T/T	T/F/T/F/F	F/T/T/F/F	T/T/T/T/T	F/T/T/F/F	T/T/T/T/F	F/F/T/T/T	F/F/T/T/F
⑧	T/T/F/F/F	T/F/T/F/F	T/F/F/F/F	F/T/F/F/F	T/F/T/F/T	F/F/F/F/F	T/T/F/F/T	F/F/T/F/F	F/F/F/F/F
⑨	T/T/T/F/F/F	T/F/F/T/F/F	T/F/F/F/F/F	F/T/T/F/F/F	T/T/F/T/T/F	F/T/F/F/F/F	T/T/T/F/F/F	F/F/F/F/F/F	F/F/F/F/F/F
⑩	T/T/T/F/F/F	T/F/F/T/F/T	T/F/F/F/F/F	F/T/T/F/F/F	T/T/F/T/F/F	F/T/F/F/F/F	T/T/T/T/F/F	F/F/F/T/F/F	F/F/F/T/F/F
⑪	T/T/T/F/F/F/F	T/F/F/T/F/F/T	T/F/F/F/F/F/F	F/T/T/F/F/F/F	T/T/F/T/T/F/F	F/T/F/T/F/F/F	T/T/T/F/F/F/F	F/F/F/F/T/F/F	F/F/F/F/T/F/F
⑫	T/T/F/T/T/F/F/F/F	F/F/F/F/F/F/T/T/T	F/F/F/F/T/F/F/F/F	F/F/F/T/T/F/F/F/F	F/F/F/T/F/T/T/T/T	F/F/F/T/F/F/F/F/F	T/F/F/T/T/F/T/T/F	F/F/F/F/F/F/F/T/F	F/F/F/F/F/F/F/T/F

结合真值表和 5 种覆盖标准的定义，不准找出能够满足各种覆盖标准的测试用例，详细过程兹不赘述。

【答案】

判定覆盖 ③

条件覆盖 ②

判定/条件覆盖 ⑤

条件组合覆盖 ⑩

路径覆盖 ⑤

试题 3　（1998 年试题 3）

阅读以下说明和流程图，如图 6-14 所示回答问题 1 至问题 2，将解答写在答卷的对应栏内。

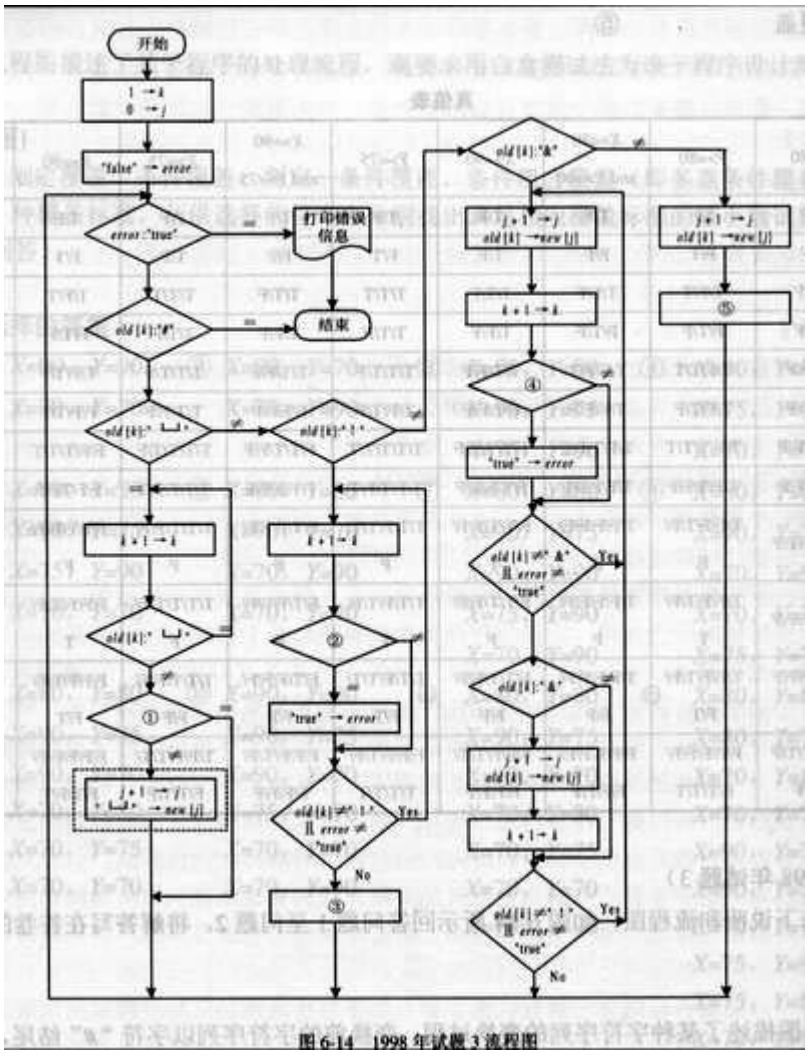


图 6-14　1998 年试题 3 流程图

［说明］

本流程图描述了某种字符序列的变换过程。变换前的字符序列以字符"#"结尾，它存放在数组 old 中，变换后的字符序列存放数组 new 中，它不含结束标志"#"。流程图按下列规则进行变换：

1、删除字符序列中的注解。注解用一对"！"字符分隔，注解中可出现除"！"和"#"外的任何字符。

2、字符常数完整地保留。字符常数用一对"&"字符分隔，字符常数中可出现除"#"外的任何字符。值得注意的是，当子符"&" 需要出现在字符常数中时，通常用两个连续的"&"表示，在保留字符常数时，这两个连续的"&"都被保留。此外，作为字符常数分隔符的一对"&"也被保留。

3、除字符常数和注解外，出现在其他地方的连续多个空格压缩成一个空格，但字符序列中先导的空格则全部删除。

4、注解和字符常数之外的非空格字符均保持不变。

本流程图对原字符序列从左到右扫描，根据遇到的当前字符来搜连续空格、注解或字符常数，然后按上述变换规则变换。若遇到当前字符是"！"，则寻找下一个"！"字符（若找不到则进行出错处理），这两个"！"间的字符全部看作注解。若当前字符是"&"，则寻找与之配对的下一个"&"字符（若找不到则作出错处理），其间的字符全部看作字符常数。

例如：

原字符序例：␣ ␣a␣ ␣ ␣b&cd&& ␣ ␣!e&f ␣ ␣g!h& ␣ ␣i ␣ ␣j#

变换后的字符序列：a␣b&cd&& ␣ ␣!e&f ␣g␣j

本流程图假定在数组 old 中一定存在"#"字符。

［问题 1］

填充流程图中的①～⑤，把解答写在答卷的相应们置上。

［问题 2］

当原字符序列中注解的前后均是连续空格时，本流程图将注解前后的连续空格分别压缩成一个空格，删除注解后，将导致变换后的新字符序列出现两个连续的空格。如：

␣ ␣g!h&␣ ␣i!␣ ␣j

本流程图将变换成 g␣ ␣j

为使变换后的新序列中除字符常数外没有连续的空格，图中的虚线框需作何改动（只需画出修改后的流程图）。

【解析】

程序流程图试题的解题关键在于把流程图看作流程图说明的实现，结合流程图说明理解流程图，确定流程图中各部分与流程说明的对应关系，然后再进行解答就非常容易了。

仔细阅读流程图，发现流程图大致分成 4 个纵列，其中第 1 纵列处理空格问题，第 2 纵列处理注释问题，第 3 纵列处理有关"&" 符号的问题，第 4 纵列处理正常字符。这样，我们就可以把整个流程图分成 4 个部分，在解答某一部分中的空白是时可以完全不考虑其他 3 个部分，大大降低了解答难度。

首先，关于空格处理的流程图说明包括两点，一是除字符常数和注解外，出现在其他地方的多个空格压缩成一个空格，二是字符序列中先导的空格全部删除。从流程图的"开始"处开始阅读，当读到 old［k］：［空格］处时，我们知道从此以下就是关于空格问题的处理，在 old［k］：为空格的情况下，k 值递加，续继比较 old［k］：是否为空格。当发现当前字符并非空格时，下面是填空①，我们暂且不管这个判断情况下可以跳过这步操作呢？即以上空格为多个空格压缩成一个空格的操作。究竟在什么情况下可以跳过这步操作呢？即以上空格为字符序列的先导空格时，可以跳过空格压缩的操作，所以该空白应该是对前位置的判断。从流程图中我们可以非常容易地确定 k 为数组 old 当前位置的指针，j 为数组 new 当前位置的指针。很明显，通过对 k 的判断不能确定处理过的空格是否为先导空格的，而当处理过的空格为先导空格时，数组 new 的当前位置应该动为 0，所以在填空①处填写"j:0"就是比较容易的事情了。

在这里可以先解答问题 2，因为问题 2 也是关于空格处理的，与流程图的基他部分没有关系。在读懂了流程图的第 1 纵列之后，应趁机这个问题 2。形成连续空格的原因是同于注解的阻隔，但我们绝对不能从这个原因入手来解决这个问题，那将使修改后的流程图规模膨胀，使本来简单的问题复杂化。问题的关键是数组 new 中不能出现两个连续的空格。虽然我们不能从原因入手解决问题，但却可以控制结果，即当需要往数组 new 中写入空格时，先判断数组 new 中当前字符是否为格。如果当前字符为空格，就不再写入空格，而是继续处理以后的字符。有了上述的分析，问题的解答就是非常容易的事情了。这只是一个思想方法的问题，问题的本身并没有难度，并键就是在考虑解决问题的方案时要从不同的角度考虑，找出最简单可行的办法。

填空②是在处理注解问题的纵列内，从流程图常识我们可以判断填空②应该填写一个判断语

句，而且后续语句在该判断为真值时完成为出错标志赋真值的操作。通过阅读流程图说明可以知道，在处理注解时，唯一导致出错的情况就是注解中出现标志字符序列结束的符号"#"，所以该处的正确解答就是非常明显的，即时判断数组 old 的当前字符是否为字符序列结束标志"#"。填空④的解答思路与此相同。

填空③处的填写内容有一定的难度，但它的难度就在于此前的判断比较麻烦，如不仔细阅读很难搞清楚这些判断的含义，但只有弄明白了上述判断的含义，就不难判定该空白是在一处注解处理完毕之后，继续做处理的准备工作。跳过该空白继续阅读流程图，发现再次进入字符处理之后仍旧要判断组 old 的当前字符是否为空格、"！"或"&"。如果在空白处不能实现指针的向后偏移，则容易把一处注解的结束标志当作另一处注解的开始标志，造成程序的判断和处理错误。填空⑤的解答思路与此相同，这里就不再多说了。

通过对本流程图的解答，可以大致地总结以下有关解答此处类题目的思想方法，即紧密结合流程图说明，理解流程图对说明的实现，则很容易找出流程图说明中提出的而在流程图中没有实现的要求，然后结合对流程图的理解，在正确的位置上填写实现这些要求的语句。

【答案】

[问题 1]

①j:n

②old[k]:"#"

③k+1→k

④old[k]:"k"

⑤ k+1→k

[问题 2]

如图 6-15 所示.

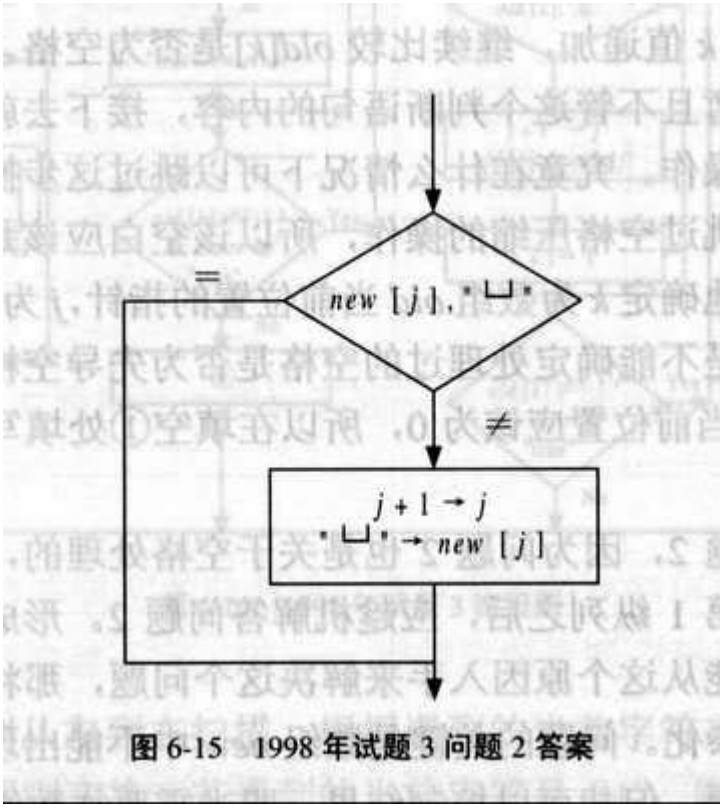


图 6-15　1998 年试题 3 问题 2 答案

5.4 1997 年度软件设计试题解析

1997 年度共 3 道软件设计题。试题 1 是一个数组处理的问题，试题说明很详细，没有涉及算法问题，这种情况在历年的试题中很常见。值得注意的是问题 2 和问题 3，这里没有涉及流程图的修改，而一个具体实例的运算，解答该类题目的关键是对处理流程的掌握，也即对流程图的全面深入掌握，而不是对流程图的修改和评价。

试题 2 考得比较细，主要涉及了 3 个方面的内容；(1) 文件在系统中的作用；(2) 操作的内容；(3) 提高效率的途径。以上 3 个方面的内容在以前的试题中都出现过，要注意题中问题 2 的设计。问题 2 基本上涵盖了上述 3 个方面，是一个比较巧妙的问题，对考生的综合能力的要求比较高。

试题 3 考查了分层的数据流程图， 这方面的内容和软件测试一样，出现的频率较小。其实在解答这类试题时，只要能够贯彻逐步细化和数据平衡的原则，应该是相当容易的。

试题 1 （1997 年试题 1）

阅读以下说明和流程图，如图 6-16 所示，回答问题 1 至问题 3，将解答写在答卷的对应栏内。

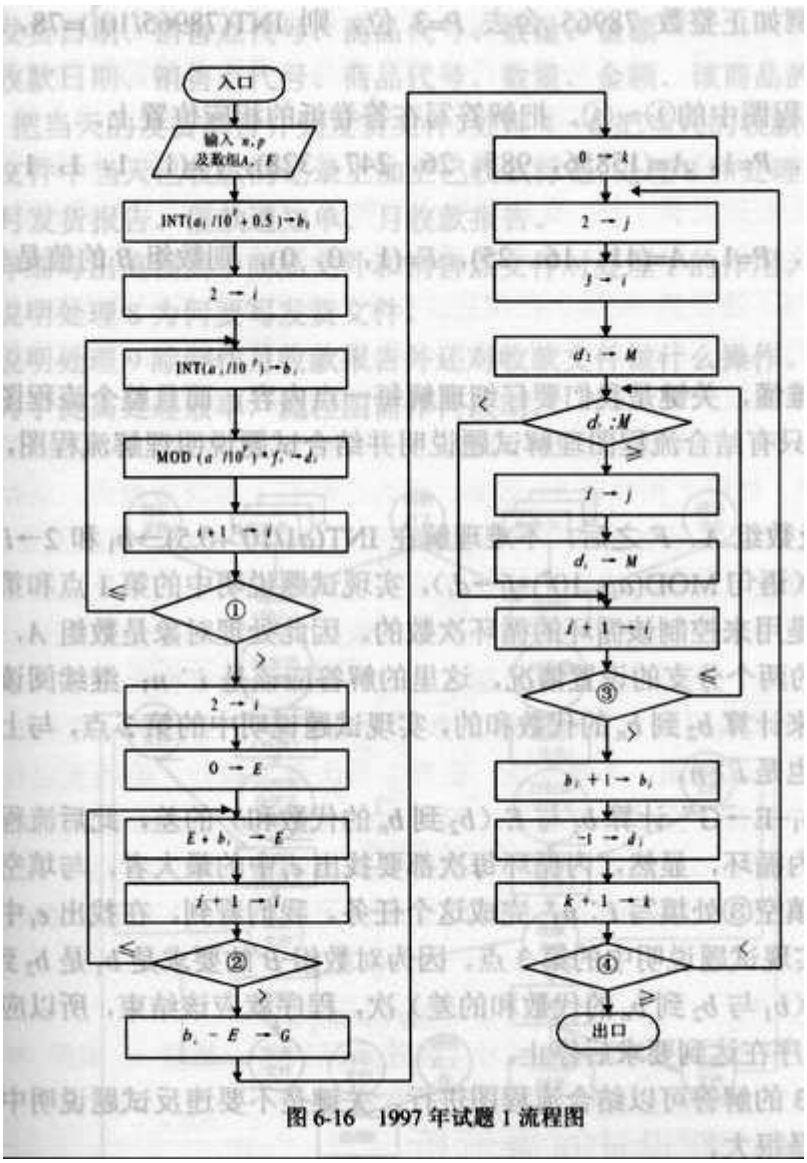


图 6-16　1997 年试题 1 流程图

［说明］

本流程图用来实现一组正整数的加权舍位平衡。已知正整数数组 A(a1a2……an)，满足条件

$a_1=\sum_{i=2}^{n}a_i \quad (n\geq 3)$ 。现将数组 A 中每个数舍去 P 位，得到另一正整数数组 B（b1，b2……bn)。

它满足如下条件：

1.b1 是 a1 是舍 P 位后四舍入五所得，即 b1=INT（a1/10p+0.5)。

2. $b_1=\sum_{i=2}^{n}b_i$

3.bi=INT(a1/10^P)+ei(i=2,3……，n)其中 ei 取值为 0 或 1，当 e1=时，称 ei 第 i 项数据的进位。

4.ei（i=2，3 ……，n）之值根据余数 MOD(ai,10^P)乘上权 fI（f≥0）后的数值大小来决定（其算法见图 15-9 所示的流程图)。权 fi 存放在数组 f 中，其中 INT 是取函数，MOD 是余数函数，例如正整数 7895 舍去 P=3 位，则 INT（78935/10^3）=78，MOD（7895，10^3）=965。

［问题 1］填充流程图中的①～④，把解答写在答卷纸的相应位置上。

［问题 2］若 N=5，P=1 A=（15856，985，26，247，328)，F=（1，1，1，1 ，1）则数组 B 的值是多少？

［问题 3］若 N=3， P=1，A=（41，16，25）F=（1，2、0，0）则数组 B 的值是多少？

【解析】

试题说明并不难懂，关键是我们要仔细理解每一点内容，而且整个流程图实现了试题说明的 1～4 项内容，只有结合流程图理解试验题说明并结合试题说明理解流程图，才能真正理解试题说明和流程图。

在输入 n、p 及数组 A、F 后，不难理解在 INT（a1/10^P+0.5）→b1 和 2→i 之后，循环计算 bi 的基本值和 ei（语句 MOD（ai,10^P）*fi-di)，实现试题说明中的第 1 点和第 4 点要求的操作。显然，填空合①是用来控制该循环次数的。因此处理对象是数组 A，所以其元素个数为 n。结合判决的两个分支的设置情况，这里的解答应该是 i:n；继续阅读流程图，填空②之前的循环是用来计算 b2bn 的代数和的，实现试验题说明中的第 2 点，与上面的解答过程类似，这里的判断也是 i:n。

显然，语句"b1→E→G"计算 b1 与 E（b2 到 bn 的代数和）的差，此后流程图进入一个二重循环。首先研究内循环，显然，内循环每次都要找出 ei 中的最大者，与填空①、②的解答相似我们在填空③处写 i:n，完成这个任务。我们看到，在找出 ei 中的最大者后，相应的 bj 即加 1，实现试题说明中的第 3 点。因为对数组 B 的要求是 b1 是 b2 到 ba 的代数和，当加 1 操作进行 G(b1 与 b2 到 bn 的代数和的差)次，程序就应该结束，所以应在空④处填写 k:G，以控制程序在达到要求后停止。

问题 2 和问题 3 的解答可结合流程图进行，关键是不要违反试题说明中的各种要求，解答的难度应该不是很大。

【答案】

［问题 1］①i:n

［问题 2］159，98，3，25，33

［问题 3］4,1,3

试题 2 （1997 年试题 2）

阅读以下说明和流程图 6-17，回答问题 1 至问题 4，将解答写在答卷的对应栏内。

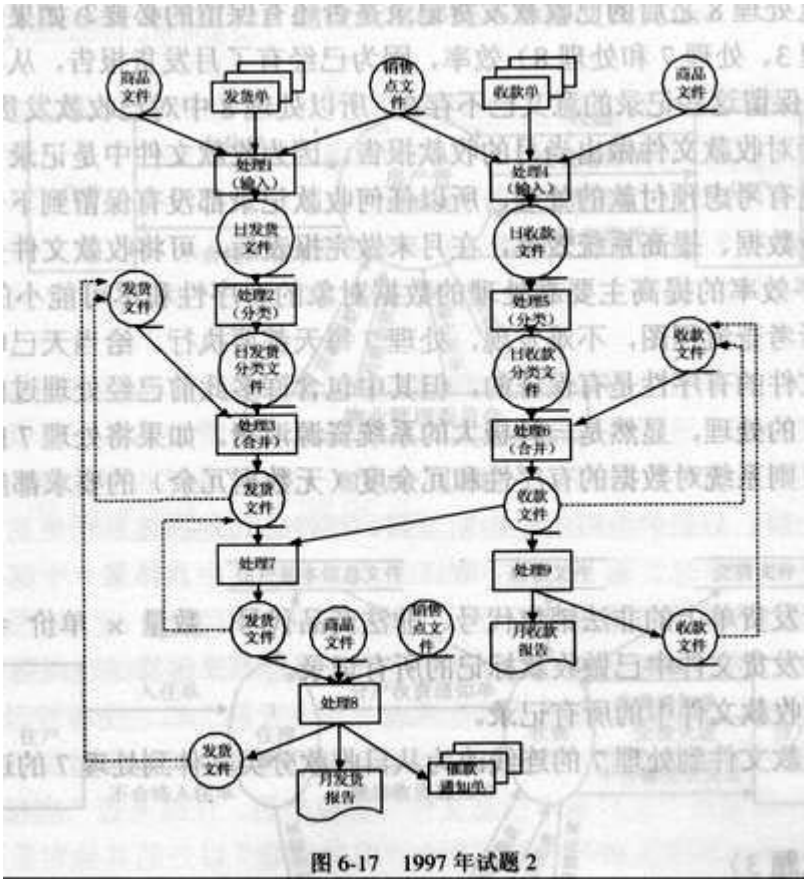


图 6-17　1997 年试题 2

［说明］

某公司将其生产的商品通过若干销售点进行销售。销售点在收到商品后的规定时间内把货款江给公司。

流程图描述了该公司发货、收款、催款的处理过程。其中部分文件和单据的格式如下。

商品文件：商品代号、商品名称、单价

销售点文件：销售点代号、销售点名称、地址

发货单：发货日期、销售代号、商品代号、数量、金额

收款单：收款日期、销售代号、商品代号、数量、金额、该商品的发货日期

处理 1～3 把当天的发货单合并到发货文件。处理 4～6 把当天的收款单合并到收款文件。处理 7 在发货文件中当天已收款的记录确良加上已收款标记。处理 8 和处得 9 在月末执行一次，主要用于输出月发货报告、催款通知单、月收款报告。

［问题 1］详细写出流程图中商品文件和销售点文件对处理 1 的作用。

［问题 2］说明处理 8 为何要写发货文件。

［问题 3］说明处理 9 除制作月收款报告外还对收款文件做什么操作。

［问题 4］为了提高处理效率，流程图需作何改动。

【解析】

本题的流程图描述了某公司发货、收款和催款的处理过程。通过阅读题目所给出的流程图可以了解到下面几点内容：流程图中描述的一方面是对发货单的日常处理，另外一方面是对收款单的日常处理，两者的交叉点是每天在发货文件中当天已收款的记录上加上已收款标志，还有每月对发货文件和收款文件的报表打印和催款通知单的打印。

一般来说，在对数据输入和处理的过程中都要进行数据的合法性和一致性的检查。本题的流程图中处理 1 和处理 4 都是数据输入处理，这一过程必然要进行数据合法性的检验。在这里

合法性的检查主要有在将发货单合并成日发货文件的时候，要检查发货单中数据的正确性，即发货单中商品代号、销售代号、数量以及金额的正确性。那么这里是如何进行合法性检查的？考查商品文件、销售点文件及发货单的数据项，不难推断，对销售点代码的检查可以通过在销售点代码录入有误。通过商品文件来检查商品代码的合法性原理与相应的记录，说明发货单中数量和金额两个数据项的检查也可以通过商品文件来实现，如果金额/数量≠单价，则说明数量或金额的录入有误。这里的前提是商品文件和销售文件是准确无误的。

处理 8 的内容是根据发货文件、商品文件和销售点文件来生成月发货报告、催款单文件。究竟为什么要写发货文件，这要通过此前此后发货文件的使用来解答。很明显，发货文件在参与处理 8 之前参与了处理 7，处理 7 在发货文件中当天已收款的记录加上了已收款标记。这样，在处理 8 中的发货文件记录可以分成两个集合，一是已收款的发货记录，一是未收款的发货记录。未收款的发货记录是生成催款单通知的根据，已收款的发货记录，一是未收款的发货记录。未收款的发货记录是生成催款单通知的根据，已收款的发货记录是生成月发货报告的根据。经过处理 8 之后的已收款发货记录是否还有保留的必要？如果保留，必须影响此后的处理（处理 3、处理 7 和处理 8）效率，因为已经有了月发货报告，从存留档案资料的角度看，在系统中保留这些记录的意义已不存在，所以处理 8 中对已收款发货记录进行删除。

处理 9 则是针对收款文件做出当月的收款报告。因为收款文件中是记录当月的收款单，而且在流程图中没有考虑预付款的处理，所以任何收款记录都没有保留到下一个月的必要。据此，为消除冗余数据、提高系统效率，在月末做完报表后，可将收款文件全部清空。

流程图/程序效率的提高主要靠处理的数据对象的有序性和尽可能小的数据冗余来实现。结合以上分析查流程图，不难发现，处理 7 每天都要执行。给当天已收款的记录加已收款标志，收款文件的有序性是有保证的，但其中包含许多此前已经处理的记录，每天都要参与一次处理 7 的处理，显然是一种极大的系统源浪费。如果交处理 7 的数据来源改为日收款分类文件，则系统对数据有序性和冗余度（无数据冗余）的要求都能够得到满足。

【答案】

［问题 1］检查发货单上的非法销售代号、非法商品代号、数量×单价≠金额等错误。

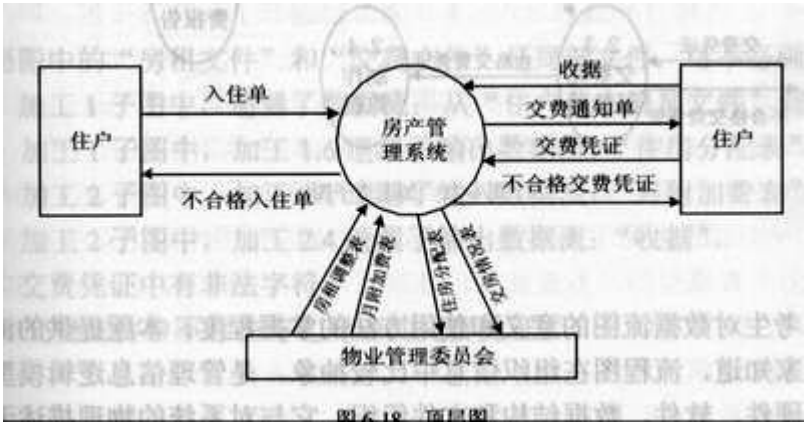
［问题 2］删除发货文件中已做收款标记的所有记录。

［问题 3］删除收款文件中的所有记录。

［问题 4］从收款文件到处理 7 的连线改为从日收款分类文件处理 7 的连线。

试题 3 （1997 年试题 3）

阅读以下说明和流程图（如图 6-18 至图 6-21 所示），回答问题 1 至问题 3，将解答写在答卷的对应栏内。



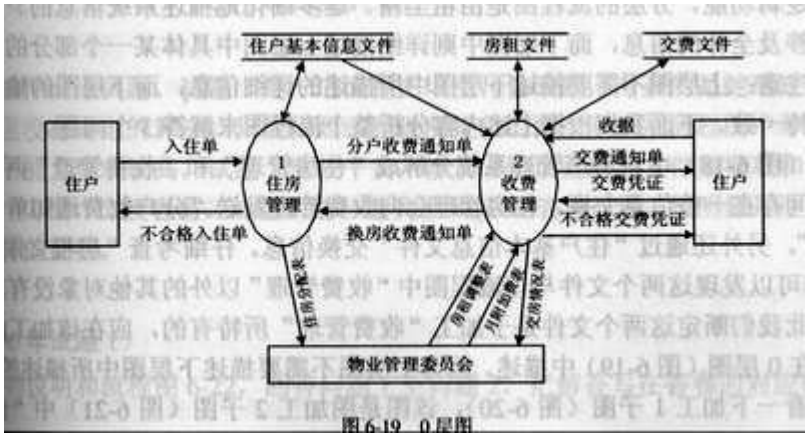


图 6-19　0 层图

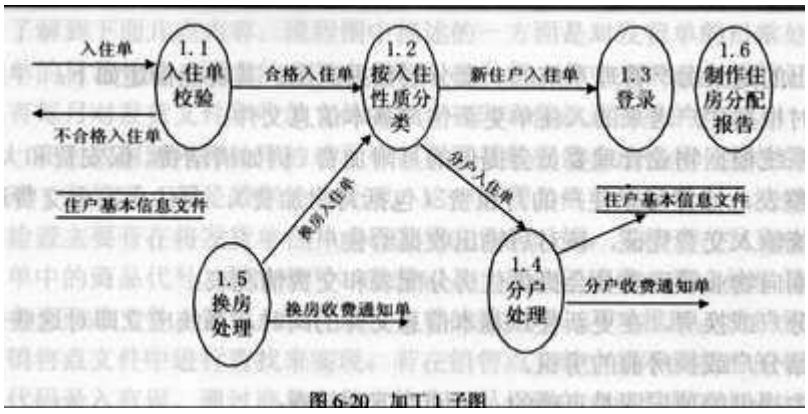


图 6-20　加工 1 子图

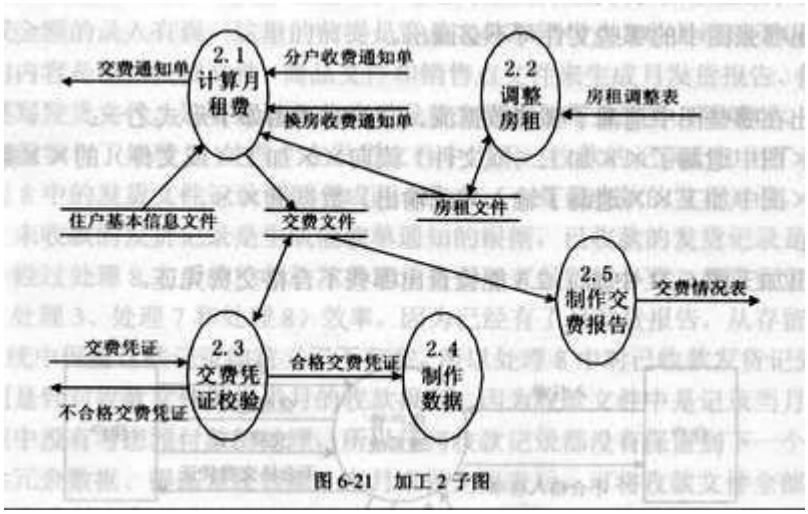


图 6-21　加工 2 子图

［说明］

下面给出的是某房产管理系统的一套分层数据流图。其功能描述如下：

系统随时根据住户送来的入住单新住户基本信息文件。

每月初系统根据物业管理委员会提供的月附加费（例如清洁费、保安费和大楼管理费等）表和房租调整表，计算每家住户的月租费（包括月附加费），向住户发出交费通知单，住户交费时，系统输入交费凭证，核对后输出收据给住户：

系统定期向物业管理委员会提供住房分配表和交费情况表；

住户因分户或换房，在更新住户基本信息文件的同时，系统应立即对这些住户做月租费计算，以了结分户或换房前的房租。

假定题中提供的顶层图是正确的，请回答下列问题：

［问题 1］
指出哪张图中的哪些文件可不必画出。
［问题 2］
指出在哪些图中遗漏了哪些数据流。回答时请用如下形式之一：
XX 图中遗漏了 XX 加工（或文件）流向 XX 加工（或文件）的 XX 数据流；
XX 加工 XX 遗漏了输入（或输出）数据流 XX；
［问题３］
指出加工图６－２１中加工２．３能检查出中些不合格交费凭证。
【解析】
本题是考查考生对数据流图的意义和使用方法的掌握程度。本题提供的流程图是一种分层的流程图。大家知道，流程图在组织信息中比较抽象，是管理信息逻辑械型的主要形式，这人模型不涉及硬件、软件、数据结构和文件组织，它与对系统的物理描述无关，只是表不一种系统的逻辑功能。分层的流程图是由粗至精、逐步细化地描述系统信息的。上层图中描述的是粗略涉及全体的信息，而下层图中测详细描述上层图中具体某一个部分的内容。另外我们一定要注意：上层图不需要描述下层图中所描述的详细信息，而下层图的输入与输出应与上层图保持一致。下面我们根据上述内容分析整个流程图来解答 3 个问题。
顶层图（图 6-18）中将房产管理系统分解成"住房管理"和""收费管理两个子加工，两个部分之间存在一些信息交换。住房管理会向收费管理发送"分户收费通知单"和"换房收费通知单"，另外还通过"住户基本信息文件"交换信息。仔细考查"房租文件"和交费文件"，我们可以发现这两个文件与该流程图中"收费管理"以外的其他对象没有其他任何信息交流，由此我们断定两个文件是子加工"收费管理"所特有的，应在该加工的子图中描述，而不应在 0 层图（图 6-19）中描述。因为上层图不需要描述下层图中所描述的详细信息。
我们再看一下加工 1 子图（图 6-20），该图是图加工 2 子图（图 6-21）中"住房管理"的具体细化，它分为 6 个子加工，即入住单校验、按入住性质分类、登录、分户处理、换房处理和作住房分配报告。从流程图中可看出，"入住单校验"没有借助任何外部数据，不借助外部数据的校验是无法想像的，除非入住单数据项上有某些固定的对应关系，但似乎是不可能的。首先考虑入住单中会有什么错误，不合格的入住单无非就是对住房的重复分配。"住房管理"加工中所有的外部数据源，应根据"住户基本信息文件"来校验入住单，所以加工 1.1 缺少由"住户基本信息文件"流向它的数据流。根据 0 层图（图 6-19）中加工的输入输出流来检查加工 1 子图（图 6-20），与 0 层（6-19）中的输入流和输出流相比较，加工 1 子图（图 6-20）中缺少了"住房分配表，"这一输出流，分析加工 1.1 至加工 1.6，可知制作住房分配报告由加工 1.6 负责，因此断定加工 1.6 缺少了输出数据流"住房分配表"。
接下来分析加工 2 子图。该图是顶层图"收费管理"的具体细化，该加工分为 5 个子加工，即月租费、调整房租、缴费凭证校验、制作收据和作交费报告，并且附加了两个文件，即文件和交费文件。同样我们也看一下 0 层图（图 6-19）的输入输出数据流：输入流"分户收费通知单"、"换房收费通知单"、"交费凭证"、"房租调整表"都存在，但少了一个"月附加费表"，这个表是在计算月租费的时候使用，所以加工 2.1 缺少了输入数据流"月附加费表"。输出流"交费情况表"、"不合格交费凭证"、"交费通知单"都存在，但缺少了"收据"这一项，而"收据"应由加工 2.4 产生，因此加工 2.4 缺少了输出数据流"收据"。
在子加工 2.3 中可以检查出的不合格交费凭证是：①是凭证的语法错误，也就是输入信息不合规范；②是输入的数据虽然合乎规范，但是内容不对（参照原解答修改此处）。
【答案】
［问题 1］0 层图中的"房租文件"和"交费文件"是局部文件，可不必画出。
［问题 2］①加工 1 子图中，遗漏了数据流：从"住户基本信息文件"到加工.1.1。

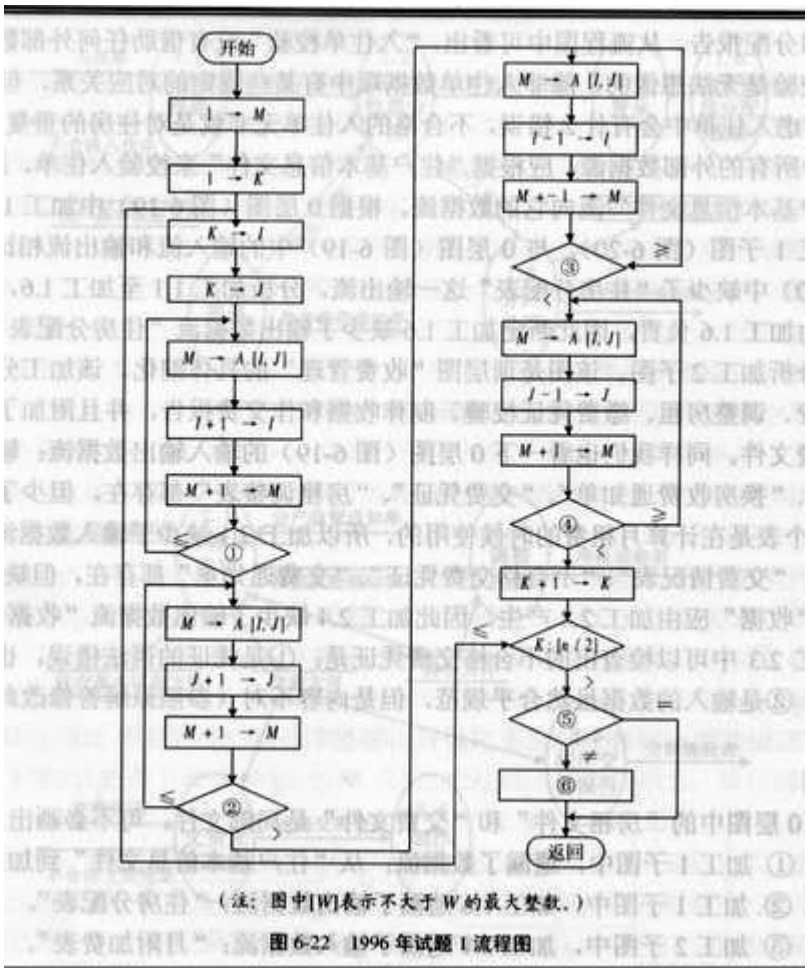
②加工 1 子图中，加工 1.6 遗漏子输出数据流："住房分配表"。
③加工 2 子图中，加工 2.1 遗漏了输入数据流："月附加费表"。
④加工 2 子图中，加工 2.4 遗漏了输出数据流："收据"。
［问题 3］①交费凭证中有非法字符。
②交费文件中不存在与之对应的交费凭证。

5.5 1996 年度软件设计试题解析

共 3 道软件设计试题。试题 1 是一道流程图填空修改题，考查的是蛇形矩陈的生成，很明显，这里没有固定的算法，只能依照试题说明提供的两个示例从流程图中寻找算法的实现，对考生的能力要求较高。试题 2 比较平实，涉及了错误查找、文件数据项和提高处理速度的途径 3 个方面，比较容易。试题 3 是一道关于 SQL 语言的度题，很明显是注重考查考生对流程图总体的把握。

试题 1 （1996 年试题 1）

阅读下列说明和流程图 6-22，回答问题 1 至问题 2，把解答写在答卷的对应栏内。



（注：图中[W]表示不大于 W 的最大整数。）

图 6-22　1996 年试题 1 流程图

［说明］
本流程图将数字 1，2……，N（N≥2）接逆时针方向依次写在 N×N 矩阵中。下图给出了 N=4 和 N=5 的情况：

1　12　11　10　　1　16　15　14　13

2	13	16	9
3	14	15	8
4	5	6	7

（N=4 时）

2	17	24	23	12
3	18	25	22	11
4	19	20	21	10
5	6	7	8	9

（N=5 时）

［问题 1］

填充流程图中的①～⑥，使之成为完整的流程图。

［问题 2］

若将数字 1，2……，N2 按顺时针方向依次写在 N×N 矩阵中，则只需将上述流程图中的⑦改成⑧即可。

【解析】

粗读流程图可知，程序开始先初始化 M 和 K 为 1，然后就进入一个大循环。在这个大循环中，包括 4 个循环①、②、③、④。在循环①中 J 值不变，I 值每循环一次加 1，即列号不变，行号递增，是从上往下将值写入矩阵中；在循环③中 J 值不变，J 值每循环一次减 1，即列号不变，行号递减，是从下往上将值写入矩阵中；在循环④中 J 值不变，I 值每循环一次减 1，即行号不变，列号递减，是从右往左写入矩阵中；4 个小循环结束以后，K 值加 1。

由于本题的功能是将自然数按逆时针方向依次写在 N×N 矩阵的一圈，而外层循环的功能则是控制里面循环的次数，即总共写几次，也即 N×N 矩阵总共有几圈，所以 K 的功能的功能则是控制我层循环的次数，同时也控制写入第几圈及开始写入的位置。

而 M 的功能则相对简单，M 在每个小循环中都是递增，且其值被赋给矩阵的元素，可知 M 为填入矩阵的值，它对循环没有什么影响。

在任一圈上依次按 4 条边写入矩阵时，每一边的起点都在对角线上，终点都不在对角线上，每一边要写入的元素的个数都相同，当前的圈数为 K，则每一边所应写入矩阵的元素的个数均为 N-24K+1，即写入每一边的小循环次数应为 N-2K+1。

再仔细阅续每个小循环的流程图。循环①的功能是从上往下写数，那么其结束的标志应该是已经把这一列写完了，该边行的起始位置为 K，所以结束条件就为 I：N-K，这样总的循环次数为 N-2K+1。

循环②的功能是从左右写数，那么其结束的标志应该是已经把这一行写完了，该边列的起始位置为 K，所结束条件就为 J：N-K，这样总的循环次数为 N-2K+1。

循环③的功能是从下往上写数，那么其结束的标志应该是已经把一行写完了，该边行的起始位置为 N-K+1，所以结束条件就为 I：K+1，这样总的循环次数为 N-2K+1。

循环④的功能是从右往左写数，那么其结束的标志应该是已经把这一行写完了，该边行的起始位置为 N-K+1，所以结束条件就为 J：K+1，这样总的循环次数为 N-2K+1。

外循环每循环一次，K 值加 1，若 K 大于 N/2，此时已达到矩阵中心，则外层循环结束；否则尚未达到矩阵中心，则从 A［K，K］开始重复上述操作。若已达到矩阵中心，又有两种情况：若 N 为偶数，则所有的元素均填写完毕，程序结束；若 N 为奇数，则矩阵中心元素所以⑤的答案为 MOD（J，2）：0 或其他任何能判断出 N 的奇偶性的表达式。

观察矩阵可知，只要将按逆时针方向写入得的到的矩阵置，即将矩阵的行列互换，就得到将字 1，2……， N2 按顺量针方向写入得到的矩阵。在流程图中，I 中代表行，J 代表列，将 J 与 I 互换即可达到行到互换的目的，所以只要将 A［I，J］改成 A［I，J］即可得到按顺时针方向写成的 N×N 矩阵。

【答案】

［问题 1］

①I:N-K 或 I:N-J 或 M:4(K-1)(N-K+1)+N-2K+1

②J:N-K 或 J:I-1 或 M:4(K-1)(N-K+1)+2(N-2K+1)
③I:K+1 或 I:N+2-J 或 4(K-1)(N-K+1)+3(N-2K+1):M
④J:K+1 或 J:I+1 或 4K(N-K):M
⑤MOD(N,2):0 或 [N/2]*2:N 或 [N/2]:[N+1/2] 或 N/2: [N/2] 或 [N/2]:[N/2] 或 N:2(K-1) 或 I+N=N 或 M-1:N^2
⑥→A[K,K]
［问题 2］
⑦A[I,J] 或 M→A[I,J]
⑧A[J,I] 或 M→A[J,I]

试题 2（1996 年试题 2）

阅读下列说明和流程图 6-23，回答问题 1 至问题 3，把解答写在答卷的对应栏内。

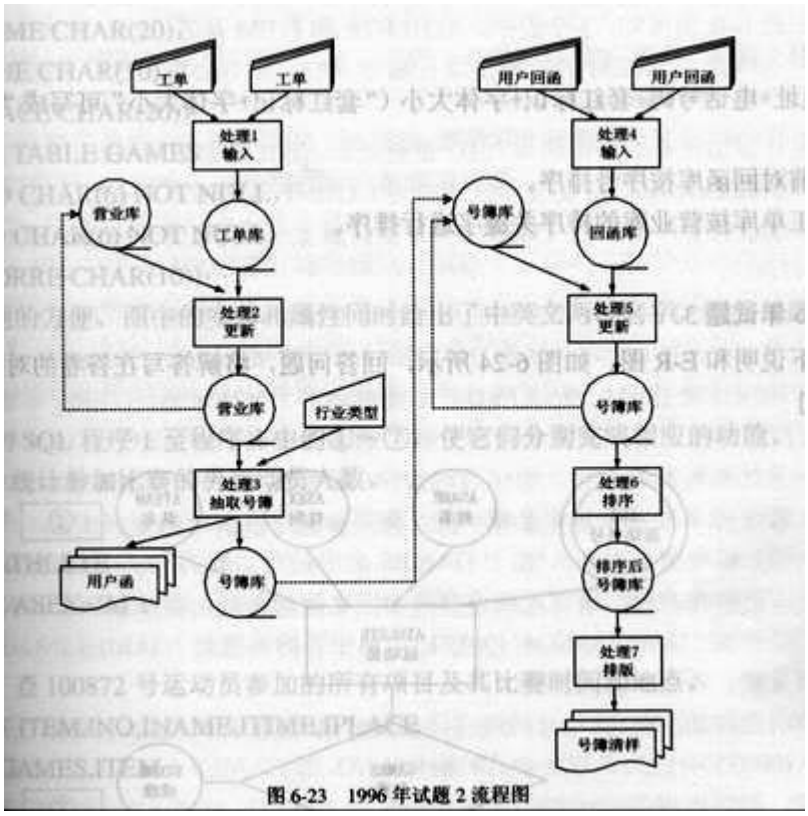


图 6-23　1996 年试题 2 流程图

［说明］
本流程图描述了某行业分类电话号码薄（简称号薄）出版系统的处理流程。全市所有电话的基本信息均存入在营业库中。系统输入工单，工单中包括电话的新装、拆除、移机和更改（更改户名、地址和电话号码等）信息。为确保输入工单的正确性，每张工单均由两个录入员分别录入，由处理 1 进行输入和校对，然后理更新营业库。系统根据特出版号薄的行业类型从营业库中选取该类用户的电话信息，存入在号薄库中。 同时向每个电话用户发出用户函，用户函上记录着半刊登在号薄库中的位置。用户收到用户函后，进行校对，并将修改内容和印刷要求（字体大小和是否套红）填写在用户回函中，系统按收到用户回函的先后顺序依次输入用户回函，然后更新号薄库，最后通过排版输出经用户校对并符合其印刷要求的号薄清样。
系统中部分单据和文件的格式如下：
工单=工单类型=户名+地址+电话号码

营业库记录=户名+地址+电话号码+分类信息
用户函=序号+户名+地址+电话号码
用户回函=序号+户名+地址+电话号码+套红标记+字体大小

［问题 1］
流程图中哪些处理能发现工单的哪些错误，并举例说明。
［问题 2］
指出号薄库文件的记录至少应包括哪些据项。
［问题 3］
为提高和理速度，流程图需作何改进。
【解析】
在哪些处理中能发现哪些错误,也就意味着要在哪些处理中检查哪些错误。一般来说，在数据进入系统后需要马上进行数据合法性检查，及时排除错误数据，保证以后的处理有正确的数据来源。考查流程图，可以看出处理 1 和处理 4 完成数据录入员录入工单，并对录入的数据进行核对的。此时可以发现两人录入的不同之和和录入的非法数据,核对后的数据放入工单库中。处理 4 由两个录入员入用户回函信息，核对后放入回函库，这一步也可以发现两个录入员输入不同信息和输入非法数据。但是我们不要忽视处理 2 处理 2 根据营业库中已有的用户信息对照工单营业库进行更新。这一步可以发现工单库与营业库数据不一致之处，例如工单库中有一人顾客要安装电话，而营业库中却发现该电话码已存在，即一个电话号码的重复分配。
为考查号簿库包含的数据项，我们需要考查号簿库的数据来源的的去向。号簿的数据来源是营业库，而营业库的数据项是由工单而来，但因为号簿库与工单已经没有直接联系，所以能够断定营业库中的数据项是可以包含在号簿库数据项中的。是否需要全产包含，尚不能断定。用户函是由号簿库而来的，我们推敲用户函在以后处理中的使用情况，可以推知在用户函中每个用户可以选自己的信息印刷格式。如字体大小和套红标记，所以号簿文件中还应含有印刷要求，如字体大小和套红标记。现在再考虑号簿库中是否需要分类信息？显然是不需要的，因为号簿库在生成时已经是按分类存放了（参见处理 3），所以分类信息可以从号簿库中排除出去。
从软件上提高数据库处理速度最常用性的方法是保证数据的有序性，因此我们可以在处理 5 前对回函库按序号排序，这样可以提高处理 5 的更新速度，节省处理时间。此外，还可以在处理 2 前对我单库按营业库的排序关键字进行排序，这样可以加快营业库更新的速度，节省更新时间。
【答案】
［问题 1］
处理 1 能发现两个人录入不全同的工单和非法数据（如非法字符和数据越界等）。
处理 2 能发现与营业库不一致的工单（如新装电话重号、移机、拆机、更改和原电话号码不存在等）
处理 4 能发现两人录入不全同的用户回函和非法数据（如非法字符和电话号码长度不对等）。
［问题 2］
户名+地址+电话号码+套红标识+字体大小（"套红标记+字体大小+可写成"印刷要求"）。
［问题 3］
处理 5 前对回函库按序号排序。
处理 2 前对工单库按营业库的排序关键字进行排序。

试题 3 （1996 年试题 3）

阅读以下说明和 E-R 图，如图 6-24 所示，回答问题，将解答写答卷的对应栏内。

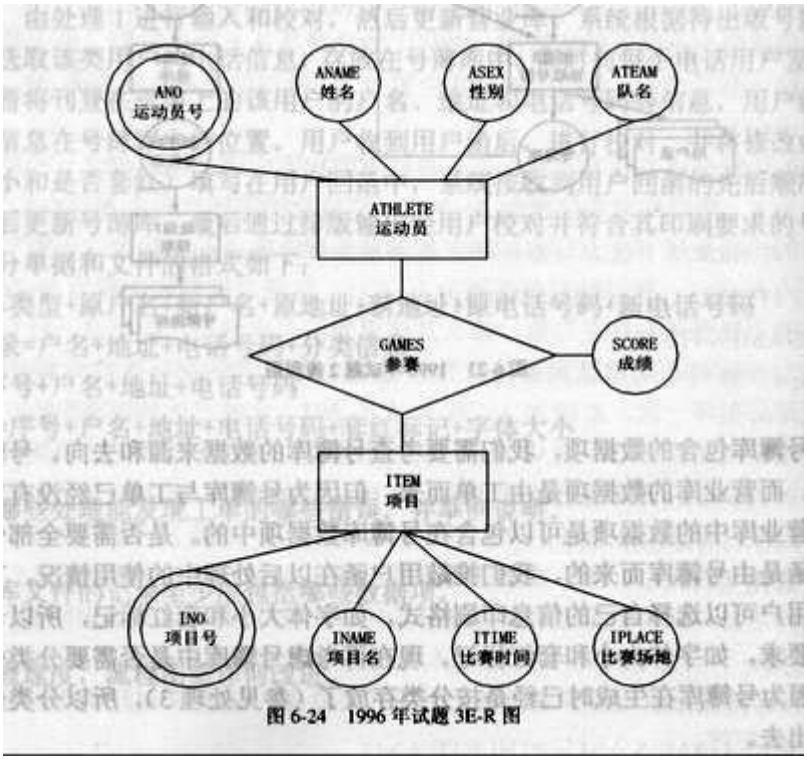


图 6-24　1996 年试题 3E-R 图

［说明］

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

```
CREATETABLEATHLETE
(ANOCHAR(6)NOT NULL,。
ANAMECHAR(20)
ASEXCHAR(1),
ATEAMCHAR(20);
CREATETABLEITEM
(INOCHAR(6)NOTNULL,
INAMECHAR(20),
IPLACECHAR(20);
CREATETABLEGAMES
(ANO CHAR(6)NOTNULL,
INOCHAR(6)NOTNULL,
SCORRECHAR（10）；
```

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

［问题］

填充下列 SQL 程序 1 至程序 4 中①～⑦，使它们分别完成相应的功能。

程序 1：统计参加比赛的男运动员人数。

```
SELECT ①
FROM ATHLETE
WJEREASEX="M"
```

程序 2：查 100872 号运动员参加的所有项目及其比赛时间和地点。
SELECTIEM.INO,INAME,ITIME,IPLACE
WHERE ②
AND ③
程序 3:查参加 100035 项目所有运动员名单。
SELECTANO，ANAME，ATEAM
FROM ATHLETE
WHERE ④
SELECT ⑤
WHEREGAMES.ANO=ATHLETE.ANO
ANDINO=100035;
程序 4：建立运动员成绩视图
⑥ ATHLETE-SCRE
ASSELECTATHLETE.ANO,ANAME,ATEAM,INAME,SCORE
FROM ⑦
WHERE ATHLET.ANO=GAMES.ANO
ANDGAMES.INO=ITEM.INO；
【解析】
本题是关于系数据库标准语言--SQL（Structured Query Language）语言的题目，由题目中经给出的 E-R 图可知，3 个表中，ATHLETE 和 ITEM 是基本表，表 ATHLETE 的主键是运动员编号 ANO，表 ITEM 的主键是项目编号 INO，表 GAMES 是一个视图，以 ANO、INO 为外键。
程序 1 统计参加比赛的男运动员人数，也就是表 ATHLETE 中，AEX='M'的记录的个数，所以要用到库函数 COUNT（*）。这里注意的是 COUNT 与 COUNT（*）的区别，COUNT 的功能是对一列中的值计算个数，而 COUNT（*）。这里要注意的确良 COUNT 与 COUNT（*）区别，COUNT 的功能是对一列中的值计算个数，而 COUNT（*）才是计算数据库中记录的个数。所以填空①的答案为"COUNT（*）"。
程序的 2 统计 100872 号运动员参加的所有项目及比赛时间和地点，所以 SELECT 后面的内容是项目编号 ITEM.INO、项目名称 INAME 时间 ITIME 及地点 IPLACE。统计涉及到比赛表 GAMES 和项目表 ITEM，所以 FROM 后面的内容为 GAMES、ITEM。本题考的是连接查询，所谓连接查指的是涉及两个以上的表的查询。由于是统计 100872 号运动员参加的所有项目及比赛时间和地点，所以查询条件中必然有 GAMES.INO='100872'（程序中引用到字段时，若字段名在各个表中是唯的，则可以把字段名前的表名去掉，否则，应当加上表名作为前缀，以免引起混淆）。由于 GAMES 表中只有比赛的成绩，那些关于项目的数据必须从项目表 ITEM 中取得，所以还应该有两个表之间的关联，即 GAMES.INO=ITEM.INO。所以填空②和③可交换，不影响查询结果。
程序 3 要求查参加 100035 项目的所有运动员名单。分析查询表达式，必首先查 GAMES 表，找出参加 100035 项目的那些运动员的编号 ANO,即 GAMES.ANO=ATHLETE.ANO AND INO='100035',然后再根据查询到的运动员的信号 ANO 从 ATHLETE 表中取妯运动员的数据。所以填空④的答案为"EXISTS"或"ANOIN"，填空 ⑤的答案为"ANO"
程序 4 要求建立运动员成绩视图。建立视图的命令为 CREATEVIEW，所以填空⑥的答案一定是"CREATEVIEW"。建立的是运动员成绩视图，那么一定涉及运动员情况、运动员参加的项情况和该项目的成绩，所以要用到 ATHLETE、ITEM 和 GAMES 这 3 个表，因此 FROM 子句后为 ATHLETE、GAMES、ITEM，3 个表可以是任意次序，不影响结果。

【答案】
①COUNT（*）（若答 COUNT 或 COUNT*得 2 分）
②GAMES.INO=ITEM.INO
③GAMES.ANO='100872'（注：②、③可互换、无前缀得 1 分）
④EXISTS
⑤*或 ANO 或 INO 或 SCORE 或后 3 个列名的任意组合
⑥CREATEVIEW
⑦ATHLETE， ITEM， AMES（3 项可交换。）
注：④、⑤也可为
④ANOIN
⑤ANO

5.6 1995 年度软件设计试题解析

1995 年共有 5 道软件设计试题供选答。试题 1 总体说难度不大，问题 1 和问题 2 的形式是在广大考生的意料之中的，问题 3 涉及对流程图的评估，只要能够仔细理解系统处理流程，应该是比较容易的。流程图填空（试题 2、试题 3）分别涉及用枚举法录找符合要求的自然数与对紧缩存储的一元多项式相乘两个方面的问题，而且都涉及了对流程图进行修改的问题。应该说，这两道试题的考查重点是能力考查，如果不能迅速理解试题说明所提供的算法和流程图的结构，解答问题的难度将非常大。试题 4 采用状态转换矩阵来检验非空算术表达式的语法正确性，解题关键是要明确各个状态含义，这在客观上增大了答题的工作量，而且解题所需信息只能从试题说明中获得，这是一道纯碎的能力考查试题。试题 5 是一道关于软件测试的题目，很明显，这部分知识的概念是容易记忆的，但在知识的具体运有上却并非易事。本题是一道知识与能力考查并重的试题，能否成功解答本题关键是看知识掌握程度和解决的能力。

试题 1（1995 年试题 1）

阅读下列说明和流程图（如图 6-26 所示），回答问题 1 至问题 3，把解答写在答卷的对应栏内。

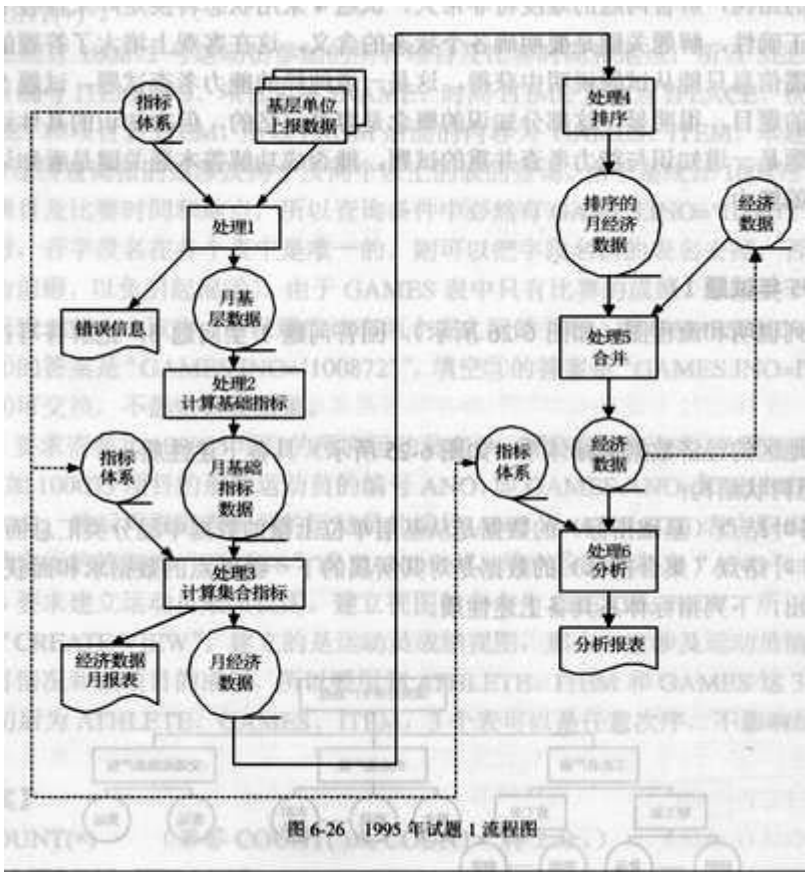
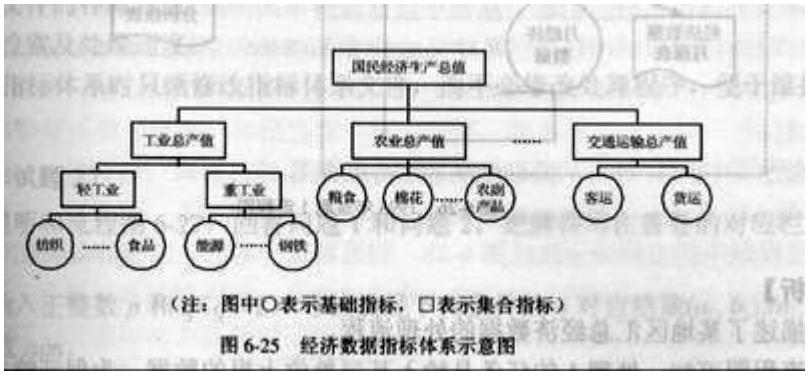


图 6-26　1995 年试题 1 流程图

［说明］

假定某地区的经济数据指标体系（如图 6-25 所示）具备下述性质：



（注：图中○表示基础指标，□表示集合指标）

图 6-25　经济数据指标体系示意图

（1）呈树状结构：

（2）各叶结点（基础指标）的数据是从基层单位上报的数据中经分类汇总而获得；

（3）非叶结点（集合指标）的数据是对其所属的下一级结点的灵敏据求和而获得的。不难看出，下列指标体系具备上述性质。

例如：粮食、棉花……和农副产品产值（基础指标）可以从各县及各农场每月上报的数据中经分类汇总获得，农业总产值（集合指标）=粮食产值+棉花产值+……农副产品产值。

本流程图用来计算月基础指标数据和月集合指标数据，产生经济数据月报表，并把月经济数据加载到经济数据文件中，产生分析报表。

假定有关的文件和单据的记录格式如下：

指标体系文件：指标代码、指标名称、计量单位

基层单位上报数据：单位名称、指标名称、产值

［问题 1］

简述处理 1 的处理内容。

［问题 2］

指出月基础指标数据文件的记录格式至少应包含哪些内容。

［问题 3］

简要叙述指标体系文件中的指标代码的主要作用。

【解析】

本题描述了某地区汇总经济数据的处理流程。

阅读流程图可知，处理 1 的任务是输入基层单位上报的数据。为保证输入的数据的合法性，处理 1 还应对输入数据进行合法性检查。同时我们看到，指标体系文件参与了处理 1，这样将基层单位上报数据中的"指标名称"改为"指标代码"就成为可能，而且这样也可以为减少文件的存储量，使处理 2 和处理 3 能够更方便的汇总。

考查月基层数据的数据来源，它是由指标体系文件和基层单位上报数据经过处理 1 而成的，这样，我们就可以确定月基层数据的数据必须直接或间接从指标体系文件和基层单位上报数据中取得。而月基础指标数据经处理 2 生成的，从流程图来看，月基础指标数据的数据项必须全部直接从月基层数据取得，这样我们就可以确定月基础指标数据文件数据项的最大范围，即指标代码、指标名称、计量单位、单位名称和产值等 5 个数据项。考查上述 5 个数据项，显然，指标代码和指标名称的并存是一种存储浪费，考虑到文件长度和处理的简便性，我们保留指标代码；另外，由于流程图是用于地区的经济数据汇总，汇总后的数据是指整个地区的各指标的总产值，因此单位名称将不会参与以后的处理，可以删除。最后，基层单位上报的数据中只有产值，没有计量单位，由此我们可以推断对同一个指标各单位产值的计量单位都有是相同的，所以计量单位不必保存在月基层数据文件中，至于在输出经济数据月服表和分析报表时需要用到计量单位，则可以从指标体系文件中取得。

根据上述分析，月基础指标数据文件的记录只有指标代码和产值两个数据项。但要考虑到，基础指标数据文件中，对应于一个基指标码的记录只有一个，它存储汇总后的该基础指标的总产值。

指标代码的主要作用集中表现在以下 3 个方面：(1) 指标码可用来代替指标名称，以减少文件的存储容量；(2) 只要科学地设计指标代码，不但能描述各指标体系中的相互关系，而且由于指标代码明显比指标名称简单，便于处理，能够提高检索及处理速度；(2) 当指标体系改变时，只须改指标体系文件，而不必须改程序，使程序与数据的相关性大大降低，提高了软件的可维护性。

【答案】

［问题 1］

(1) 输入基层单位于报的数据，并进行合法性检查。

(2) 用指标代码替换指标名称，形成月基层数据文件。

［问题 2］

指标代码，产值。

［问题 3］

(1) 减少文件的存储容量；

(2) 提高检索及处理速度；

(3) 修改指标体系时只须修改指标体系文件，而不必修改理程序，便于维护。

试题 2（1995 试题 2）

阅读下列说明和流程图 6-27，回答问题 2，把解答写在答卷的对应栏内。

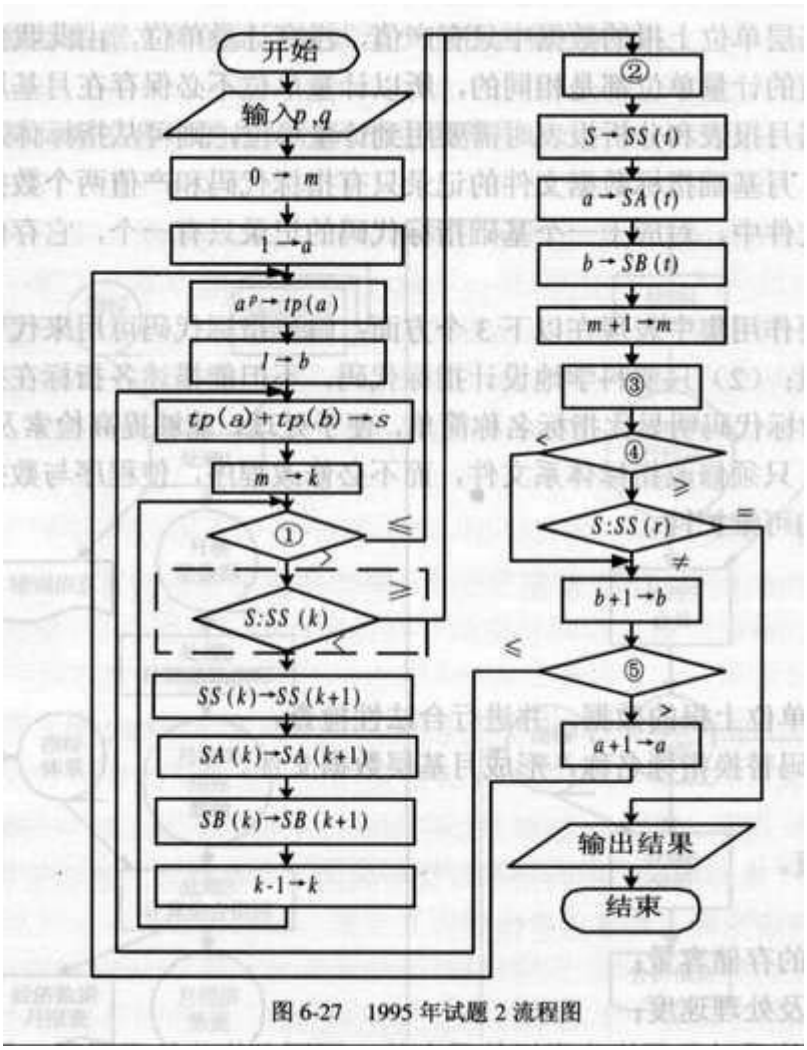


图 6-27　1995 年试题 2 流程图

［说明］

本流程图输入正整数 p 和 q（q≥2），寻找满足下列车员条件的 q 对自然数(a1b1),(a2b2),……(aq,bq)及最小的整数 sum。

1.ai ≥bI(I=1,2……q)

2.当 i≠j 时（aj,bj）(I=1,2…q)

3. sum=$a1^p+b1^p=a2^pb2^p=…aq^p+bq^p$

例如：

当 p=2，q=2 时，sum=$50=7^2+1^2=5^2+5^2$；

当 p=3, q=2 时，sum1729=$12^3+1^3=10^3+9^3$;

当 p=3, q =3 时，sum=8753919=$426^3+167^3=423^3+228^3=414^3+225^3$。

本流程图采用枚举法，列举各种 aj^p、bj^p(a≥bi)及其和 sum=aj^p+bj^p，当发现 q 个相同的和时，即输出结果。图中，数组元素 tp(K)=K^p(k=1,2…)，枚举过程中产生的 sum 按递增顺序存放在数组 SS 中，相应的 ai 和 bI 存放在数组 SA 和 SB 中。

［问题 1］

填充流程图中的①～⑤，使之成为完整的流程图。

［问题 2］

若将流程图中的虚框部分改成图 6 -28,则流程图中的③、④应作怎样的修改。

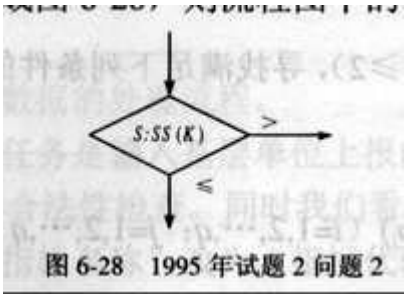


图 6-28　1995 年试题 2 问题 2

【解析】

该题用来寻找满足下列条件的 q 对自然数(a1,b1),(aq,bq)以及最小整数 sum：

1．ai＞= bI (i=1，2…，q)

2．当 I＜＞j 时，(aibi) (I=1,2,…q)

3. sum=$a1^p+b1^p=ap^q+bp^q$

仔细研究试题说明和流程图，我们可以确定流程图中各语句框的作用，并把流程图分解为 3 层循环；结合试题说明，仔细体会 3 重循环的功能，这里的关键是如何判断流程图中语句与试题说明的对应情况。

我们的思路沿流程图走，当走到填空①处时，不外乎有两种可能，即沿填空①的两个判断分支进行。很明显，如果我们沿"大于"分支前时，当执行 K-1→K 时将会出现数组越界问题，因为此前 K 的初值为 0。所以此处的判断该处寻找 S 在数组 SS 中合适位置的操作，则可知在两种情况下完成该操作，即 S 大于等于 SS［K］或 K 小于等于 0 时,这样,填空①的答案就非常明显了。那么填空②呢？②显然应该先给 t 赋初值，然后再在程序中使用，关键是赋什么初值，从上面的分析我们不难看出，S 应该插入 SS［K］的后面，这样我们就在②处填写 K+1→t。考查上述两个解答，它们能够完成在第一次循环时把处理导向②处的功能，应该是正确的解答。

我们且不论填空③和④及以后的语句是什么意思，有一点是可以肯定的，即在③和④中应该完成对 r 赋初值的操作，显然④是不能够胜任的,这样,我们应该在③中对 r 赋初值。流程图的剩余部分就是判别是否已有 q 个这样的 S （即 sum ）值。由于数组 SS 已按递增次序排列，所以具有相同的 S 值的元素一定在数组 SS 的连续存储区中。因为图中当 S＞=SS［K］时即将 S 插入，所以 S 的插入点 t 一定是该连续存储区的最后一个单元。如果存在 q 在相同的 S 值，则该连续存储区的第一个单元应在 t-q+1 处。所 r 的初值应该是 t-q+1，又由于在填空②中我们执行了 K+1→K 的操作，所以这里也可以赋初值 k-q+2。计算出 r 的值之后，显然，r 的值应该在区间［1，m］内。如果 SS［r］=SS，则认为找到一个解答，所以填空④的解答应该是 r:1。

结合试题说明中 aI 大于等于 bI 的提示，我们把结束中层循环的条件设置为 b:a，以便在 b 大于 a 时退出中层循环。

如果将图中虚线部分的判断框的出口条件改成=＜和＞，则仅当 S ＞SS（K）时才插入 S。此时 S 插在上述连续存区的第一个单元处。如果存在 q 个相同的 S，则该连续存储区的最后一个单元在 t+q-1 处。因此，应将填空③改成 t+q-1→r。由于 q＞0，所以 t+q-1=r＞=1 必定成立，故填空④应改成 m:r。

【答案】

［问题 1］

①K：0

②K+1→或 K+2-q→r

③t-q+1→或 K+2 -q→r

④r:1

⑤b:a

［问题 2］

③t+q-1→或 K+q→r

④m:r

试题 3（1995 年试题 3）

阅读下列说明和流程图,如图 6-29 所示,回答问题 1 和问题 2,把解答写在答卷的对应栏内。

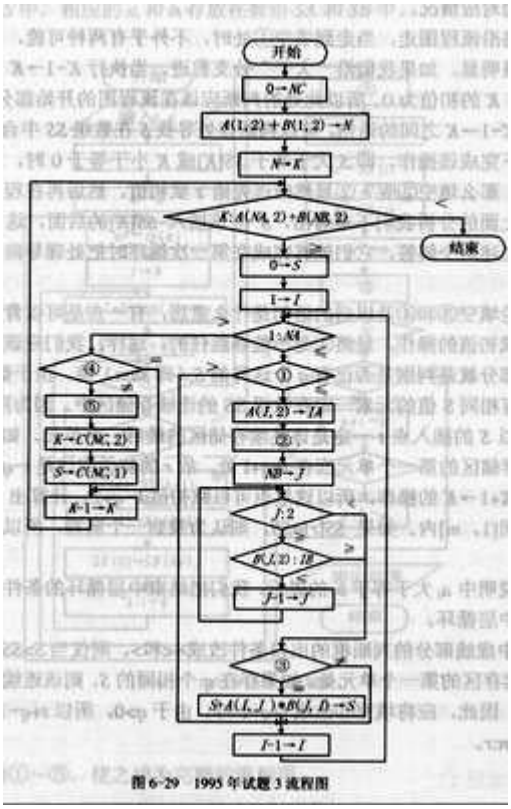


图 6-29　1995 年试题 3 流程图

［说明］

当一元多项式 $\sum_{i=1}^{n} a_i x^i$ 中有许多系数为零时，可用一个二维数组 D（M，2）来紧缩存储，其中 M 为多项式中非零系数的个数，数组的第一列存入非零系数的值，第二列存放该非零系数所对应的幂次，并且规定，数组元素按幂次的递减速次序存放。

例如，对应于多项式 $8X^5-2X^2+7$ 的二维数组内容如下所示：

$$\begin{pmatrix} 8 & 5 \\ -2 & 2 \\ 7 & 0 \end{pmatrix}$$

本流程图用来计算两个多项式的乘积，多项式的系数和幂次均按上述方式存放。数组 A、B 存放两个数欲相乘的多项式，它们的非零系数个数分别为 NA（＞0）和 NB（(＞0)），数组 A、B 存放两个数欲乘的多项式，它们的非零系数个数分别为 NA（＞0）和 NB（＞0），数组 C 存放结果（乘积）多项式，其非零系数个数用 ND 存储。

［问题 1］

填充流程图中的①～⑤，使之实现上述功能。

［问题 2］

若将流程图中 J：2 改成：J：1，则流程图能否正常工作，为什么？

【解析】

该题的流程图用来计算二个一元多项式的乘积。试题中建立一个二维数组，采用压缩存放的方式用来存放一元多项式，数组元索按幂次的递减次序存放，数组的第一列存放多项式中非零系数的值，第二列存放该非零系数所对应的幂次。题中还规定两个压缩存储的一元多项式 A 和 B 的乘积 C 也采用压缩存储。

分析流程图，该流程图由一个三重循环组成。外循环的循环变量为 K，初始值是 A（1，2）+B（1，2）。当 K 小于 A（NA，2）+B（NB，2）时外循环中 K 递减 1。由此我们推断，K 为 C 的某一个可能幂次。计算幂次为 K 的系数值，需要找出满足 A（1，2）+B（NB，2）＞K 时，满足 A（1，2）+B（J，2）=K 的 J 不可能存在，应该结束该循环，此时经 I+1→I 取下一个 A（I，2）。所以填空①应为 A（I，2）：K-B（NB，2）或 A（I，2）+B（NB，2）：K。当 A（I，2）≤K-B（NB，2）时先将 A（I，2）送给 IA，阅读流程图不难看出。内循环用来寻找值为 IB 的 B（J，2）。为使 A（I，2）+B（J，2）=K，IB 应为 K-A（I，2）。图中有语句 B（J，2）：IB，而此前没有对 IB 的赋值操作，所以填空②处应对 IB 赋值，至此，在填空②处填写 K-A（1，2）→IB 已经是非常明显的了。

由于内循环在 J＜或 B（1，2）≥IB 时结束，所以填空③应为 B（J，2）=IB 时才将 A（I，1）累加到 S 中。如此循环，直至 I＞NA，此时 S 中应该是对应于乘积的幂次为 K 的系数的值。考虑到经累加后的 S 值有可能为零，而 C 中只存储非零系数的值，所以填空④应为 S：0，这样只有在 S≠0 时才将其存储到 C 中。由于 NC 是非零系数的个数，在每一次将 S 存储到 S 中去之后，应该是对 NC 的递加，所以填空⑤处应为 NC+1→NC。

如果将流程图中的 J：2 改成 J：1，则当 J=1 时，B（J，2）：IB 的比较要继续进行。这样当 B（1，2）＜IB 时，J 的递减使 J=0，此时因 J＜1 而终止循环，必须造成③中 B（J，2）的下标 J 超出数组下标范围而发生数组越界。

【答案】

［问题 1］

①A（I，2）：K-（NB）或 A（I，2）或+B（NB，2）：K

②K-A（I，2）→IB 或 K-IA→IB

③B（J，2）：IB

④S：0

⑤NC+→NC

［问题 2］

不能，因为当 J=1 且 B（1，2）＜IB 时将因 J=0 而终止循环，从而导致③中 B（J，2）的下标超出

数组下标范围。

试题 4 （1995 年试题 4）

阅读下列说明和流程图，如图 6-30 所示，回答问题，把解答写在答卷的对应栏内。

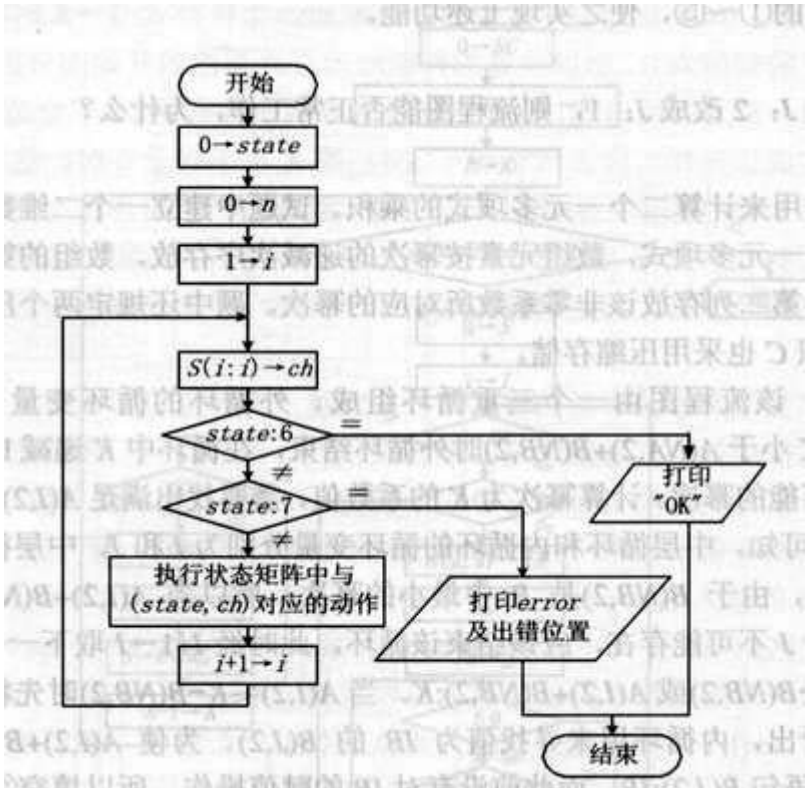


图 6-30　1995 年试题 4 流程图

［说明］

本流程图采用状态转换矩阵的方法来检验算术表达式（非空）的语法正确性，若发现错误，则指出发现错误的位置。

假定算术表达式中的运算对象仅由自然数及变量名（用标识符标识）组成，运算符均是双目运算符，有+ 、-、*、/**（由两个连续的"*"组成，表示乘幂运算）等 5 种，表达式中可以出现左右圆括号，并以字符#作为结尾。

状转换矩阵（见表 6-5）中的行代表当前状态（State），列代表读入字符，矩阵的内容（本题中只给出部分内容，空白部缺省）指出在当前状态下面临相应的读入字符时应执行的操作序列。

例如，若录前状态为 0 时面临的读入字符为"（"，则应执行的操作序例是"n+l→n:4→state" ，即括号嵌套重数加 1，并把当前状态转为 4。

流程图中用字符串 S 存放被检验的算术表达式，用 S（I ：j）表示字串 S 中第 i 个字符至第 j 人字符（j≥i）的子串，其中 S(i:i)即为读入字符。State=5 表示退出一重括号嵌套；state=6 表示表达式语法正确；state=7 表示表达式语法出错。

［问题］

填充状态转换矩阵中的①～⑦ ，把相应的操作序列写在答卷的对应栏内。回答时可以使用如下形式的操作：

if 条件 then 操作 1 else 操作 2 ；

或 if 条件 then 操作；

【解析】

该题采用状态转换矩阵来检验非空算术表达式的语法正确性。

在该题的算术表达式中，可能出现的对象有自然数、变量名、运算符、左括号和括号。仔细阅读流程图可推知，开始状态为 0 状态。试题说明中还提到状态 6 表示被分析的表达式语法

正确，状态 7 表示表达式语法出错，状态 5 表示退出一重括号的嵌套。

解答本题的关键是要明确各个状态的含义。阅读题中给出的流程图及状态转换矩阵可知，在 0 状态（开始状态）时遇到数字（表示表达式以自然数开头），则下一状态为 1，这样我们就可以断定 1 状表示前一个字符是自然数中的一位数字；在 0 状态时遇到字母（表示表达式以变量名开头），则下一状态为 2，这样我们就可以断定 2 状态表示前一字符是变量名中的一个符号；在 0 状态时遇到"（"（表示表达式以左括号开头），则将 n 加 1（从流程图可知，n 的初值为 0，n 加 1 表示进入一层括号），并使下一状态变为 4（即 4 状态表示前一字符是"（"）；在 1 状态时遇到"+"或-或"*"，或"1"，则下一状态为 3。即 3 状态表示前一个字符 运算时。据此，可以认为本题中的状态含义如下所示：

当前状态 前一个字符所属的对象

0　　开始

1　　自然数

2　　变量名

3　　运算符

4　　（

5　　）

6　　正常结束

7　　出错

有了以上的分析，只要我们能够对流程图和试题说明有充分的理解，解答①～⑥就非常容易了，兹不 述。只有⑦的解答应该仔细推敲。阅读状态转换矩阵可知，⑦是在当前状态为 3 时读入字符为"*" 且前一个字符也为"*"时才能组成乘幂次运算符，其他情况下不可能在当前状态为 3 时读入一个运算符，由此可以推断⑦时对"*"的处一。考虑到可能出现"***"这样的情况，所以⑦应为 ifs(I-1:i)="**"ands(I-2:I-2)＜＞"*"then3→state else7→state。

【答案】

①2→state

②2→state

③3→state

④7→state

⑤n-1→n;if n＞=0 then→state else7→state

⑥ifn=0 then6→state else 7→state

⑦ifs(I-I:I)="**"and (I-2:I-2)＜＞"*"then3→state7→state

试题 5 （1995 年试题 5）

阅读下列说明和流程图，如图 6-31 所示，回答问题，把解答写在答卷的对应栏内。

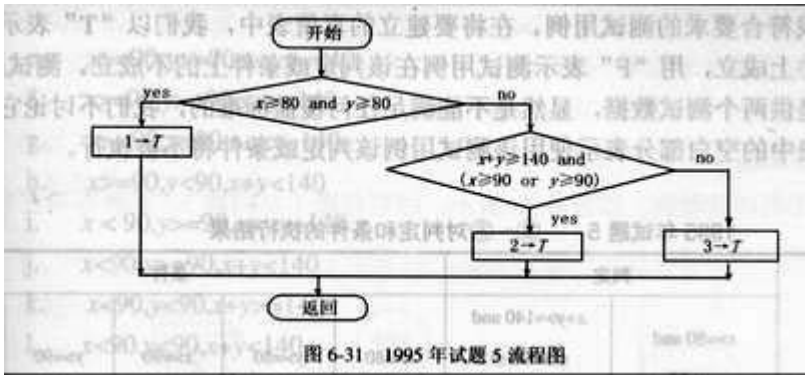


图 6-31　1995 年试题 5 流程图

［说明］

本流程图描述了某子程序的处理流程，现要求用白盒测试法对其进行测试。

［问题］

根据判定覆盖、条件覆盖、判定/条件覆盖、多重和条件覆盖（条件组合覆盖）和路径覆盖等 5 种覆盖标准，从供选择的答案中分别找出满足相应覆盖标准的最小的测试数据组（用①～⑩回答）。

供选择的答案

①x=90,y=90

②x=50,y=50

③x=90,y=90
x=50,y=50

④x=90,y=70
x=40,y=90

⑤x=90,y=90
x=50,y=50
x=90,y=70

⑥x=90,y=70
X=70,Y=90
x=50,y=50

⑦x=90,y=90
x=50,y=50
x=90,y=70
x=70,y=90

⑧x=90,y=90
x=50,y=50
x=90,y=50
x=80,y=80

⑨x=90,y=90
x=90,y=70
x=90,y=30
x=70,y=90
x=30,y=90
x=70,y=70
x=50,y=50

⑩x=90,y=90
x=80,y=80
x=90,y=70
x=70,y=90
x=30,y=90
x=30,y=90
x=70,y=70
x=50,y=50

【解析】

该题要求根据判定覆盖、条件覆盖、判定条件覆盖、多重条件覆盖和路径覆盖等盖标准，选择最小的测试数据组。

解答本题的关键是对 5 种覆盖标准有一个深入透彻的把握。由于此类试题在历年试题中经常出现，我们不妨在这里强调一下。判定覆盖是指选择足够的测试用例，使每个判定的所有可能结果至少出现一次。条件覆闰是指选择足够的测试用例，使判定中的每个条件的所有可能结果至少出现一次。条件覆盖是指选择足够的测试用例，使判定中每个条件的所有可能结果至少出现一次，并且每个判定本身的所有可能结构也至少出现一次。多重条件覆盖是指选择足够的测试用例，使每个判定中条件结果的所有可能组合至少出现一次。路径覆盖是指选择足够的测试用例，使流程图中的每条路径至少经过一次。

解答此类题目的简便方法是建立真值表（见表 6-6），通过测试用例在各个判定和条件上是否满足来寻找符合要求的测试用例。在将要建立的真值表中，我们以"T"表示测试用例在该判这或条件上成立，用"F"表示测试用例在该判定或条件上的不成立。测试用例 1 和测试用例 2 仅提供两个测试数据，显然是不能满足任何覆盖标准的，我们不讨论它们满足哪种标准。真值表中的空白部分表示使用该测试用例或条件将不被执行。

表 6-6　1995 年试题 5　①～⑨对判定和条件的执行结果

编号	判定		条件				
	x<=80 and y<=80	x+y>=140 and (x>=90 or y>=90)	x<=80	y<=80	x<=90	y<=90	x+y<=140
①	T		T	T			
	F	F	F	F	F	F	F
②	F	T	T	F	T	F	T
	F	F	F	T	F	T	F
③	T		T	T			
	F	F	F	F	F	F	F
	F	T	T	F	T	F	T
④	F	T	T	F	T	F	T
	F	T	F	T	F	T	T
	F	F	F	F	F	F	F
⑤	T		T	T			
	F	F	F	F	F	F	F
	F	T	T	F	T	F	T
⑥	F	T	T	F	T	F	T
	F	T	F	T	F	T	T
	F	F	F	F	F	F	F
⑦	T		T	T			
	F	F	F	F	F	F	F
	F	T	T	F	T	F	T
	F	T	F	T	F	T	T
⑧	T		T	T			
	F	F	F	F	F	F	F
	F	T	T	F	T	F	T
	T		T	T			

通过真值表，易知满足判定覆盖的测试用例为⑤，满足条件覆盖的测试用不着例为④，满足判定条件/覆盖的测试用例为⑦。至于路径覆盖，我们首先要考查流程图中共有几条路径，仔细阅读流程图可知，该流程图共有 3 条路径，分别是第一判定为 T、第一判定为 F 且第二判定为 T、第一判定为 F 且第二判定也为 F。从表 6-6 可知供选择答案⑤满足路径覆盖标准。为分析满足多重条件覆盖标准的测试数据，我们先列出两个判定和各种条件组合情况。

第一判定：

a x＞=80,y=80

b x＞=80,y＜80

c.x＞=80,y＜80

d.x＞=80,y＜80

第二判定：

e: x＞=90,y＞=90,x+y＞=140

f. x＞=90,y＞=90,x+y＜140

g. x＞=90,y＞=90,x+y＞=140

h. x＞90,y＜90,x+y＜140

i. x＜90,y＞90,x+y=140

j. x＜90,y＞,x+y＜140

k. x＜90,y＜90,x+y＞=140

I. x＜90,y＜90,x+y＜140

因为当 x＞=90，y＞=90，x+y 不可能小于 140，所以 f 情况不可能出现。此外，当 x＞=90,y＞=90 时，流程图不会执行到第二判定，所以 e 情况也不可能出现。表 6-7 列出了供选择答案⑨覆盖上述二组条件组合的情况。显然，⑨满足多重条件覆盖标准。

表 6-7　1995 年试题 5 答案⑨

编号	测试数据	覆盖的条件组合
⑨	x=90,y=90	a
	x=90,y=70	b,g
	x=90,y=30	b,h
	x=70,y=90	c,i
	x=30,y=90	e,j
	x=70,y=70	d,k
	x=50,y=50	d,l

【答案】
判定覆盖 ⑤
条件覆盖 ④
判定/条件覆盖 ⑦
多重条件覆盖 ⑨
路径覆盖 ⑤

5.7 1994 年度软件设计试题解析

1994 年共有 5 道软件设计试题。试题 1 是软件分析试题，较多地涉及了流程图的评价与修改，这就要求考生在解答题目时将更多的注意力放在具体问题的解决上，仅有对给定流程图的理解还是不够的，关键是把握问题的实质，重在如何把问题的解决反映在流程图的评价与修改上。试题 2 和试题 3 也是难度适中的试题。试题 2 是从一个数组中查找一个特定的元素，试题说明中对算法有所交代，但关把握问题的实质，重在如何把问题的解决反映在流程图的评价与修改上。试题 2 和试题 3 也是难度适中的试题。试题 2 是从一个数组中查找一个特定的元素，试题说明中对算法有所交代，但关键部分语言不详，只能靠考生通过流程图来获得了。试题 3 是数据压缩变换的程序，难度较之试题 2 还要低些。试题 4 的考查形成式比较特别，但难度还是较底。除了流程图修改部分以外，本年度的流程图填空试题总体来说难度较低，试题 5 年 SQL 语言查询比较容易。

试题 1（1994 年试题 5）

阅读下列说明和流程图，如图 6-32 所示，回答问题 1 至问题 3，将解答写在答卷的对应栏内。

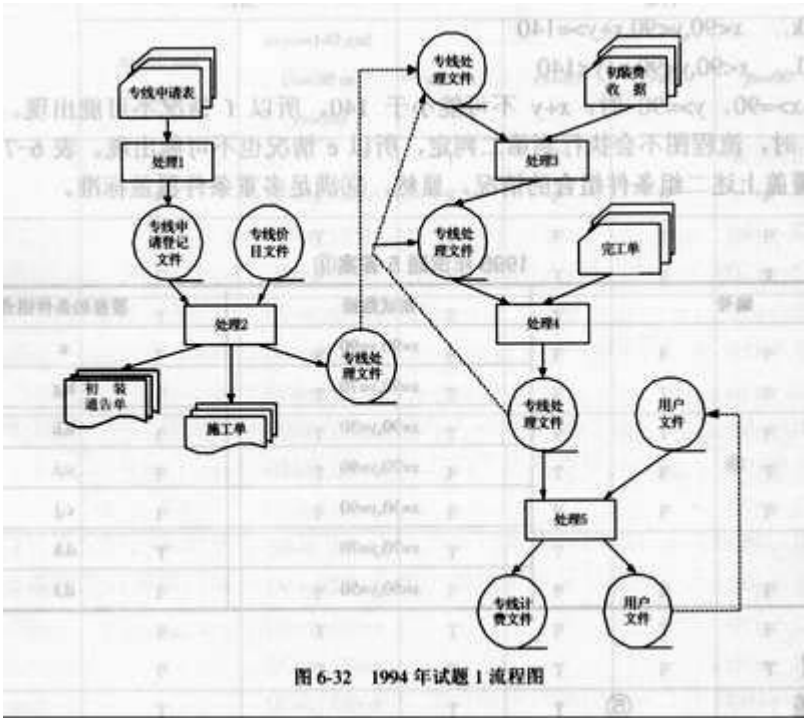


图 6-32　1994 年试题 1 流程图

［说明］

流程图描述了某电信局数据通信专结计费业务管理系统的部分处理流程。

1、凡申请专线者，均需填写专线申请表。系统把申请表存储在专线申请登记文件中，等待分配专线号。

2、系统为申请者指定专线号，并根据通信距离（按地区计算）、通信速率计算初装费和月租费，然后发初装通知单给用户，并产生施工单交有关部门施。同时产生专线处理文件。

3、施工结束后，系统更新用户文件，并产生专线计费文件，作为以后收费的依据。

4、一个用户可以租用多条专线，用户可用现金或银行托付两种方式支付租金，但一个用户只能使用一种付款方式。系统每月按用户（而不是专线）为单位计费出帐。

5、流程图中各数据或文件及有关单据所含的数据项如下。

专线申请表及专线申请登记文件：

申请号、用户名称、付款方式、开户银行代码、帐号、主端名称、主端地址、对端地址、对端所在地区、通信速率、设备接口、申请日期

专线处理文件：

申请号、专线号、用户名称、付款方式、开户银行代码、帐号、初装费、月租费、完工日期

初装费收据：

专线号、初装费、交费日期

施工单：

施工单号、专线号、主端名称、主端地址、对端地址、对端所在地区、通信速率、设备接口、申请日期、施工期限

完工单：

施工单号、专线号、完工日期

用户文件：

用户编号、用户名称、付款方式、开户银行代码、帐号

专线计费文件

专线号、用户编号、月租金、开通日期

［问题 1］
专线价目文件由哪些数据项组成。
［问题 2］
为了避免在用户尚未支付初装费时就去施工，有人提议将图中从处理 2 产生施工单改成从处理 3 产生施工单。试问从处理 3 能否产生施工单？为什么？
［问题 3］
当一个用户租用多条专线时，若允许该用户对其中的一些专线采用现金支付，对另一些专线采用银行托付方式，则在尽量减少数据冗余的前提下，应如何调整有关的数据文件。
【解析】
本题的流程图描述了某电信局数据通信的专线计费业务。
要解答问题 1，应该首先考查专线价目文件在流程图中参与了哪些处理，通过这些和理的输入数据和输出数据的对比来确定专线价目文件由哪些数据项组成。阅读流程图，确认专线价目文件只对与了处理 2。处理 2 的输入数据源是专线申请登记文件和专线价目文件，而输出数据流测是初装单通知、施工单和专线处理文件。结合试题说明，切实把握各文件的数据项组在。除初装通知单在试题说明中没有提到外，输出数据流中的初装费和月租费尚未明确数据来源，应该在专线价目文件中提供。应该注意到，初装费和月租费是根据通信距离和计算初装费和月租费是很困难的。很明显，对某个具体的电信局来说，只要明确了对端所在地区，就相当于明确了通信距离。这样左专线价目文件中只要设置对端所在地区、通信速率、初装费用和月租费 4 个数据项就可以保证系统的正常运行了。
对问题 2 的解答仍然需要考查处理的输入数据流和输出数据流。我们年，处理 3 的输入数据源是专线处理文件和初装费收据，它们数据项的并集中没有生成施工单所需要的主端名称、主端地址、对端地址、对端所在地区、通信速率、设备接口和申请日期等数据项，所以不能由处理 3 来生成施工单，除非由另外的数据文件参与处理 3。
本题的说明中规定，一个用户可以租用多条专线。用户可用现金或银行托付两种方式支付租金，但限制一个用户只能使用其中的一种付款方式。如要取消这一限制，允许租用多条专线的用户对具体的专线采用不同的付款方式，由此应该重新审视属性付款方式所在的实体，付款方式就不再是某个用户的属性，而是某条专线的属性。所以，应该在专线设计费文件中加上付款方式数据项，取消用户文件中的付款方式数据项。
【答案】
［问题 1］对端所在地区、通信速率、初装费、月租费
［问题 2］对端地址、对端所在地区、通信速率、设备接口等数据项。
［问题 3］在专线计费文件中加上付款方式数据项，取消用户文件中的付款方式数据项。

试题 2 （1994 年试验题 2）

阅读下列说明和流程图，如图 6-33 所示，回答问题 1 和问题 2，把解答写在答卷的对应栏内。
［说明］
流程图 6-33 用来实现在数组 A（n）中寻找第 K 大的元素。

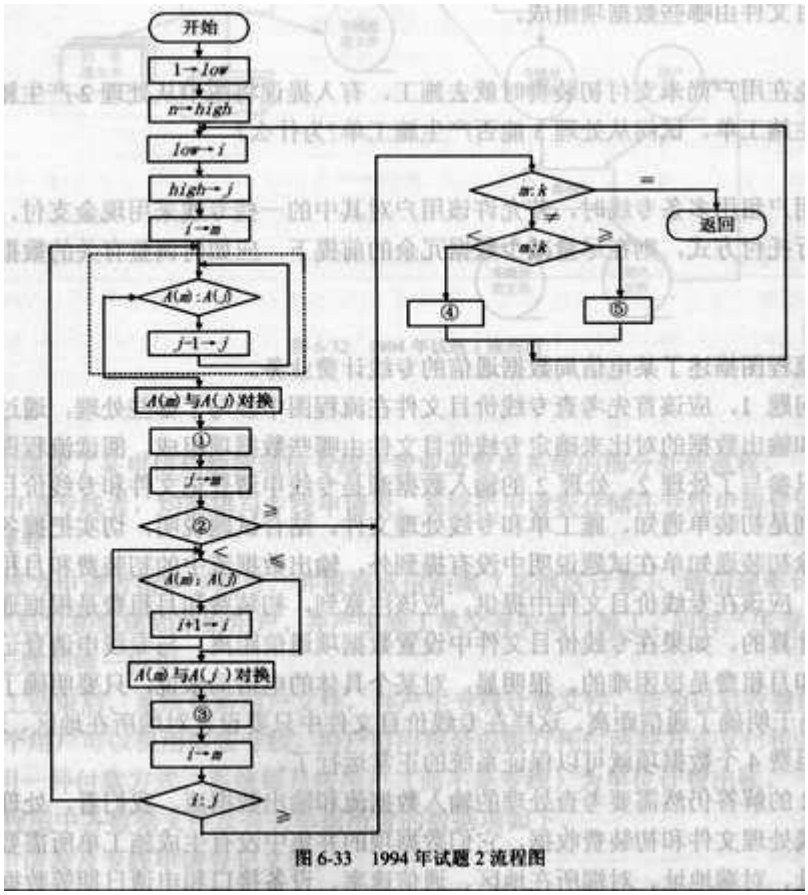


图 6-33　1994 年试题 2 流程图

算法思想是通过比较和交换，在区间 1～n 中找到一个 m,使 A(1)～A(m-1)的值均大于 A(m),A(m+1)～A(n)的值均小于 A（m）即为第 K 大元素，否则调整查找区间，继续上述查找。

［问题 1］

流程图 6-33 中的虚像框部分改成图 6-34 所示的框图，则当 A（1）为数组 A 的最大元素时，修改后的流程图能否正常运行？为什么？

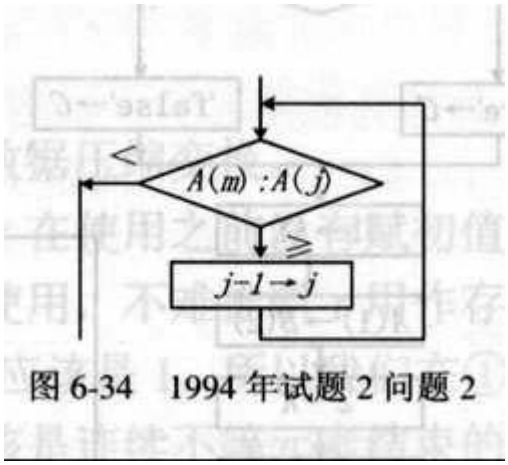


图 6-34　1994 年试题 2 问题 2

【解析】

本题的流程图用来寻找数组 A 中第 K 元素。

流程图中，首先给出了寻找的区间［low, high］，其初值为［i, n］。又给出两个指针 i 和 j，将初值分别不定义为 I=low，j=high，且 m=I。此后是一个循环，该循环完成在 A(m)大于等于 A(j) 时，A(m)与 A(j)的对调。让我们的思路沿流程图继续进行，填空①的作用是什么？

我们且看①以后的语句，先是 m=j，然后是与上面基本相同的循环，只不过参与循环的对象已经是 A(i)与 A(m)。由于前面条件中有 A(i)≥A(j)，所以比较应从 i+1 开始，故填空①应为 i+1→I（或 m+1→）考虑到可能发生对换两个数组元素相邻时本次循环应该结束，即当 i≥j 时结束循环，填空②的解答应为 i:j。本次循环中是数组中靠前的元素依次与靠后的元素比较对换，在发生一次 A(i) 与 A(m)的对换之后。应该实现 j 的递减，然后执行 m→I，重复进行上述工作。显然填空③应为 j-1→j（或 m-1→j）。

如果对试题说明有充分的把握的话，应该能够用确填空④与⑤完成查找区间的调整。考查流程图，若 m=K,则第 K 大的数已经找到，程充结束。否则，若 m＜K，则下次的查找区间为［m+1,high］，所以填空④应为 m+1→low；若 m＞K，则下次的查找区间为［low m-1,］，所以填空⑤应为 m-1→high。

若将流程图中的虚框部分按题中问题 2 所述的方式修改，则当 A（1）为数组 A 的最大元素时，由于 A(1≥)A（j）（j=2，3…，n）所以修改后的流程图将始终执行 j-1→j，以至数组下标越界，因此修改后的流程图不能正常运行。

【答案】

［问题 1］①i+j→i 或 m+1→i

②i:j

③j-1→j 或 m-1→j

④m+1→low

⑤m-1→high

［问题 2］不能，因为当 A（1）为数组 A 最大元是修改后的流程图将始终执行 j-1→j，以至数组下标越界。

试题 3 （1994 年试题 3）

阅读下列说明和流程图，如图 6-35 所示，回答问题 1 和问题 2，把解答写在答案的对应栏内。

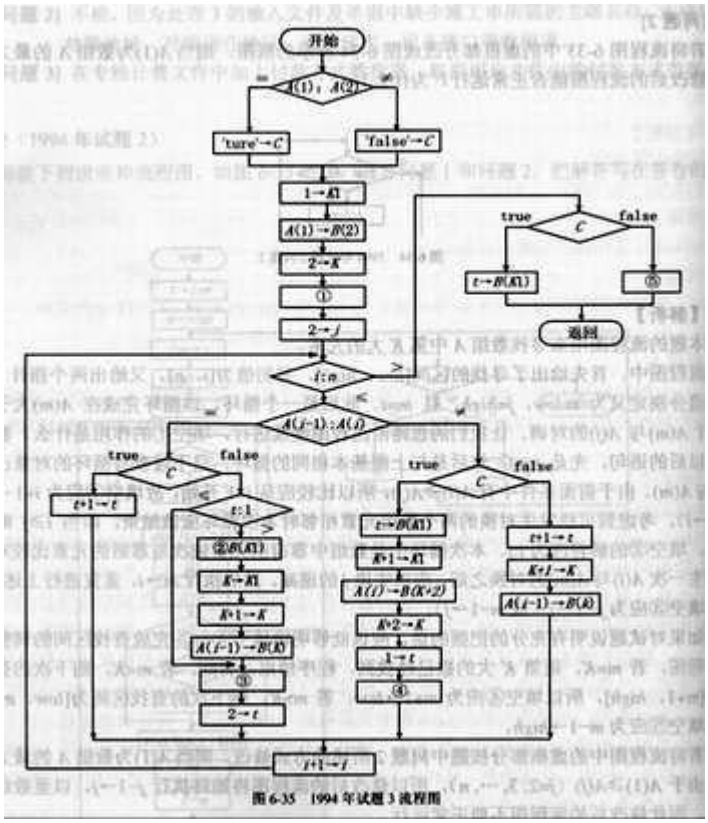


图 6-35　1994 年试题 3 流程图

［说明］

流程图 6-35 用来将数组 A 中的 n(n ≥2)个数经变换后存储到数组 B 中。变换规则如下：

1、若 A 中有连续 t 个相同的元素(t＞1)，则在 B 存入 t 和该元素的值；

2、若 A 连续 t 个元系(t≥1)，其中每个元素都与相邻的元素不相同，则在 B 中存入-t 和这 t 个元素的值。

例如：A={3，3，3，5，7，6，3，6，2，2，2，1，2}

则变换后 B={4，3，2，5，-4，7，6，3，6，4，2，-2，1，2}

流程图中，逻辑变量 C 用来区分正在进行连续相同元素的计数还是连续不等元的计数，KI 用来记录数组 B 存放 t 或-t 的元素的下标。

［问题 1］

填充流程图中的①～⑤，使之成为完整的流程图。

［问题 2］

如果删除流程图中的判断框 t:1，那么，当数组 A+{5，5，4，4}时，经改变后的流程图的变换，数组 B 将会有什么样的元素值？

【解析】

本题的流程图的功能是进行数据压缩变换。

仔细阅读流程图，发现变量 t 在使用之前没有赋初值，这样填空①的作用应该是给 t 赋实值。考查变量 t 在流程图中的使用，不难推断 t 用作存放连续相同元素个数或连续不同元素的个数，很明显，t 的初值就应该是 1，所以我们在①中填写 1→t。结合我们对流程图的理解，填空②所在分支处理的应该连续不等元素结束的情况，这是应该连续不等元素个数的负值赋给 B（K1），那么我们在②中填写-t 吗？如果仔细考查 t 在流程图中的作用，就可以推知这不是正确的答案，因为当前元素 A(I-1)已经不能包含在连续不等元素中了，所以我们在②中填写 1-t。

考查 C 的作用，易知当前处理的是连续相同的元素时，C 的值为 true；当前处理的是连续不

等元素时，C 的值为 false。根据编程常识，这样的逻辑变量应该根据处理对象的不同而变反，如果我们敢于大胆推断，我们应该在填空③和④中填写将 C 变反的语句，即将 C 的值修改为进入该分支时 C 值的逻辑非。回过头来理解流程图，可以确认上述解答的正确性。

填空⑤与语句 1→B（K1）作用应该是相同的，由于最后一个元素页包括含在 t 的计数中，我们已经不能参照②的解答思路，而是填写-t→B（K1）。

值得注意的是，当 A（i-1）≠A(i)且 C='true'时，虽然将'false'送给 C，但下一组并不一定是不同元。在问题 2 中给出的数组 A=（5，5，4，4）就是这种情况。当 i=3 时因原先 C='true'，现在 A（2）≠A（3），因此 1→t，'false'→C，且使 K1=3，K=4。然后 I 加 1 后变成 4，此时因 A（3）=A（4）且 C='false'，并且因 t=1 而执行'true' →C 和 2→t 如果取消流程图中的 t:1 框，那么就把前一组法作不同元来存入，此时 1≠A（3），因此 1-t'fase' →C 且使 KI=3K=4。然后 I 加 1 后变成 4，此时因 A（3）=A（4）且 C='false'并且因 t=1 而执行'true' →C 和 2→t 。如果取消程图中的 r;1 框，那么就把前一组当作不同元来存放，此时 1-t=0→B（3），4→Ki，A（3）→B（5 ），并且 K=5。当 i 加 1 变成 5 后，因 5＞n 所以执行 t→B（K1）那 2+B（K1）那 2+B（4）。因此当删除流程图中的判断框 t:1 后，该数组经变换后，B={2，5，0，2，4}。

【答案】

［问题 1］①1→t ②1-t ③'true'→C ④'false'→C ⑤-t→B(K1)

［问题 2］B={2，5，0，2，4

试题 4 （1994 年试题 4）

阅读下列说明和流程图，如图 6-36 所示，回答问题 1 至问题 3，把解答在答卷的对应栏内。

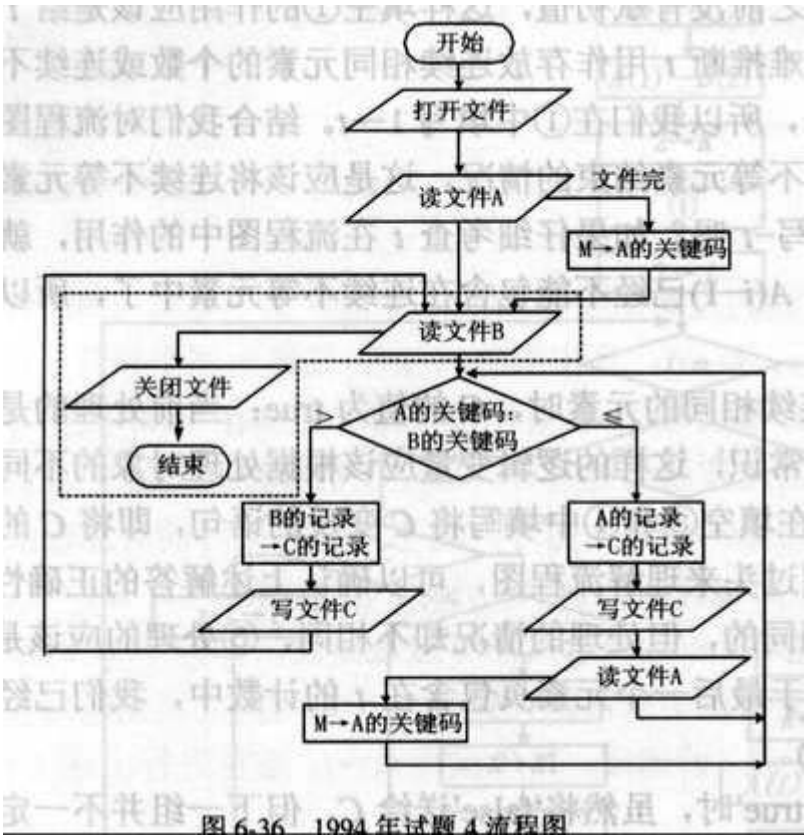


图 6-36　1994 年试题 4 流程图

［说明］

流程图 6-36 用于把文件 A 和文件 B 合并成按上升顺序分类（排序）的文件 C。已知文件 A 和 B 中的关键码均小于 M。

［问题 1］

为使流程图 6-36 能正确工作，文件 A 和文件 B 诸记录的关键码必须满足什么条件？

［问题 2］

按流程图 6-36 的处理方式，分别指出文件 A 和文件 B 至少应含有的记录个数。

［问题 3］

若将流程图 6-36 中的虚线部分改为框图 6-37，则图中的"判断条件"应是什么？

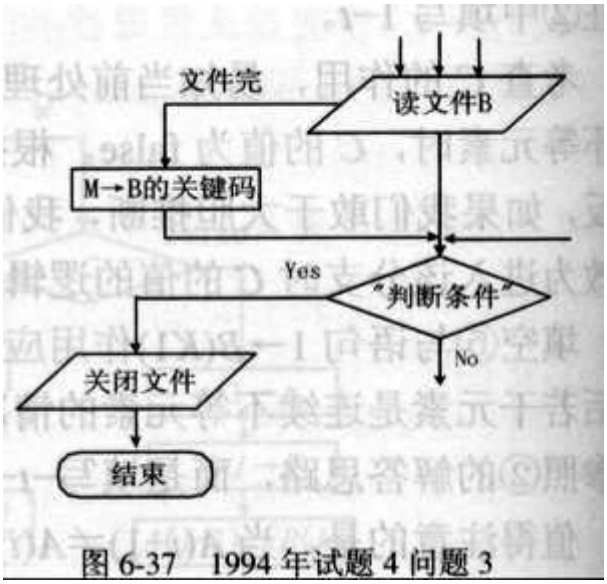


图 6-37　1994 年试题 4 问题 3

【解析】

本题的流程图用业实现两个文件的合并排序。

阅读流程图可知，图中逐个读取文件 A 记录和文件 B 的记录，当 A 记录的关键码大于 B 记录在关键码时，将 B 记录写入文件 C；反之，将 A 记录写入文件 C。显然，利用上述算法实现文 C 按升序排列，要求文件 A 和文件 B 在合并前必须已按升序排列，否则不能保证程序达到预期的要求。

流程图中，当文件 A 读完时，将 M 赋给 A 记录的关键码。结合试题说明可知，文件 A 和文件 B 中的关键码均小 M。所以，当赋给 A 记录的关键码后，A 的关键码总是大于 B 的关键码，使 B 记录依次写入文件 C，直至文件 B 读完。也就是说，当文件 A 先读完时能确保将文件 B 的剩余记录全部写入文件 C。但是，当文件 B 先读完时，就立即关闭文件，并结束运行。这样，当文件 B 先读完时，文件 A 的剩余记录无法写入 C 中。为达到将文件 A 和 B 的所有记录都合并到文件 C 中去的目的，本流程图要求必须先读完 A，后读完 B。这要求文件 A 的最后一个记录的关键码应该小于或等于文件 B 的最后一个记录的关键吗。

阅读流程图可知，当文件 A 为空时，流程图能正常运行（即使文件 B 也为空），运行的结果是文件 B 的所记录依次写入 C。然而，当文件 A 非空时，文件 B 必须非空，否则，文件 A 记录无法写入文件 C，因为文件 B 的空将导致程序的结束。因此文件 A 至少有零个记录；当文件 A 非空时，文件 B 至少有 1 个记录；当文件 A 为空时，文件 B 至少有零个记录。

根据上述分析可知。该流程图要求文件 A 的最后一个记录的关键码应小于或等于文件 B 的最后一个记录的关键吗，这人要求使程序对数据的要求非常苛刻，显然不是一个成功的程序。

问题 3 将流程图作了改进，即当文件 B 读完时也将 M 赋给 B 记录的关键吗。这样不管哪个文件先读完，都能确保未读完的文件的剩余记录依次写入文件 C 中。由于将 M 赋给 A（或 B）记录的关键码的操作只能发生在文件 A（或 B）读完时，所以此时结束运行的判断条件应是两个文件都已经读完，即 A 的关键码等于 M 并且 B 的关键码等于 M。

【答案】

［问题 1］文件 A 和文件必须均按升序顺序，并且文件 A 的最后一个记录的关键码应小于或等于文件 B 的最后一个记录的关键吗。
［问题 2］文件 A 至少有个零个记录；当文件 A 非空时，文件 B 至少有 1 个记录；当文件 A 为空时，文件 B 至少有零个记录。
［问题 3］A 的关键码等于 M 并且 B 的关键吗等于 M。

试题 5 （1994 试题 5）

阅读下列说明和 E-R 图 6-38，回答问题，把解答写在答卷的对应栏内。

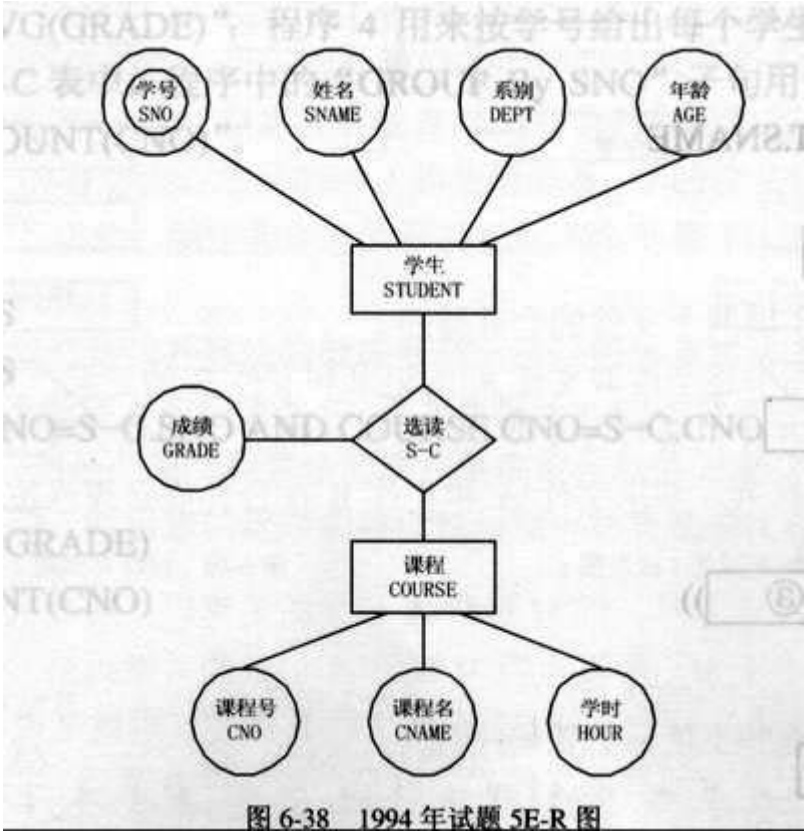


图 6-38　1994 年试题 5E-R 图

［说明］
设有下列关于学生成绩管理系统的 E-R 图。图中矩形表示实体，圆表示属性，双圆表示关键字属性，菱形表示实体间的联系。假定已通过下列 SQL 语言建立了基本表：

```
CREATETABLESTUDENT
（SNOCHAR（6）NOTNULL，
SNAMECHAT（20），
DEPTCHAR（20），
AGESMALLINT）；
CREATETABLECOURSE
（CNOCHAR6）NOTNULL，
CNAMECHAR（20），
HOUPSMALLINT）；
CREATETABLES-C
（SNOCHAR6），
CNOCHAR（6），
GRADESMALLINT）；
```

为了答题的方便，图 6-38 中的实体和属性时给出了中英两种名字，回答问题时只须写出英文名即可。

［问题］

填充下列 SQL 程序 1～4 中的①～⑥，使它们分别成以下查询功能。

程序 1：检索选读所有课程的学生姓名。

程序 2：给出全体学生人数。

程序 3：按学号给出每个学生的平均绩。

程序 4：按学号给出每个学生选读课程的门数。

［程序 1］

```
SELECTSTUDENT.SNAME
FROMSTUDENT
WHERE ①
（SELECT*
FROMCURSE
WHERE ②
（SELECT*
FROMS-C
WHERE ③
```

［程序 2］

```
SELECT ④
FROMSTUDENT
```

［程序 3］

```
SELECT ⑤
FRMS-C
GROUPBY SNO
```

［程序 4］

```
SELECT ⑥
FROMS-C
GROUPBY SNO
```

【解析】

本题采用 SQL 语言对基本表进行查询操作。解答此题的关键在于对 SQL 语言对基本表进行查询操作。解答此题的关键在于对 SQL 语句的透彻理解和灵活运用。从历年考试来看对 SELECT 语句的考果、查是重点，能否灵活地构造 SELECT 查询是提高此类题目得分率的关键。

程序 1 用来检索选读所有课程的学生姓名，程序 1 是一个嵌套的查询语句。最外层的 SELECT 是在 STUDENT 表中查找学生的的姓名，中层的 SELECT 在 COURSE 表中查找元组，最内层的 SELECT 在 S-C 表中查找元组。根据题目要求，找到的学生应"不存在这个/这些学生不学的课程"，即寻找那些中层 SELECT 的查找结果是空集的学生，所以填空①应为"NOTEXISTS"。同样，中层 SELECT 查找的课程应"是这个/这些学生不学的"，即寻找那些最内层 SELECT 的查找结果是空集的课程，所以填空②也是"NOTEXISTS"。填空③显然应是"STUDENT.SNO=S-C.SNO AND COURSE.CNO=S-C.CNO"。

程序 2 用来给出全体学生的人数，全体学生的人数就是 STUDENT 表中的元组个数，用 SQL 的内部函数 COUNT（*）即可得到，故填空④为"COUNT（*）"程序 3 用来按学号给出每个学生的平均成绩，学生成绩存放在 S-C 表中，程序中的"GROUPBySNO"子句用来按学号 SNO

分组，学号为 SNO 的学生的平均成绩可用 SQL 的内部函数 AVG（GRADE）计算，故填空⑤为"SNO，AVG（GRADE）"：程序 4 用来按学号给出每个学生选择读课的门数，学生的选课情况存放在 S-C 表中，程序中的"GROUP By SNO"子句用来按学号 SNO 分组，故填空⑥ "SNO，COUNTCNO"。

【答案】

①NOTEXISTS

②NOTESISTS

③STUDENT.SNO=S-C.SNO AND COURSE.CNO=S-C.CNO

④COUNT(*)

⑤SNO,AVG(GRADE)

⑥SNO,COUNT(CNO)

5.8 1993 年度软件设计试题解析

试题 1 提供了两种算法来生成一个数字序例，最后要求考生比较两种算法的优缺点。本题对考生的能力要求较高，而且由于提供了两种算法，解答试题的工作量也非常大，试题 2 竟然又是字符处理，这与 1992 年度试题 2 有雷同之处，突出了研究历年试题的重要性。试题 3 考查的是二叉树的非递增归遍历，这一直是一个难点，而且这里的二叉树又是一个顺序存储年度试题 4 大体相同，涵盖了数据文件、排序关键项和提高效率 3 个方面。另外几道试题涉及输出数字序列特定位置的数码、字符处理、SQL 语言、二叉树的非递归遍历等几个方面的问题。试题 5 是 SQL 语言，只要认真学习过数据库知识，解答本题还是比较容易。

试题 1（1993 年试题 1）

阅读下列说明和流程图，如图 6-39 和图 6-40 所示，回答问题 2，把解答写在答卷的对应栏内。

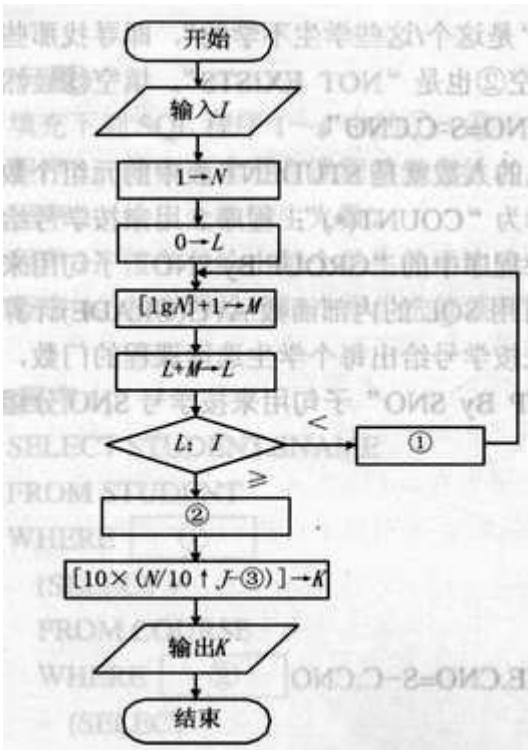


图 6-39　1993 年试题 1 流程图 1

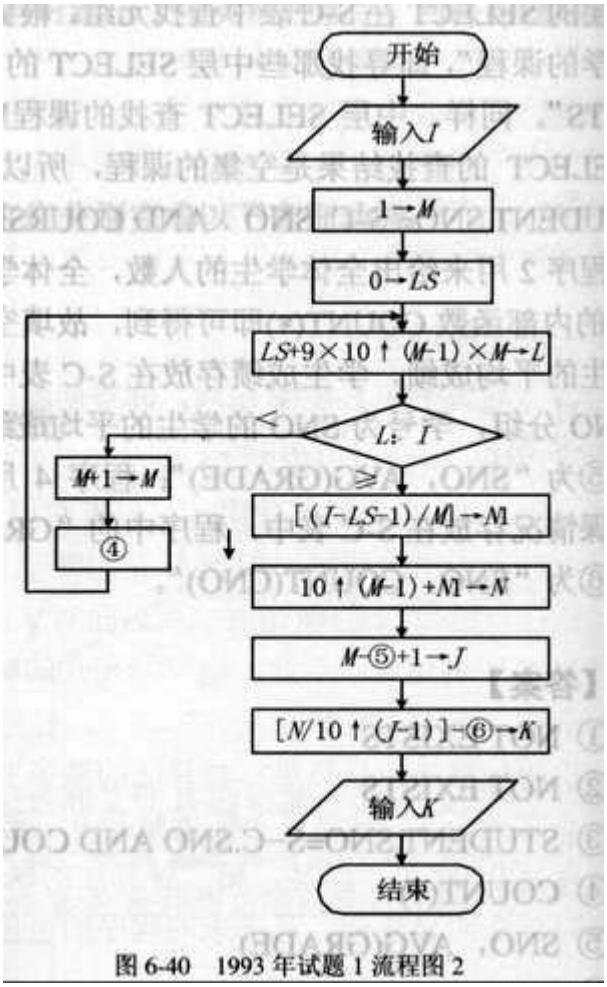


图 6-40　1993 年试题 1 流程图 2

［说明］
将自然数依次排列成如下所示的数码序列：
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16…
流程图 6-39 和流程图 6-40 都能输出从头数起的第 1 个数码。
流程图中 K 存输出数码，N 存放自然数，M 存放自然数的位数。图中表示乘幂运算，［W］表示不超过 W 的最大整数。
流程图 1（图 6-39）采用逐个增添自然数的方法。
流程图 2（图 6-40）采用一次增添位数相同的自然数序列段的方法。
［问题 1］填充流程图 1 和流程图 2 中的①～⑧ ，使之成为完整的流程图。
［问题 2］比较流程图 1 和流程图 2 的优缺点。
【解析】
本题提供了两种算法，从给定的数码序列中查找并输出第 I 个数码。仔细阅读试题说明中关于变量含义的说明，这对理解流程图有重要的作用。
先研究第一种算法。从流程图的开始部分往下阅读，结合对变量含义的理解，［lgN］+1 应该是自然数 N 的位数。从 L+M→来看，L 应该是到自然数 N 为止在序列中所占的位数。当 L 大于或等于 I 时，即表示 I 应该从此自然数中取得，所以我们在填空①中填写 N+1→N。接下来，填空②后面的语句使用了变量 J，但此前流程图中并没有给 J 赋值的操作，在程序中不经过赋值而直接使用量是不可思议的，由此可断定②是给 J 赋值的语句，但究竟赋什么值呢？这要结合上下文理解。从②的后续语句可知，J 表示序列中的第 I 个数码在 N 中的位数，所以我们在②中填写 L-I+I→J。为什么断定 J 表示序列中的第 I 个数码在 N 中位数呢？这要结合对填空③的解答来看。②的后续语句，［10x(…)］→K，就是说，10X（…）的结果是一个不大于 10 的实数，该实数的整数部分即为序列中的第 I 个数码。括号中得出的结果是一个小数，该小数的十分位为序列中的第 I 个数码，N/10J、③即为这个小数，由此印证 J 确实是序列中的第 I 个数码在 N 中的位数，并且了推断出填空③的解答应该是 N10J 的限整。
研究第二种算法，本算法采用一次将位数相同的自然数计入序列长度的方法。9×10（M-1）×M 实际上计算了位数为 M 的然数在序列中所占的长度，详细原因，不复详述，请读者自己理解。由此可以推断 LS 为已经计算的位数，L 为计入该段自然数后在序列中所占的总长度。由于 LS 需要循环使用，可以推断填空④的解答应该是 L→LS。当 L 大于等于 I 时，结合以上对流程图的理解，NI 应该是自然数 N 在当前自然数段中的位置，10（M-1）+NI 计算出应该在自然数 N 的确切值。从后面对 J 使用来看，J 的含义仍旧是序列中第 I 个数码在自然数 N 中的位置。至此，填空⑤和⑥的解答已经非常明显了，我们在这里给读者留一点思考的余地，不一一解释了。
分析两种算法，显然第二种算法的效率要高一些，因为该算法中的循环部分在实际执行中效率非常高。第 1 次循环 L 的增量是 9，第 2 次循环 L 的增量就达到了 180，第 3 次循环 L 的增量就是 2700 了。但是，第二种算法的缺点是确定自然数 N 和 N 中的第 J 位为序列所求的数位时算法明显复杂，写出来的程序的可读性与第一种算法相比相去甚远。
[答案]
［问题 1］①N+1→N ②L-I+1→J ③[N/10↑J] ④L→LS ⑤(I-LS-N1*M) ⑥[N/10↑J]×10
［问题 2］流程图 2 的处理效率比流程图 1 高，流程图 2 的算法比流程图 1 复杂。

试题 2 （1993 年试题 2）

阅读下列说明和流程图，如图 6-41 所示，回答问题 1 至问题 3，。把解答写在答卷的对应栏内。

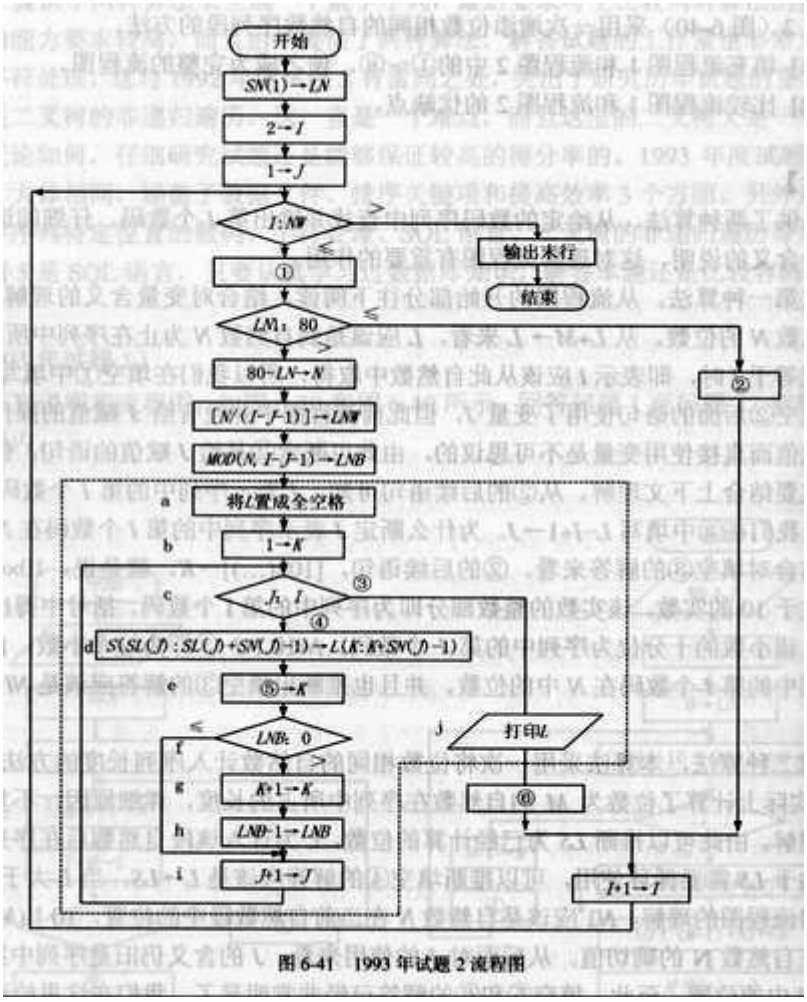


图 6-41　1993 年试题 2 流程图

［说明］　本流程图的功能是对联预处理后的正文件进行排版输出。

假定预处理后的正文件存放在字符串 S 中，S 由连续的单词组成，单词由连续的英文字母组成。在预处理过程已产生以下信息：

变量 NW 存放正文中单词的个数，数组元素 SL（I）存放正文中第 I 个单词在 S 中的字符位置，SN（I）存放正文中第 I 个单词的长度。规定 S 中的字符位置从 1 开始计数，每个字符占一个位置。字符串 S 中的某个单词可用如下的子串形式来存取：

S（单词起始位置：单词终止位置）

并规定化在对字符串（或子串）赋值时，赋值号两端的字符串（或子串）长度必须相等。

排版输出的要求如下：

①每行输出 80 个字符;

②一个单词不能输出在两行中；

③除最后一行外，所有输出行既要左对又要右对齐。即每行的第一个字符必须是某个单词的第一个字母，最后一个字符必须是某个单词的最后一个字母；

④单词之间必须有 1 个或 1 个以上的空格；

⑤最后一个行只须左对齐，且单词之间均只有一个空格；

⑥使空格尽可能地均匀分布在单词之间，即同一行中相邻的单词的空格数量多相差 1。

假定正文中至少有两个以上的单词，每个单词的长度均小于 40。此外，流程图中省略了数据的输入部分。图中［W］表示不超过 W 的最大整数。

［问题 1］填充流程图中的①～⑥，使之成为完整的流程图。

［问题 2］图中的"输出末行"框未经细化，如果将图中的虚线部分复制到"打出末行"框上，那么复制部分应作怎样的修改？可用图中所标的 a、b……j 来回答，例如 a 改成 1→I ，删除 b。

［问题 3］ 如将图中开始部分的 SN（I）→LN 改成 0→LN，2→I 改成 1→I，则修改后的流程图是否正确。

【解析】

本流程图实现预处理后的正文排版输出。试题说明中给出了变量含义和排版的各种要求，我们应该形成如下的概念：流程图中使用了这些变量来实现符合要求的排版输出，各种要求都会在流程图中得到体现和实现，因此对变量说明和排版要求的切实理解就成为解答题目之前极为重要的工作。

阅读流程图，不难以现在①之后开始了对变量 LN1 的使用，由此推知①的作用是为 LN1 赋值，问题是赋什么样的值。考查下文，从 LN1：80 来看 LN1 应该是到此为此已经算在行中的单词和空格的长度（不能在最后一个单词后面缀上一个空格，这可以由试题说明的（3）推知）。那么此时的 LNI 究竟应该是什么呢？从流程图的开始部分仔细读下来，显然应该是 LN+SN（I）+1，这是符合要求的表示方式。解答了①，让思路顺着判断 LN1：80 小于等于的分支向前发展，LN1→LN 作为②的解答是顺理成章的事情，否则第二次运行到①处如何进行？

沿着 LN：80 的大于分支研读流程图。首先我们能够判断这是对单词总长度加间隔空格长度大于 80 后的处理。80-LN-N，这里的 LN 是尚未加入第 I 个单词时的行长度（包括间隔空格），因为加入第 I 个单词及该单词前的空格后行长将大于 80，所以不能将第 I 个单词包括在本行内。由此可知 N 的含义就是到第 I-1 个单词为止（包括间隔空格）本行剩余的位置（空格）。基于以上分析，以后的两句，先求将剩余的空格均匀分布在各单词的间隙里时每个间隙分担多少空格，后求均匀分担后尚剩余多少空格无法均匀分布。由此我们可以推断 J 是本行第一个单词的序号。这样③和④的答案就很明显了。注意③填写大于等于是因为第 I 个单词不包含在本行内。不准理解 d 处的语句是取本行某个单词到 L 中，但⑤应该给 k 赋什么值？读到 J+1→J，我们才知道在 d 处之后需要计算下一个单词的超始位置。这里有 3 项工作要做，一是计入第 J 个单词的长度 SN（J），二是计入正常间隔空格的长度 L，三是计入每两个单词之间需要分担的均匀分布的空格数 LNW。现在这 3 项工作都需要在⑤中完成，所以⑤的答案应该是 K+SN（J）+1+LNW。

现在看⑥打印完一行后理所当然要打印下一行。在流程图的开始部分有 SN(1)→LN 所以这里的解答与之相似，应该是 SN(I)→LN ，以便开始新一行的处理，否则，LN 的值将停留在上一次 LN1→LN 时。注意这里的 LN 是上一行的最后一个单词，即第 I-1 个单词的长度，如果将该值带入下一行的处理，将会得到错误的结果。的处理，将会得到错误的结果。

处理未行要考虑的问题显然要少许多。f 处、g 处与 h 处是可以删除的，因为这里已经不存多余空格的均匀分布问题了，谈到这一点，⑤处应该改为 K+1+SN（J）。

如果按问题中的方案修改案修改流程图，结合①的解答，这将导致在统计第一行第一个单词时多加了一个 1，程序执行的结果当然也就是错误的。如果怀疑是在解答①时出错。那么可以试一试，这样的修改要改动多少地方才能实现试题说明所提出的各种要求。

[答案]

［问题 1］①LN+1+SN(I)→LN1 ②LN1→LN ③≥

④< ⑤K+1+LNW+SN(J) ⑥SN→LN

［问题 2］删去 f,g,h 框，将 e 改成 K+1+LNW+WN(J)→K。

［问题 3］不能。

试题 3

［说明］

本流程图对顺序存储的二叉树按排非递归形式进行后序遍历打印。顺序存储的二叉树存放在数组 data（I:m）中，data 存放结点的值，right 存放指针值。

本题中顺序存储的二叉树是指对树中任意两个结点 nodel 和 nde2（它们在顺序存储数组中的下标分别为 q1 和 q2），它们的指针值满足下列条件：

（1）如果子 nodel 是根结点， 则 q1=1。如果从 nodel 出发按前序遍历所得到的下一个结点是 node2，则 q2=q1+1。

（2）如果 nodel 的左后件是 node2,则：

如果 nodel 存在左后件，则 right(q1)=(q2+1)。

如果 nodel 不存在左后件，则 right(q1)=-(q2+1)

（3）如果 nodel 没有右后件，则 right（q1）=1；

如果 nodel 存在左后件，则 right(q1)=;

如果 nodel 不存在左后件，则当 node1 是按前序遍历的最后一个结点时，right(q1)=0，否则 right(q1)=1。

例如，下列二叉树的顺序存储情况见表 6-8。

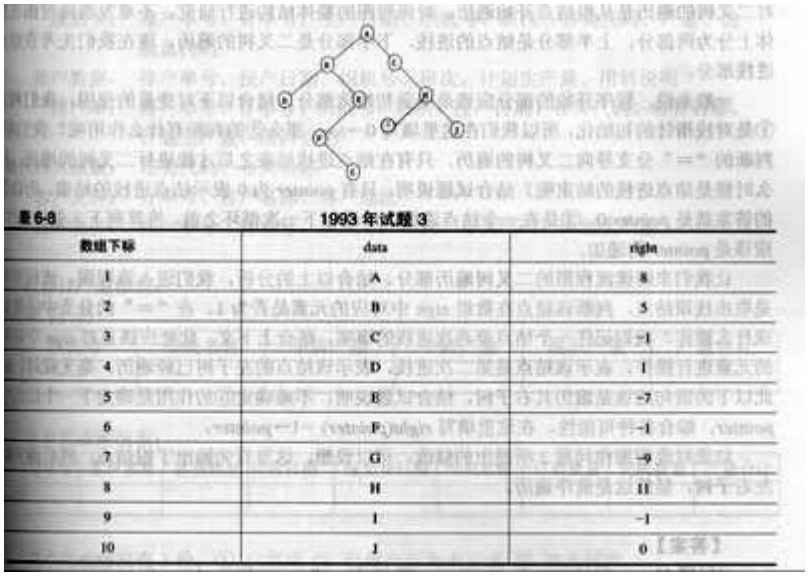


表 6-8　1993 年试题 3

数组下标	data	right
1	A	8
2	B	5
3	C	-1
4	D	1
5	E	-7
6	F	-1
7	G	-9
8	H	11
9	I	-1
10	J	0【答案】

流程图中 stak(1:N)是遍历过程中存放顺序存储数组下标的栈，sig(1:N)是配合栈操作设立的标志位（第一次进行栈时值为 1，第二次进栈时值为 2），变量 top 是栈顶指针，变量 first 是顺序存储二叉树的首指针。若树空，则 first=0，否则 first=1。指针 pointer 用来指出结点在数组中的的下标。

假定给出的顺序存储二叉树是正确的，且 stack 和 sign 都足够大， 不会溢出。

［问题 1］ 填充流程图中的①～⑤，如图 6-42 所示，使之成为整的流程图。

［问题 2］将流程图中的"打印 data(pointer)"框移至@处。则流程图执行的结果是什么？

【解析】

本题实现了一种按非递归形式对顺序存储的二叉树进行后序遍历输出的算法，仔细阅读试题说明，可以发现本题的难度不在于二叉树的后序遍历，而在于试题中提供的顺序存储方式，只有切实理解了了这种顺序存储方式，才能理解本题实现的后序遍历打印算法。

首先明确二叉树的前序遍历和后序遍历。前序遍历是指先输出根结占，再按前序遍历根结点的左子树，最后按前序遍历根结点的右子树。后序遍历是指先按后序遍历根结点的左子树，再按后序遍历根结点的右子树，最后输出根结点。我们知道，按非递归形式对二叉树进行遍

历时需要栈的支持，一个结点在遍历过程中要两次进栈，第一次进栈是为了按后序遍历以该结点为根结点的左子树，第二次进栈是为了按后序遍历以该根结点的右子树。

首先我们要仔细推敲试题说明中的（1）、（2）和（3），对顺序存储的二叉树各结点的指针值的计算进行研究。然后阅读流程图，流程图的第一句就是将 first 赋给 pointer，这意味着对二叉树的遍历是从根结点开始遍历。对流程图的整体结构进行研究，不难发现流程图在整体上分为两部分，上半部分是结点的进栈，下半部分是二叉树的遍历。现在我们先考查结点进栈部分。

一般来说，程序开始的部分应该是变量初始化部分，结合以下对变量的使用，我们判定①是对栈指针的初始化，所以我们在这里填写 0→top。那么 ②的判断有什么作用呢？我们看，判断的"="分支导向二叉树的遍历，只有在结点栈结束之后才能进行二叉树的遍历。什么时候是结点进栈的结束呢？结合试题材说明，只有 pointer 为 0 表示结点进栈的结束，所以②的答案就是 pointer:0。③是在一个结点进栈之后进行下一次循环之前，推算剩下未做的工作，应该是 pointer 是递加。

让我们来研读流程图的二叉树遍历部分。结合以上的分析，我们进入流程图，流程图先是取出栈顶结点，判断该结点在数组 sign 中对应的元素是否为 1，在"="的分支中④要完成什么操作？我们记住一个结点要两次进栈的事实，结合上下文，此处应该是对 sign 中对应的元素进行操作，表示该结点是第二次进栈，表示结点的左子树已经遍历。毫无凝问，由此以下的语句应该是遍历其右子树，结合试验题说明，不难确定⑤的作用是确定下一个结点的 poiner，综合各种可能性，在这里填写 right(pointer)-1-pointer。

如果对流程图作问题 2 所提出的修改，可以设想， 这里首先输出了根结点，然后遍历其左右子树，显然这是前序遍历。

[答案]

［问题 1］①0→top ②pointer:0 ③pointer+1→pointer

④0→sign(top) ⑤right(pointer)-1→pointer

［问题 2］按前序遍历。

试题 4 （1993 年试题 4）

阅读下列说明和流程图，如图 6-43 所示，回答问题 1 至问题 3，把解答写在答卷的对应栏内。

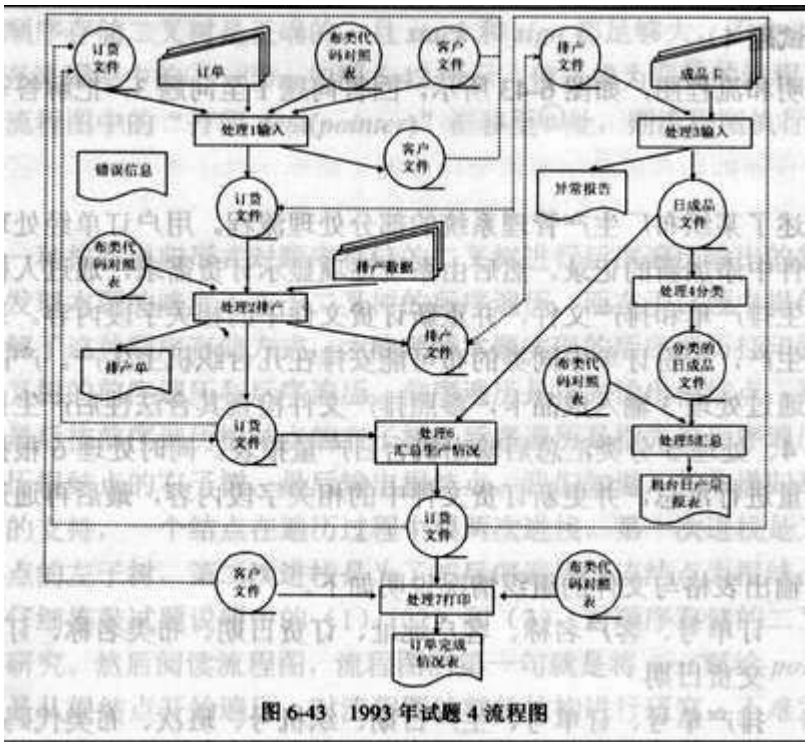


图 6-43　1993 年试题 4 流程图

［说明］

本流程图描述了某织布厂生产管理系统的部分处理流程。用户订单经处理 1 输入，在订货文件和客户文件中添加新的记录。然后由系统逐项显示订货需求，通过人机交互在处理 2 中进行排产，阁下排产单和排产文件，并重新订货文件中的相关字段内容。系统产生的排产单交给值机人员生产，一份订单中同类的布可能安排在几台织机上生产。产品合格后填写成品卡。系统每天能过处理 3 输入成品卡、参照排产文件检查其合法性后产生日成品文件。日成品文件经处得 4、处理 5 分类汇总后输出机台日产量报表。

输入单据、输出表格与文件的组织情况说明如下。

订单：订单号、客户名称、客户地址、订货日期、布类名称、

订货数量、应交货日期

成品卡：排产单号、投产日期、织机布、班次、布类代码、产量、

值机员代码

排产数据：排产单号、投产日期、织机号、班次、计划生产量、

用料说明

排产单：排产单号、订单号、织机号、班次、投产日期、布类代码、

布类名称、计划生产量、用科说明

布类代码对照表：布类代码、布类名称

客户文件：订单号、客户名称、客户地址

排产文件：排产单号、订单号、织机号、班次、布类代码、计划生产量

机台是产量报表：

年 月 日

织机号	布类名称	早班产量	中班产量	晚班产量	产量合计

订单完成情况表：

订单号	客户名称	客房地址	订货日期	应交货日期	布类名称	订货数量	完成数量	产量合计

其中完成情况有 3 种：①已完成 ②已排产但俚尚未完成 ③尚未财产。

［问题 1］分别指出订货文件和日成品文件至少应包括哪些数据项。

［问题 2］指出处理 4 分类的第一、第二关键项。

［问题 3］为提高处理 6 的效率，流程图（如图 6-43 所示）需作何补充？（用文字简述）

【解析】

本题描述了某织布厂生成管理系统的部分处理流程。仔细考查试题说明和流程图，发现在阅读流程图时要注意订货文件和日成品文件在系统中的数据来源和数据流向，还要注意处一 4 和处理 6 处理的特点。

定货文件的主要数据来源是订单，考查布类代码对照表和客户文件，为减少存储和处理订单文件的系统开销，订单中和客户名称和客户地址在以后的处理中可以通过订单号从客户文件中取得。布类名称 可以简人继布类代码，在以后的处理中通过布类代码对照表取得。仔细向后阅读，在处理 7 通过订单文件产生订单完成情况表，订单中的完成数量和完成情况尚未在系统中出现，我们怀疑它来自日成品文件，但处理 6 汇总生产情况时就该是将日成品文件中的苛些数据写入订单文件，然后由订单文件独自参国处理 7，这样，我们就把完成数量和完成怀况两个数据项加入订单文件。

考查是日产文件，发现日成品文件的主要作用不是参与处理 6，而是流程图的右半部分，经处理 4 和处理 5 最后生成机台日产量报表。机台日产量报表中的数据必须能够直接或间接从日成品文件取得。由于日成品文件是参与处理 6 的，如果该文件中没有数据项订单号，则在处理 6 中难以实现完成数量和完成情况的汇总；由于处理 5 有布类代码对照表的参与，在日成品文件的中尽可以使用布类代码而不是布类名称。此外，机台日产量报表中有早班产量、中班产是、晚班产量和产量合计 4 个产量数据，我们可以在日成品文件中加入班次和产是两个数据项来支持以上数据需求。这样，包括日成品文件中不可缺少的织机号，日成品文件中就应该包括含订单号、织机号、布类代码、班次和产量 5 个数据项。

考查处理 4 的和理对象日成品文件和分类的日成品文件经处理 5 生成机台日产量报表等事实，如果处理 4 中分类关键项设置合理的意义就在于提高处理 5 的效率。由机台日产量报表的数据结构可知 ，知机号和布类名称的有序性对处理 5 的效率由很大影响，所以我们断定处理 4 分类的第一关键项是织机呈，第二关键项是布类代码。

考虑处理 6 进行汇总处理，提高处理 6 的效率的关键在于处理对象的有序性，并且两个处理对象是否经过按相同的关键项进行的有充化处理。考查日成品文件和订单文件，发现它们都可以近订单和布类代码进行有序化处理，由于处理 6 的主要功能是对订货文件进行处理，所以如果订货文件和日成品文件在通过处理 6 时先按订单号（第一关键项）和布类代码（第二关键项）进行有序化处理，处理 6 的处理效率必将大大提高。

【答案】

［问题 1］ 订货文件：订单号、订货日期、应交货日期、布类代码、订货数量、完成数量、完成情况。

日成品文件：订单号、织机号、布类代码、班次、产量

［问题 2］第一关键项是织机号，第二关键项是布类代码。

［问题 3］在处理 6 前分别对订货文件和日成品文件按相同的关键项（订单号和布类代码）分类。

试题 5 （1993 的试题 5）

阅读下列说明和 E-R 图 6-44，回答问题 1 至问题 3。

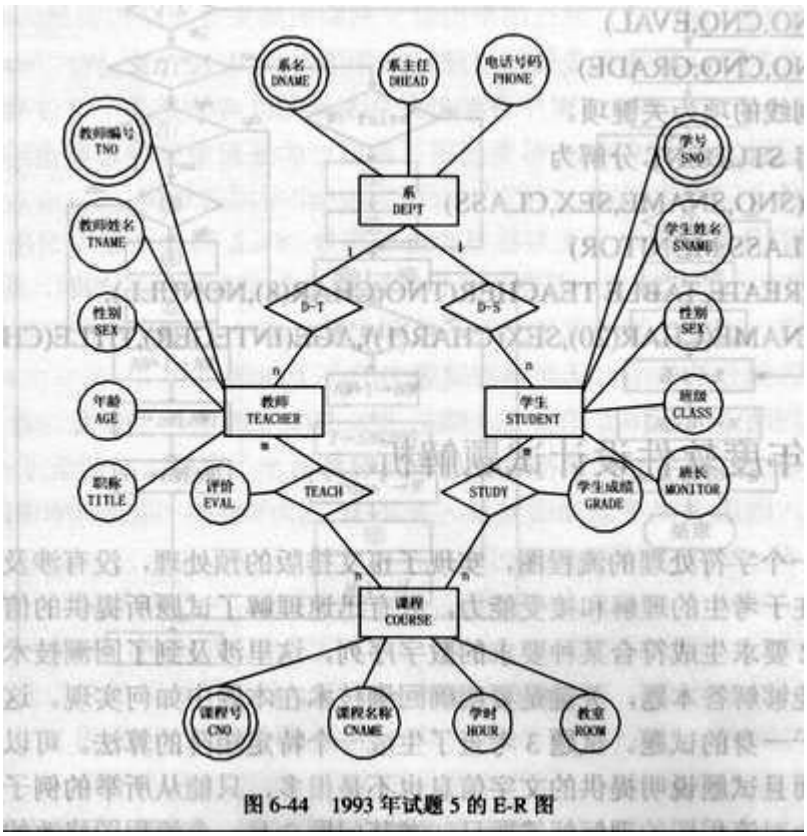


图 6-44　1993 年试题 5 的 E-R 图

［说明］设有下列关于教务管理系统的 E-R 图。图中矩形表示实体，圆表示属性，双圆表示关键字属性，菱形表示实体间的联系。为了答题的方便，图中的实体和属性同时给出院中英文两种名字，回答问题时只须写出英文名即可。

［问题 1］写出与上述 E-R 图对应的关系模式，并且用下划线标明相应的关键字。

［问题 2］问题 1 中的关系模式属于第几范式？如果属于第三范式，则说明理由；如果不属于第三范式，同将它化为第三范式（回答时只需写出修改的部分）。

［问题 3］试用 SQL 语言定义教师（TEACHER）模式。回答时字段的数据类型以及题中未指明的名字由考生自己定义。

【解析】

本题不作详细解析，有关数据库的内容经常在下午试题中出现 。只要对有关数据库的理论知识和 SQL 语言有切实的把握，解答此类题目的成功率应该是很高的。建立 E-R 模型的基本步骤是，确定实体类型和联系类型，据此画出 E-R 图，然后确定实体类型和联系类型的属性，其中联系类型至少应包括与之联系的所有实体类型的关键项。

【答案】

［问题 1］

DPT（DNAME，DHEAD，PHNE）

TEACHER（TNO，TNAME，SEX，AGE，TITE）

STUDENT（SNO，SUAME，SEX，CLASS，MONITOR）

COURSE（CNO，CNAME，HOUR，ROOM）

D-T（DNAME，TNO）

D-S（DNAME，TNO）

TEACH（TNO，CNO，EVAL）

STUDY（SNO，CNO，GRADE）

其中有下划线的项为关键项。

［问题 2］将 STUDENT 分解为
STUDENT（SNO，SUAME，SEX，CLASS）
SCLASS（CLASS，MONITOR）
［问题 3］CREATETABLETEACHER（TNO（CHAR（8），NONULL），
TNAME（CHAR（20），SEX（CHAR（1））AGE（ENTEGER），TITLE（CHAR（20））

5.9 1992 年度软件设计试题解析

试题 1 是一个字符处理的流程图，实现了正文排版的预处理，没有涉及固定的算法，解答问题的关键在于考生的理解和接受能力，只有迅速理解了解试题所提供的信息才能够正确解答问题。试题 2 要求生成符合苛种要求的数字序列，涉及到了回溯技术，但并不是掌握了回溯技术就能够解答本题，关键是要明确回溯技术在本题中如何实现。这是一道集知识考查与能力考查于一身的试题。试题 3 考查了生成一个特定矩阵的算法，可以说，本题没有涉及固定算法，而且试题说明提供的文字信息也不是很多，只能从听举的例子中体会算法要实现的结果，结合对流程图的理解解答题目，尤其问题 2 是一个流程图修改的问题，如果没有对流程图的透彻把握，解答这个问题要大费周折。

试题 1（1992 年试题 1）

阅读下列说明和流程图，如图 6-45 所示，回答问题 1 和 2，把解答填入答卷的对应栏内。

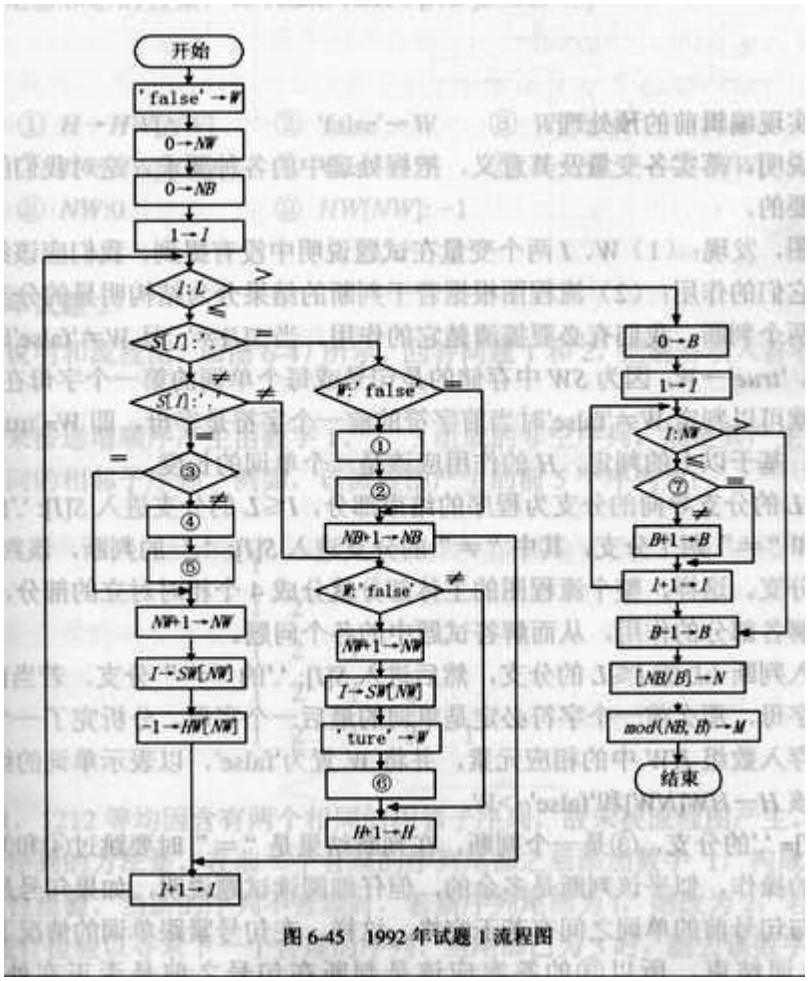


图 6-45　1992 年试题 1 流程图

［说明］

在字符数组 S 中存放着一行和度为 L 的正文，每个数组元素存放一个字符。现假定正文仅由单词、空格和句号组成，单词组由连续的英文字母组成。单词与单词之间可以有 1 个或多个空格，单词至名号之间或句至单之间或名号至单词之间可以有 0 个或 0 个以上的空格，两个名号之间除空格符外至少有一个单词。S 的第一个非空格字符不能是句号，最后一个非空格字符不一定是句号，S 的两端可以有 0 个或 0 个以上的空格。现准备对 S 中的字符串进行编辑，使得空格尽可能均匀地分布在单词这间。

该流程图实现编辑前的预处理。在流程图中，首先统计 S 中单词和句号的总数 NW、空格总数 NB、句号或每个单词的第一个字母在 S 中的位置（存放在数组 SW 中）和每个单词或句号的长度（存放在数组 HW 中，句号的长度置为-1）。然后，计算单词以及句号至单词间的间隔总数 B，最后计算每个间隔的平均空格数 N（整数）和剩余的空格数 M。

[问题 1]

填充流程图的①～⑦框，使之成为完整的流程图。

[问题 2]

为了检查正文中句号的用法是否正确，需要在流程图的（i）处增设判断及报错处理（见图 6-46）。试用题中的有关符号填写判断框⑧和⑨的内容。

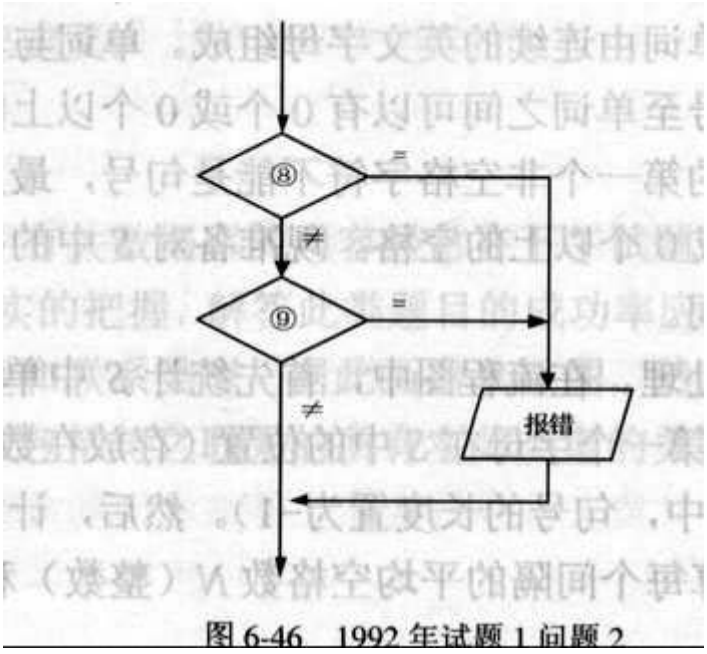


图 6-46　1992 年试题 1 问题 2

【解析】

该流程图实现编辑前的预处理。

阅读试题说明，落实各变量及其意义，把握处理中的种要求，这对我们阅读和理解流程图是极为重要的。

粗读流程图，发现：（1）W、I 两个变量在试题说明中没有提到，我们应该结合对流程图的理解来确定它们的作用；（2）流程图根据若干判断的结果分为结构明显的分支。

W 关系到两个判断，我们有必要搞清楚它的作用。当 且，时，句号数加 ，因为 SW 中存储的是句号或每个单词第的第一个字母在 S 中位置，这样，基本上就可以判定 W 时当前字符有前一个字符是字母，即 W 时程序正在职分析某个单词，基于以上的判定，H 的作用应该是一个单词的长度。

I：L 的 I L 的分支导向的分支为程充的结束部分， 的分支进入 的判断，该判断生"="和"≠"两个分支进入 ，的判断，该判断产生"="和"≠"两个分支。这样，整个流程图的主体部分就分成 4 个相对对立的部分，我们可以结合试题说明理解各部分的作用，从而解答试题中的各

个问题。

我们先进入判断 I：L 的 I≤L 的分支，然后进入 的"="分支。若前字符是"且前一个字符是字母，那么前一个字符必定是单词的最后一个字母。分析完了一个单词，应该将当次的长度存入数组 HW 的相元素，并将 W 置为'false '，以表示单词的结束，所以①和②的答案应该 H HW NW 和 falseW。

再进入 S 的分支，是一个判断，在判断结果"="时要跳过④和⑤，直接执行以下处理句号的操作，似乎该判断是多余的，但仔细阅读试验题说不得明，如果句号是可以紧跟单词的，也可以与句号前的单词之间有若干空格。这样，在句号紧跟单词的情况下，我们还应该处理一个单词结束，所以③是一个判断，在判断结果"="时要跳过④和⑤，直接执行以下处理句号的操作，似乎该判断是多余的，但仔细阅读试题说明，如果句号是可以紧跟单词的，也可以与句号前的单词之间有若干空格。这样，在句号紧跟单词的情况下，我们还应该处理一个单词结束，所以③的答案应该是判断在句号之前是否正在处理一个单词。④和⑤的处理与①、②的处理是相同的，答案也应该相同。

现在进入 的分支，若当前字符不是句号且 W ，这是在处理一个单词，只要在单词的长度上加 1 即可，由此可以推知 H 的作用是记录单词的长度；若当前字符不是句号有 W='false'，说说当前字符是单词的第一个字母，此时应该进行的操作是可以推断而知的，结合⑥后 H+1-H， 我们在⑥处填写 0H。

I＞L 的分支是统计完成以后的工作，即试题说明中的最后两句。各变量的意义已经明确，流程图的任务也经明确。从流程图的最后 3 句可以看出，B 应该是正文单词数减 1。如何得出 B，结合我们对流程图的理解，只有对数组 SW 或 HW 进行统计。两个数组中都有关于句号的记录，且在 SW 中句号的记录与单词起始位置的记录无区别的，但在数组 HW 中句号的记录却有一个特殊值。综合以上分析，我们在⑦中填写"HW -1"。

题中的问题 2 要求在流程图的 I 处增设判断及报错处理。仔细阅读处理说明中关于处理的要求，如下两点应该注意：(1) 正文的第一个非空字符不能是句号；(2) 且两个句号之间除空格外至少有一个单词。我们解答应该以这两点为基础来展。阅读流程图可知 ，句号在数组 HW 中的记录为-1，它在数组 HW 的位置由 NW 指定，NW 初值为 0。所以，当遇到句号，且 NW=0 时，表示该句号是正文的第一个非空字符。若 NW≠0,但 HW =-1，MJ 则表示当前的句号之前，除空格外也是句号，以上的分析函盖了我们提出的两点，用试题中的符号表示就是⑧和⑨的答案，即 NW：0 和 HW［NW］-1。

【答案】

［问题 1］①H→HW［NW］ ②'false' →W ③ W：'false'

④H→HW［NW］ ⑤'false' →W ⑥O→H ⑦HW

［问题 2］⑧NW：0 ⑨HW［NW］：-1

试题 2 （1992 年试验题 2）

阅读下列说明和流程图，如图 6-47 所示，回答问题 1 和 2，把解答案填入答卷的对应栏内。

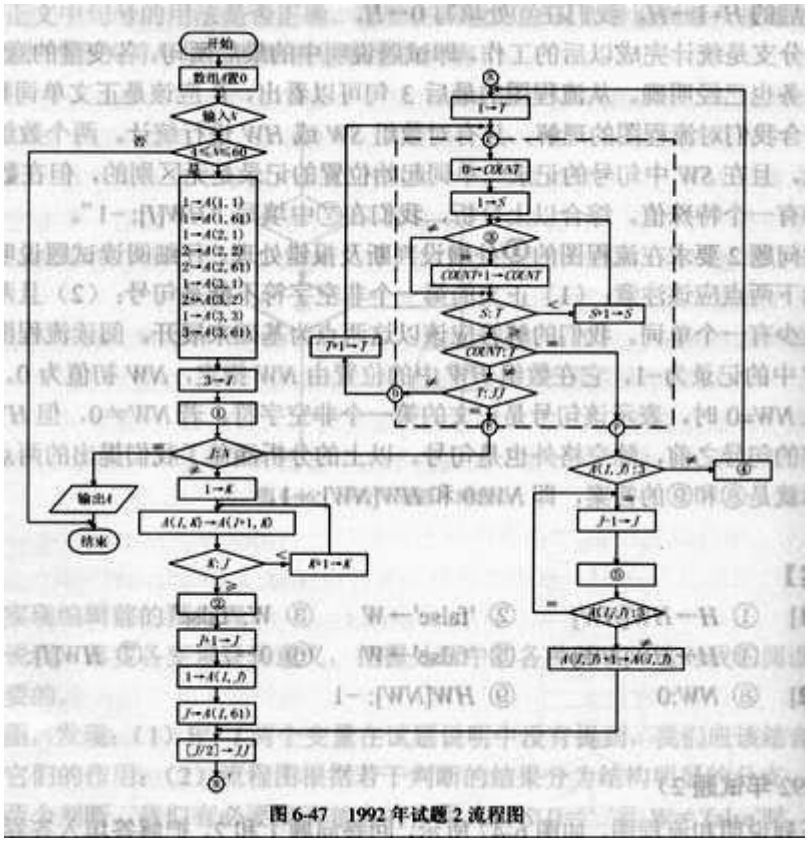


图 6-47　1992 年试题 2 流程图

[说明]

流程图用来按递增顺序产生由数字 1、2、3 组成的非空序列。规定其产生的每个序列为均不存在两个相同的相邻子序列。例如，该程流程图产生的前 5 个序列分：

1

1 2

1 2 1

1 2 1 3

1 2 1 3 1

序列 1211，1212 等均因含有两个相同的相邻子序列，故未被流程图产生。

产生一个序列的方法是，在前一个合理的序列尾部之后添加数字 1，构成一个新序列，然后进行合理性检验。当新的序列不合理时，右新列尾部为 1，则改为 2，若为 2，则改为 3，然后继续进行检验以寻找下一个合理的序列。当尾部已为 3 时，删去该数字后，继续进行上述更改和检验。

本流程图寻找长度大小等 N，（4≤N≤60）的合理序列。二维数组 A 的每一行用于存数找到的一个序列，其中数组元素 A 存放第 1 个序列的长度。

图中[W]表示不超过 W 的最大整数。

［问题 1］

填充流程图中的①～⑤ ，使之成为完整的流程图。

［问题 2］

现对流程图中虚线所围的部分给出了图 6-48 所示的简化形式，试填充其中的⑥～⑧ ，使之成为正确的简化。

【解析】

本题采用回溯技术，按递增顺序产由数字 1、2、3 组成的非空序列，规定其产生的每个序列

均不存在两个相同的相邻子序列。

解答问题不一定要拘泥成法，一个简单的逻辑就是能够快速准确地解答问题才是好的解答方法。在阅读试题说明和流程图后，研究从判断 J：N 到 J-A（I，61）之间的流程图，可以推知在 1 和 2 之间的流程图是从上一个序列生产新序列首部的操作。从判断 K：J 及其两个分支可以推知 J 的含义是一个序列的长度（新序列首部的长度），而此前又没有对 J 赋值操作，在程序中不赋值而使用变量是不符合常规的。结合 J 含义，可知 J 的初始值应该为 3，所以的解答应该是给 J 赋初值 3 的任何语句。利用以上对流程图的理解 K≥J 表示上一人序列已经完全复制为新序列的首部，把直到 JA（I，61）之间的内容与试题说明中的产生一个新序列的方法是……添加数字 1"结合起来理解，不难发现这部分实现在上一个序列后添加数字 1 的操作。结合上下文，该处应该是 I 的递加，否则 J-AN（I，J）将会把新序列中现存的最后一个元素（即上一个序列的最后一元素）覆盖掉。

考查流程图基余的部分，不难发现这是进行合理性检验和回溯处理的。从 到判断 A（I，J）：处，是对新序列进行合理性检验的处理，此判断的 A（I，J）=3 分支到 A（I，J）+1A（i，j）是回溯部分。结合试题说明，当新序列不合时……则改为 3，答案显然是 A（I，J）+1A（I，J）；当尾部已 3 时，则删去该数字后，继续进行上述更改和检验，联系语句 J-1-J 序列的长度减 1，对应的 A（I，61）也该减 1，所以凡是实现 A（I，61）所以凡是实现 A（I，61）减 1 的解答都是合理解答。

现在我们集中精力解答，由以上分析可知，所处的部分进行合理性检验，检验的实质是检测该序列中最否存在两个相同的相邻子序列。由于该序列除最后一个元素以外已经是一个合理的序列，所以，合理性检验也就简化为含有最后一位数字的子序列是否有相同的邻子序列。考虑 意义，如果序列中包含有最后一位数字的不序列是否有相同的相邻子序列。考虑 意义，如果序列中包含相邻且相同的两个字序列，那么其最大长度应该是 JJ，因此可以判断 T 为当前检测的子序列的长度。若 T==JJ 且未发现不合理的子序列，则表示检测完毕，该序列通过合理性检验。逐层深入，可以推知 S 为当前检测的子序列中正在检测的位置。根据以上分析，要判断两个序列相同，就要使对应位置的元素全部相同，由此我们又可以推断 COUN 表示位置元素相同的个数。若 COUNT 与当前检测子序列的长度相同，则证明该序列不合理。应该是判断对应位置的元素是否相同，因此我们填写。

COUNT 是必须的吗？只要有一对对应位置的元素不同，即表示当前检测的子序列合理，把试题给出的简化流程图与删除有关 COUNT 语句框后虚框中的流程图进行比较，不准推知的作用是相同的，判断 极为相似，不妨在两处填写 ST 与 S+1-S。然后我们回过头来推敲上述 3 个答案，上述解答完全能够实现子序列的合理性检验，由此我们回过头来推敲上述 3 个答案，上述解答完全能够实现子序列的合理性检验，由此我们印证解答的正确性。

【答案】

［问题 1］①A(3,16)→J ②I+1 →I ③ A(I,J-T-S+1):A(I,J-S+1)

④A(I,J)+1→A(I,J) ⑤J →A(1,61)

［问题 2］⑥A(I,J-T-S+1):A(I,J-S+1) ⑦S:T ⑧为 S+1→S

试题 3 （1994 年试题 3）

阅读下列说明和流程图，如图 6-49 所示，回答问题 1 和问题 2，把解答填入答卷的对应栏内。

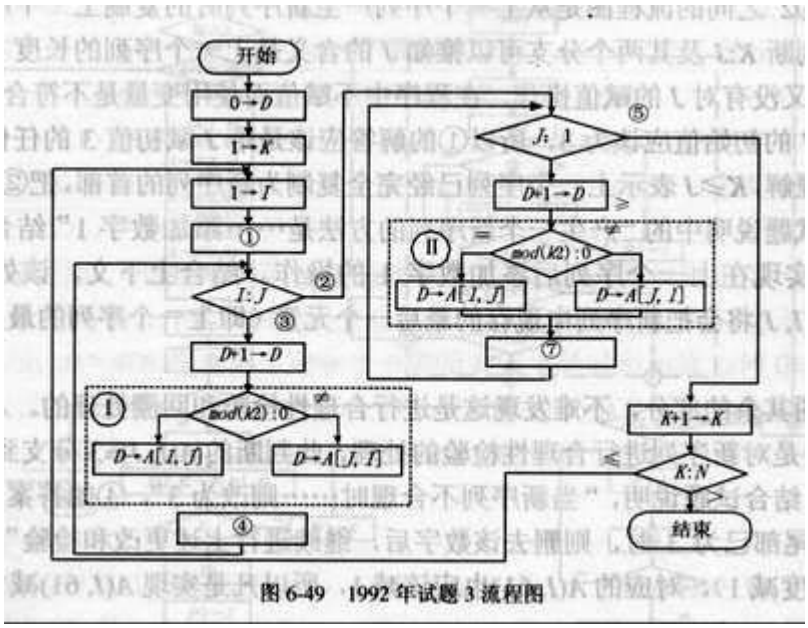


图 6-49　1992 年试题 3 流程图

[说明]

本流程图用来将自然数据按图 6-50 所示的次序依次存放到 N*N 的才维数组 A 中，图中描述了 N=5 时，A 中各元素的值及其赋值次序。

流程图中省略了数据输入和输出。

［问题 1］填充流程图中使之成为完整的流程图。

［问题 2］若将流程图中的两个虚线框部分都改成图 6-51 的处理框，试写出 N=3 时该流程图所产生的数组 A 有各元素值。

图 6-50　1992 年试题 3 说明

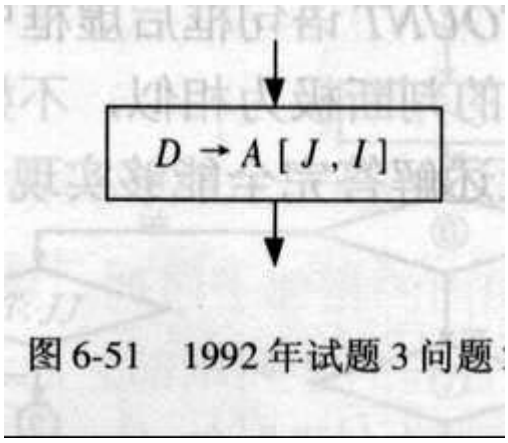


图 6-51　1992 年试题 3 问题

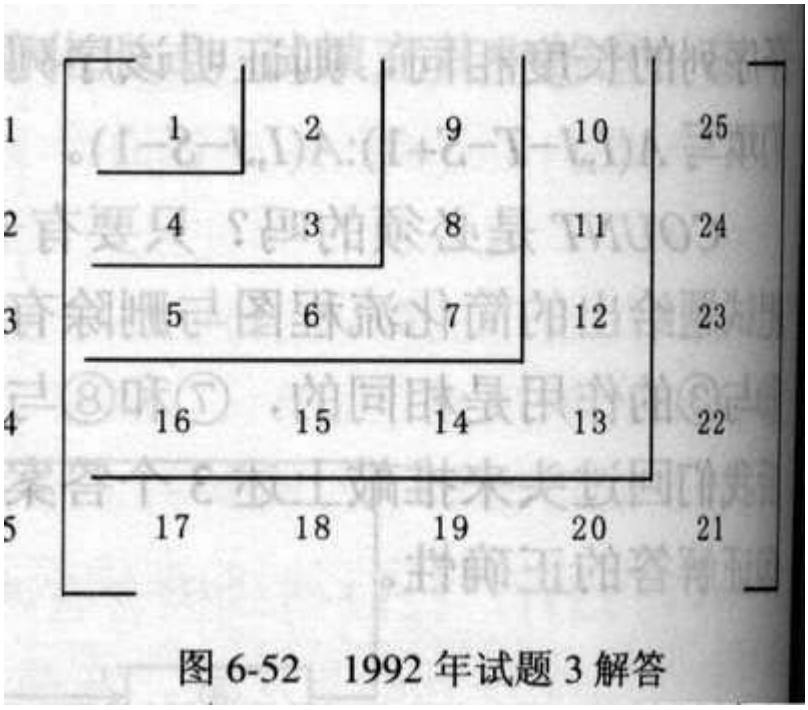


图 6-52　1992 年试题 3 解答

本题是按某特定的次序生成一个 N*N 的二维数组。图 6-52 描述了 N=5 时 5 人区域的划分情况。我们称一个区域为一个直角行。对于 N*N 的二维数组，共有 N 个直角行，其行号如图 6-52 所示，根据题中规定的赋值次序可知，偶数直角行的赋值次序是先赋垂直方向的直角边，后赋水平方向的直角边，我们称它为先列后行。奇数直角行的赋值次序是先赋水平方向的直角边，后赋垂直方向的直角边，我们称它为先行后到的。表 6-9 给出了第 K 直角行的先列后行和先行列的赋值次序。

表 6-9　1992 年试题 3

赋值顺序	1	2	…	k	k+1	k+2	…	2k-1
先列后行	A[1，k]	A[2，k]	…	A[k，k]	A[k，k-1]	A[k，k-2]	…	A[k，1]
先行后列	A[k，1]	A[k，2]	…	A[k，k]	A[k-1，k]	A[k-2，k]	…	A[1，k]

仔细研究试题说明所提供的图，不难发现流程图有 4 点应该注意：(1) 处理至第 1 行时，列号加 1，行号为 1；(2) 处理至第 1 列时，列号为 1：(3) 行号等于列号时，视当前处理情况改变处理方向。(4) 在行号与列号码奇数和偶数时处理方向是不同的。下面我们结合流程图理解以上 4 点是如何实现的。

从判断 I：J 和 JI 作为数组 A 的下标来看，实现了第 4 点，由 I：J 和 I：J 引导的两个循环

实现了第 3 点，两个循环配合实两第 1-2 点。

现在开始阅读流程图。结合的上下文，此前同有对变量 J 的操作，此后使用了 J，所以应该是对 J 的赋值操作，而且该初值应为 1。I 实现了行号列号为 1 的操作，但行号列号的加 1 呢？显然，如果不借助其他变量的支持，此处是不同时承担赋初值和加 1 两项任务的。考查 K 及流程图的末尾两句，不难推知两点：(1) K 在流程图的大循环中递加；(2) K 与行数列数（N）关系密切，由 K 与 N 的比较来判断程序是否结束。结合 K 的初值为 1，我们在处填写 K-J，实现赋初值和加 1 两项任务。

当我们以 1 和 2 地进入系统时，发现两点：(1) 如果 1、2、3 与 5、6 的分支导出情况一致，那么无论是 1-A 还是，I 总是不能进入数组 A，所以我们判断两个判断的分支导出情况应该正好相反。的判断结果不应该是"="和"≠"，否则只有行号等到于列号的数组元素才能被操作。但 1 的操人作应该在第 1 个循环中实现还是在第 2 个循环中实现呢？很明显，如果在第 1 循环中实现，则在完成 1 后修改列号（由试题说明可知），不可避免地又会进入第 2 个循环，因此我们只好反 1 放在第 2 人循环中实现。这样，我们在 5 和 6 中填写或来实现。或放在 6 处是为了保证 1 的实现。J 在 7 中是无法通过合理的修改达到 J 的，详细原因不在此 述。所以我们选择，这样 2 和 3 的解答也就出来了。

现在剩下 4 和 7 没有解答，显然，这里是在修改行号或列号，答案无非是 I 或 J 的加 I 或减 1，但如何确定他们的答案？可以考虑，如果 4 不是 I 的递加，则循环就会陷入死循环；同样的思路我们可以判定的解答是 J 的递减。

题中的问题 2 把图中的两个虚线框改成 D-AJ，根据我们对流程图的理解可知，D 是先行后列的赋值次序，这样修改后的流程图总是按先行后列的次序进行赋值。因此，当 N=3 时流程图所产生的数组 A 的各元素值为 149、238、567。

【答案】

［问题 1］①K→J ②≥ ③< ④I+1→I ⑤< ⑥≥ ⑦J-1→J

［问题 2］149 238 567

试题 4 （1992 年试题 4）

阅读下列说明和流程图 6-53，回答问题 1 至问题 3。

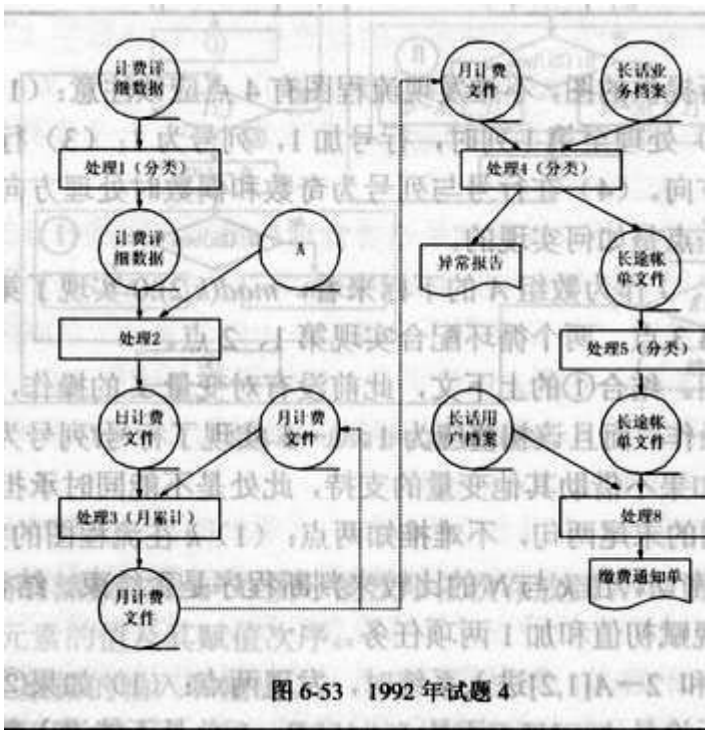


图 6-53　1992 年试题 4

［说明］

（1）流程图描述了某电话局长途电话业务及计费管理系统中的计费数据处理流程。

（2）每个用户在系统的"长话用户档案"中有一个记录，该记录由用户编码唯一标识。一个用户可能拥有多个可使用长途直拨业务的电话号码（称为有权电话号码）。每个有权电话号码在"长话业务档案"中有一个记录。它们分别含有以下数据项。

"长话用户档案"：用户编码、用户名称、用户地址

"长话业务档案"：电话号码、用户编码、国内长途许可标志、国际长途许可标志

（3）计费详细数据由电话号码程控交换机自动记录在磁带上，用作系统的输入。系统每天晚上零时处理计费详细数据磁带，计算通话次数和电话费，产生日计费文件，并把这些数据累计到"月计费文件中"。

"计费详细数据"包括以下数据项：

电话号码、受电话号码、日期、通话开始时间、通话持续时间

系统根据受电话号码可以区分国内长途和国际长途。

（4）系统每月 1 日上午进行出帐处理，给每个月户提交一份上月的包括以下内容的"缴费通知单"。每个有权电话号码每月还需向电话局付"门号费"10 元。

×××电话局长途电话用户缴费通知单

用户编码： 用户名称： 用户地址：

电话号码	国内通话次数	国内金额	国际通话次数	国际金额	门号费	金额合计
…	…	…	…	…	…	…

日期 ××年××月 金额总计

[问题 1]

除了上述说明中指明的文件外，流程图中还需用到文件 A。指出文件 A 应是什么文件？

[问题 2]，

指出“日计费文件”至少应包括哪些数据项？

[问题 3]

①指出处理 5 分类的第一、第二关键项。

②指出“长话业务档案文件”应按哪一个关键项分类？

【解析】

本题描述了长途电话计费的处理流程。

仔细阅读试题说明和流程图，不难发现，处理 2 使用计费详细数据和文件 A 生成日计费文件，日计费文件与月计费文件合并后形成每月的月计费文件。很明显，计费详细数据中只提供了计费的一人参加数，即通话持续时间，根据常识，应该有一个话费单价作为计费参加数。结合流程图的后半部分，可以推知在形成月计费文件之前不需要其他数据文件的参加，所以判定文件 A 是电话费单价文件。

考查日计费文件的数据数源和数据输出流向，发现日计费文件的数据来源是计费详细数据和话费单价文件，日计费文件中的数据应该是可以直接或间接从计费详细数据取得的，日计费文件最后合并入月计费文件，可知两者的文件结构是大体相同的。由下面的月计费文件与长话业务档案相配合生成的缴费通知单的结构可知，月计费文件至少应该包括电话号码、国内通话次数、国内金额、国际通话次数和国际金额，我们就把它当作日计费文件的结构。考查这些数据项，它们都可以直接或间接从计费详细数据取得，应该是合理的。

处理 5 是对长话帐单文件进行分类，分类后要与长话用户档案配合通过处理 8 产生缴费通知单，长话用户档案应该是按用户编码有序的，长话帐单文件是由月计费文件和长话业务档案相配合而产生的。从缴费通知单可以推断两点：（1）缴费通知单按用户编码产生，每张通知

单上有若干个电话号码；(2) 长话帐单文件中应该而且能够包含电话号码和用户编码。为使处理 8 能够获得理想的处理速度和处理效率，用户编码应该是长话帐单文件的第一关键项，电话号码应该是长话帐单文件的第二关键项。

长话业务档案可以电话号码或用户编码为关键项进行分类，但考虑到长话业务档案与月计费文件配合产生长话帐单文件，即使不考虑处理 4 的内容，当两个文件配合参与处理时，使用相同的关键项分类能够使处理获香港理想的处理速度和处一效率，从上面的分析我们可以推知，月计费文件只能以电话号码为关键项进行分类，所以长话业务档案必须电话号码为关键项进行分类。

【答案】

［问题 1］电话费单价文件

［问题 2］电话号码、国内通话次数、国内金额、国际通话次数、国际金额

［问题 3］①第一关键项是用户编码，第二关键项是电话

②电话号码

试题 5（1992 年试题 5）

阅读下列说明和流程图，如图 6－54 所示，回答问题，把解答填入答卷的对应栏内。

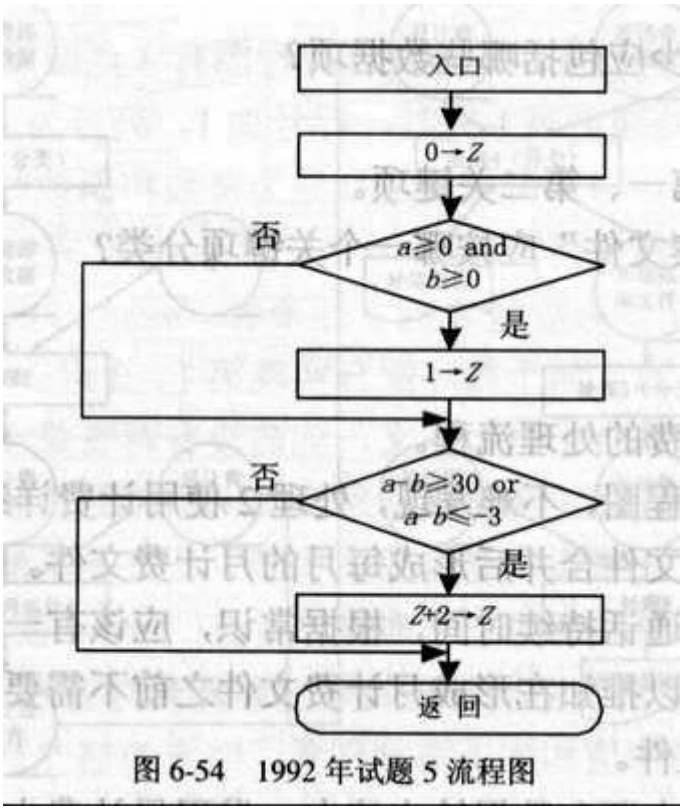


图 6-54　1992 年试题 5 流程图

［说明］

本流程图描述了某子程序的处理流程，现要求用白盒测试法对子程序进行测试。

［问题］

根据判定覆盖、条件覆盖、判定／条件覆盖、多重条件覆盖（条件组合覆盖）和路径覆盖 5 种覆盖标准，从供选择的答案中分别找出满足相应覆盖标准的最小的测试数据组（用①～⑩表示）。

供选择的答案

① a=5 b=1　　②a=5 b=-1

③ a=5 b=1
a=-5 b=-1

④a=5 b=1
a=0 b=-1

⑤ a=5 b=-1
a=-5 b=1
a=-5 b=-1

⑥a=5 b=1
a=0 b=0
a=-5
b=-1

⑦ a=5 b=1
a=0 b=1
a=0 b=-1
a=-5 b=1

⑧a=5 b=1
a=0 b=-1
a=-5 b=1
a=-5
b=-1

⑨ a=5 b=1
a=0 b=-1
a=0 b=1
a=-5 b=1
a=-5 b=-1

⑩a=5 b=1
a=5 b=0
a=5 b=-1
a=0 b=1
a=0 b=0
a=0 b=-1
a=-5 b=1
a=-5 b=0
a=-5
b=-1

【解析】

本题的关键是为该流程图的白盒法测试设计数据，解答本题的基础是对判定覆盖等 5 种覆盖标准的把握。

首先我们列出 5 种覆盖标准的要求。判定覆盖标准是指提供足够的测试用例使和每个判定的每一种可能的结果至少出现一次。条件覆盖标准是指提供足够的测试用例使得每个条件的所有可能的结果至少出现一次，并且每个判定本身的所有可能结果至少出现一次，也就是说，判定条件覆盖是既满足判定覆盖标准，又满足条件覆盖标准。多重条件覆盖标准是指提供足够的测试用，使得每个判定中条伯结果的所有可能组合至少出现一次。

分析流程图，列出流程图中所有的判定和条件，建立真值表（见表 6-10），把各组数据在各个判定或条件上的真或假在表中列出，试题的解答就非常明显了。这也是解答此类题目的一种单方法。我们以 T 表示该组数据在该判定或条件上为真值，F 表示该组数据在该判定或条件上为非真值。

表 6-10　　1992 年试题 5 真值表

测试数据	判定		条件			
	a≥0 and b≥0	a-b≥3 or a-b≤-3	a≥0	b≥0	a-b≥3	a-b≤-3
①	T	T	T	T	T	F
②	F	T	T	F	T	F
③	T/F	T/T	T/F	T/F	T/F	F/T
④	T/F	T/F	T/T	T/F	T/F	F/F
⑤	F/F/F	T/T/T	T/F/F	F/T/F	T/F/F	F/T/T
⑥	T/T/F	T/F/T	T/T/F	T/T/F	T/F/F	F/F/T
⑦	T/T/F/F	T/F/F/T	T/T/T/F	T/T/F/T	T/F/F/F	F/F/F/T
⑧	T/F/F/F	T/F/T/T	T/T/F/F	T/F/T/F	T/F/F/F	F/F/T/T
⑨	T/F/T/F/F	T/F/F/T/T	T/T/T/F/F	T/F/T/T/F	T/F/F/F/F	F/F/F/F/T
⑩	T/T/F/T/T/F/F/F/F	T/T/T/F/F/F/T/T/T	T/T/T/T/T/T/F/F/F	T/T/F/T/T/F/T/F	T/T/T/F/F/F/F/F	F/F/F/F/F/F/T/T/T

结合以上关于覆盖标准的叙述和真值表，我们可以解答满足判定覆盖、条件覆盖、判定/条件覆盖和多重条件覆盖 4 种标准要求的测试数据。

路径覆盖标准是指提供足够的测试用例，使每条路径都至少执行一次。本题中共有 4 条路径，因此至少应有 4 个测试数据。在 8 和 7 之间比较，不难分析出测试数据 7 满足路径覆盖标准。

【答案】

判定覆盖 ④

条件覆盖 ③

判定条件覆盖 ⑥

多重条件覆盖 ⑧

路径覆盖 ⑦

5.10 1991 年度软件设计试题解析

试题 1 是对一个递归函数的实现，试题 2 是关于相乘的问题，相对来说，试题 1 的难度要低一些，至少要实现的算法是明确的。试题 1 的难度要低一些，至少要实现的算法是明确的。试题 2 则不同，解答题目首先必须明确流程图所采用的算法，然后在此基础上解答试题，这对考生的能力要求较高。

试题 3 是关于分层数据流图的分析和修改。这类试题只要贯彻数据平衡原则，把握各层流程图之间的对应关系，解答难度就会相对低些。

本年度软件设计试题较有特色。试题 4 的问题 3 设计巧妙，要求考生在掌握流程图的基础上彻底修改流程图，以实现另外的功能；试题 5 的问题 2 也很特殊，要求考生修改文件的数据项以解决一个在流程图中没有解决的问题。

试题 1　（1991 年试题 1）

阅读下列说明和流程图 6-55，回答问题 1 和问题 2，把解答填入答卷的对应栏内。

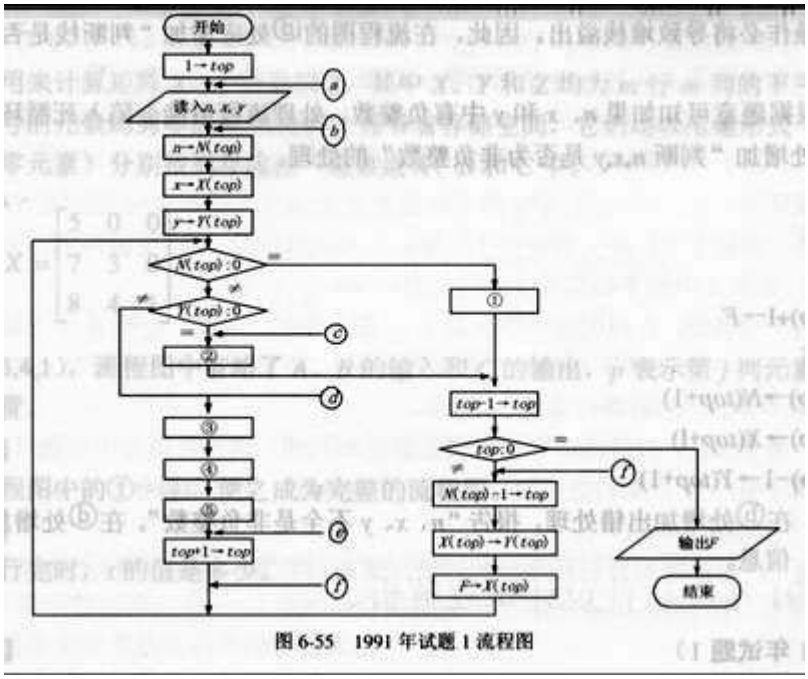


图 6-55　1991 年试题 1 流程图

［说明］

流程图实现下列递归函数的计算。递归函数定义如下：

A(n,x,y)=

x+1　　当 n=0

1　　当 n≠0 且 y=0 时

A(n 1,,A(n,x,y-1),x)　　当 n≠0 且 y=0 时

其中 n,x,和 y 均为非负整数。

递归函数的计算使用了 3 个栈，它们分别用数组来存放，为栈顶指针，中间计算结果和最终计算结果均存放在变量 F 中。

［问题 1］

填充流程图中的① ～⑤，使之成为完整的流程图

［问题 2］

指出应在流程图的哪些位置（用 a～g 表示）上增加检测错误的流程图，并分别指出这些位置能报告哪些错误信息。

【解析】

该题实现递归函数的计算，计算中使用了堆栈。堆栈 N、X 和 Y 分别用来存放函数的自变量 n、x、和 y。栈顶元素 N(top)、X(top)和 Y(top)存放的是当前正在计算的函数的自变量。

本题的流程图是对递归函数的实现，我们在解答时要紧密结合递归函数的算法，按照递归函数的的算法来确定各个需要解答的问题，所以解答之前要仔细研究递归函数的算法，为正确理解流程图打下基础。

考查流程图的结构，结合递归函数的定义，可知当 n=0 时函数值为 x+1，当 n≠0，且 y=0 时函数值为 1。流程图中的 1 和 2 应该是对此的实现。所以 1 和 2 的答案分别为 X(top)+1-F 和 1-F。

当 n≠0 且 y=0 时，函数值为 A(n-1,A(n,x,y-1)，x)的值，此时应先计算 A(n,x,y-1)的值 F，然后计算 A(n-1,F,x)即当 n≠0 且 y≠0 时需调用递归函烽 A 两次。第一次调用时，自变量 n、x 和 y-1 需要时栈。当计算出 A （n,x,y-1）的值后。栈顶元素需退栈。第二次调用 A 时，可把此时的栈顶元改为 F 和 x。当 A（n-1,F,x）计算完再时把栈顶的自变量退栈。根据以上分

析，流程图的 3、4、5 应分别为 N(top) N(top)、X(top)、X(top+1)和 Y（top）-1Y(top+1)。
在堆栈操作中，比较常用的操作就是在进栈之前判断堆栈是否已满。在堆栈已满的情况下进行栈操作必将导致栈溢出。因此，在流程图的 d 处应增加"判断栈是否已满"的处理。
此外，根据题意可知如果 n、 x 和 y 中有负整数，处理流程可能会陷入死循环，因此应在流程的 b 处增加判断 n，x，y 是否为非负整数的处理。
[答案]
［问题 1］
①X(top)+1→F
②1→F
③N(top)→N(top+1)
④X(top)→X(top)+1
⑤Y(top)-1→Y(top+1)
［问题 2］在(b)处增加出错处理，报告“n,x,y 不全是非负整数”。在(d)处增加出错处理，报告“栈满”信息。

试题 2 （1991 年试题 1）

阅读下列说明和流程图，如图 6－56 所示，回答问题 1 和问题 2，把解答填入答卷的对应栏内。

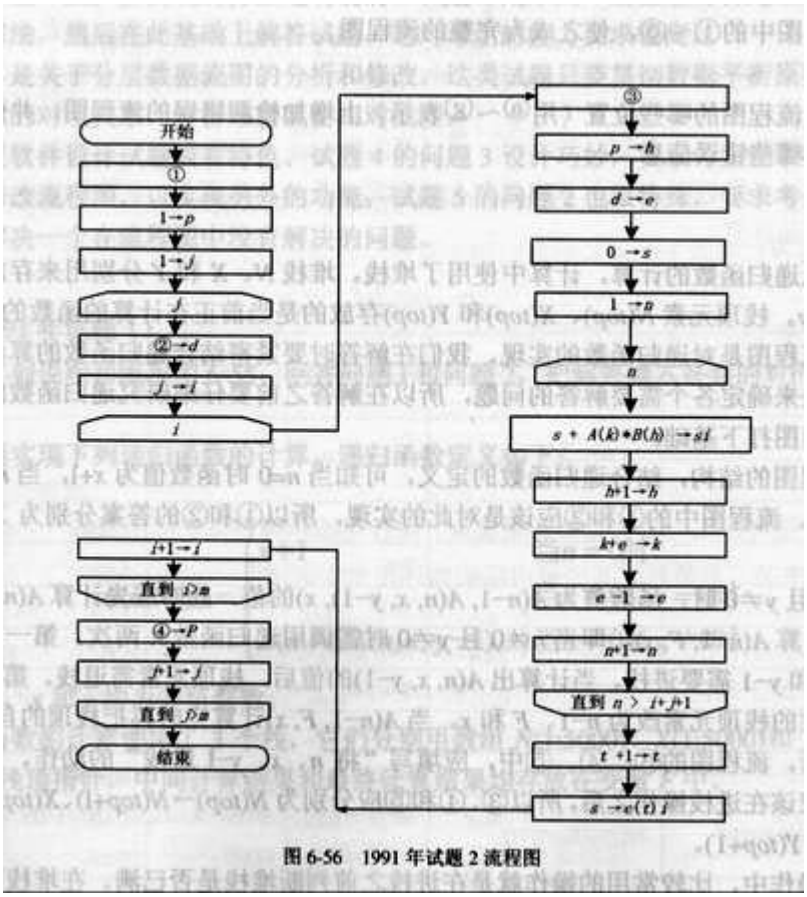


图 6-56　1991 年试题 2 流程图

［说明］
流程图用来计算矩阵 X、Y 的乘积 Z，其中 X、Y 的均为 m 行 m 列的下三角方阵。即行号小于列号的元素均为零的 m 阶方阵。为节省存储空间，它们均以压缩形式（不存放矩阵上三角中的零元素）分别按列存放在一维数组 A、B 和 C 中。

例：若 $X=\begin{bmatrix}5 & 0 & 0\\ 7 & 3 & 0\\ 8 & 4 & 1\end{bmatrix}$

则 A=（5，7，8，3，4，1）流程图中省略了 A、B 的输入和 C 的输出，p 表示第 j 列元素在一维数组中的起始位置。

［问题 1］

填充流程图中的 1、4，便之成为完整的流程图。

［问题 2］

程序执行完时，t 的值是多少。

[解析]

解答本题需要一定的数学知识，并结合流程图的结构，考查注流程图中矩阵相乘算法的实现，两方面要融会贯通，否则很难解答此类题目。

本题是计算两个下三角方阵 X、Y 的乘积 Z，其中 X、Y 和 Z 都有以压缩形式按列存储在数组 A、B 和 C 中。首先，求未经压缩的两个下三角方阵 X、Y 的乘积 Z，其计算公式为：

$$Z_{ij}=\sum_{r=1}^{m} Xir * Yrj \quad (i,j=1,2,\ldots,m)\ (1)$$

由于 Z 也是一个下三角方阵，在压缩存储的情况下只需计算 Z 的下三角部分。因此，公式（1）可变化为：

$$Z_{ij}=\sum_{r=1}^{m} Xir * Yrj \quad (j=1,2,\ldots,m;\ i=j,j+1,\ldots,m)\ (2)$$

由于 X 和 Y 均为下三角方阵，即当 i< r 时，Xir=0; 当 i<j 时，Yrj=0 ,所以公式（2）又可变化成：

$$Z_{ij}=\sum_{r=1}^{m} Xir * Yrj \quad (j=1,2,\ldots,m;\ i=j,j+1,\ldots,m)\ (3)$$

公式（3）现可以用一个三重循环来实现。结合对流程图整体结构的理解，它们应该分别对应于流程图中的三重循环，其中最内层的循环用来实现 $\sum_{r=1}^{m} Xir * Yrj$ 的计算。

明确了循环的作用，解答本题的关键就集中在如何根据循环变理找出 Xir 和 Yir 在下三角方阵 A 和 B 的正确位置。由中层循环中的 t+1→t 和 s→c(t)可以推知，公式（3）的计算顺序应该是首先计算第一列的元素，再计算第二列的元素…据此，t 的初值应为 0。

显然，下三角方阵 X 或 Y 中第 1 行第 j 列的元素在 A 或者 B 中的下标应该是下面的数据列：m,m-1,…m-j+1。这样，下三角方阵 X 或 Y 中第 I 行第 j 列的元素在 A 或者 B 的下标应该是 m+(m-1)+…(m-j+2)+i-j+1。简写这个算式，应该是 j(2m-j+1)/2+i-m.

结合对流程图结构和计算方法的分析，可以推知，在最外层循环中，当 j =1 时，公式（3）中 Yrj 为 y11，它在 B 中的下标为 1。由上述的下标计算公式可以得出如一结论：j 每增加 1，在 B 中的下标值应增加 m-j+1，流程图中的④→p 应该是对下标的修改，因此④应为 p+m-j+1.

在中层循环中，Yr,j 是从 Yjj 开始,与 p→h 相对应。在内层循环中，$Y^{r+1,i}$ 与 Yrj 在 B 的下标值之差为 1，与 h+1→h 相对应。

在内层循环中，Xi,r+1 与 Xir 在 A 中的下标值之差为 m-r,与 k+e→k 相对应，其中 e 即为 m-r

的值。当 n 增加 1 时，m-r 就减少 1，当 e-1→e 相当应，在中层循环中，公式（3）的 xir 从 xij 开始，它在 A 中的下标值为 j(2m-j+1)/2+I-m。

从流程图中可看到，在内层循环中 p 表示 yij 在 B 中的下标值，xjj 在 A 中下标的表示应该是相同的，因此 xjj 在 A 中的下标值也表示为 p+I-j>k。

对于同一个 j 值，内层循环中 e 的初值都是相同的，这也可以由中层循环中的 d→e 推知。

在外层循环中，为 d 赋初值是我们必须解答的问题，从上面的分析可知，d 中存放的是 m-j，所以我们在②处填入 m-j.

解答问题 2，可以通过分析程序的执行情况来求得答案。在程序执行完时，t 应该是 zmm 在 C 中的下标值，所以 t=m+(m-1)+…+1=m(m+1)/2。

[答案]

[问题 1]

①0→t ②m-j

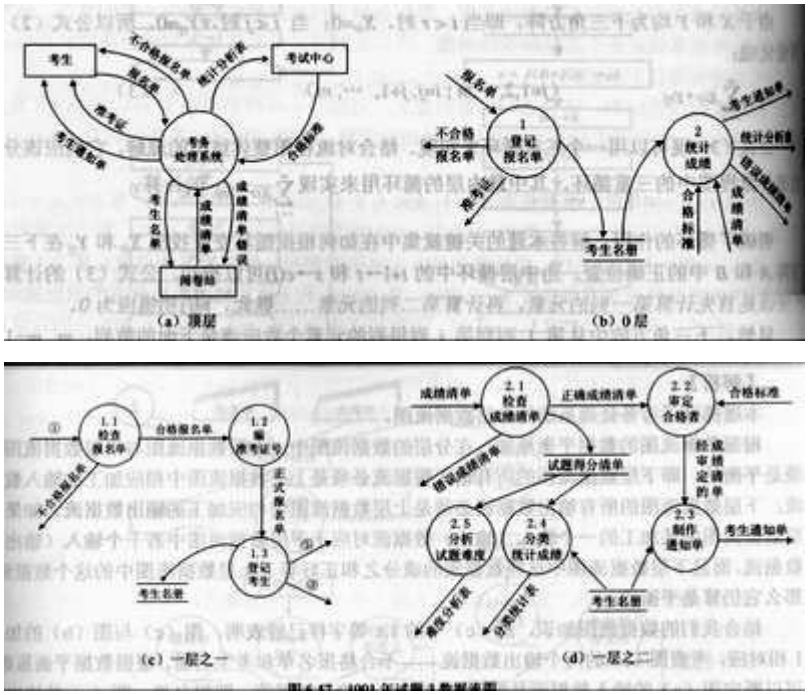
③p+I-h→k 或 j*(2m-j+1)/2+I-m→k 或 (j-1)(2n-h+2)/2+I-j+1→k

④p+m-j+1 或 j*(2m-j+1)/2+1

[问题 2] m(m+1)/2

试题 3 （1991 年试题 3）

阅读下列说明和流程图，如图 6-37 所示，回答问题 1 至问题 4，把解答填入答卷的对应栏内。



[说明]

流程图是采用结构化分析方法画出的某考务处系统的数据流程图（DFD），图中圆圈表示加工：→表示数据流；□表示数据源终点；〓表示文件。该系统有如下功能；

（1）对考生送来报名单进行检查。

（2）对合格的报名单编好准考证号后将准考证送给考生，并将汇总后的考生名单送给阅卷站。

（3）对阅卷站送来的成绩清单进行检查，并根据考试中心制订的合格标准审定合格者。

（4）制作考生通知单送给考生。
（5）进行成绩分类统计（按地区、年龄、文化程度、职业和考试级别等分类）和试题难度分析，产生统计分析表。
部分数据流的组成如下所示：
报名单=地区+序号+姓名+性别+年龄+文化程序+职业+考试级别+通信地址
正式报名单报名单+准考证号
准考证=地区+序号+姓名+准考证号+考试级别
考生名单=（准考证号+考试级别）（其中{w}表示 w 重复多次）
统计分析表=分类统计表+难度分析表
考生通知单=考试级别+准考证号+姓名+合格标志+通信地址
[问题 1]
指出一层之一（图(c)）的数据流图中①、②、③的数据流名。
[问题 2]
指出 0 层（图(b)）的数据流图中有什么成分可删去。
[问题 3]
指出一层之二（图(d)）的数据流图中在哪些位置遗漏哪些数据流，也就是说，要求给出漏掉了哪个加工的输入或输出数据流的名字。例如，加工 2。5 的输出数据流"难度分析表"。
[问题 4]
指出考生名册文件的记录至少包括哪些内容。
[解析]
本题描述了考务处理系统的分层数据流图。
根据数据流图的数据平衡原则，在分层的数据流力部，上层数据流图与下层数据流图必须是平衡的，即下层数据流图的所有输出数据流必须是上层数据流图中相庆加工厂的输出数据流。如果上层数据流力部某加工的一个输入（输出）数据流对庆于下层数据流图中若干个输入（输出）数据流，而且下层数据流图中这些数据流的成分之和正好等于上层数据流力部的这个数据流，那么它仍算是平衡的。
结合我们数据流图知识，图（c）中的 1。X 等字样已经表明，图（c）与图（b）的加工 1 相对应，考查图(c)的两个输出数据流--不合格报名单和考生名册，根据数据平衡原则，可以断定图（c）的输入数据流是图（b）加工 1 的输入数据流，即报名单。图（c）的输出数据流应该与图(b)加工 1 的输出数据流等价，所以图(c)的输出数据流②和输出数据流③应该是准考证和考生名单。
上面对图（b）加工 1 已经进行细致的分析，未发现可删除的成分。考查图（b）加工 2 的输入数据流和输出数据流，发现试题得分清单并不是系统功能所要求的，虽然在图（d）加工 2。1 之后用来产生难度分析表和分类统计表，但只是在加工时使用试题得分清单，完全可以从加工 2。1 之后用来产生难度分析表和分类统计表。由此我们断定图（ b）的输出数据流试题得分清单是可以删除的。
显然，图（ d）是与图（b）的加工 2 相对应的。根据数据平衡原则，考查图(b)加工 2 的输入数据汉和输出数据流，发现图(d)中缺少输入数据流合格标准和输出数据流错误成绩单。仔细考查图（d)，易知输出数据流错误成绩清单应该从加工 2。1 流出，而输入数据流合格标准应该流入加工 2。。
仔细阅读数据流力图，可知考生名册文件的数据源是正式报名单，并在加工 2。3 中产生考生通知单，在加工 2。4 中产生分类统计表。这样，考生名册文件数据项的来源和应用范围都已确定。结合试题说明，首先将考生通知单中除合格标志外的数据项都包括时考生名册文件。成绩要按地区、年龄、文化程度、职业和考试级别分类统计，这些数据项都在（正式）

报名单中，而加 2。4 又没有使用（正式）报史单，显然，以上 5 个数据英也要包括进考生名册文件。

[答案]

[问题 1]①报名单元 ②准考证 ③考生名单

[问题 2]文件试题"得分清单"可删除

[问题 3] 加工 2.1 遗漏输出数据流"错误成绩清单"，加工 2。2 遗漏输入数据流"合格标准"

[问题 4]

考生名册=地区+姓名+年龄+文化程度+职业+考试级别+通信地址+准考证号

试题 4

阅读下列说明和流程图，如图 6-58 所示，回答问题 1 至问题 3，把解答填入答案的对应栏内。

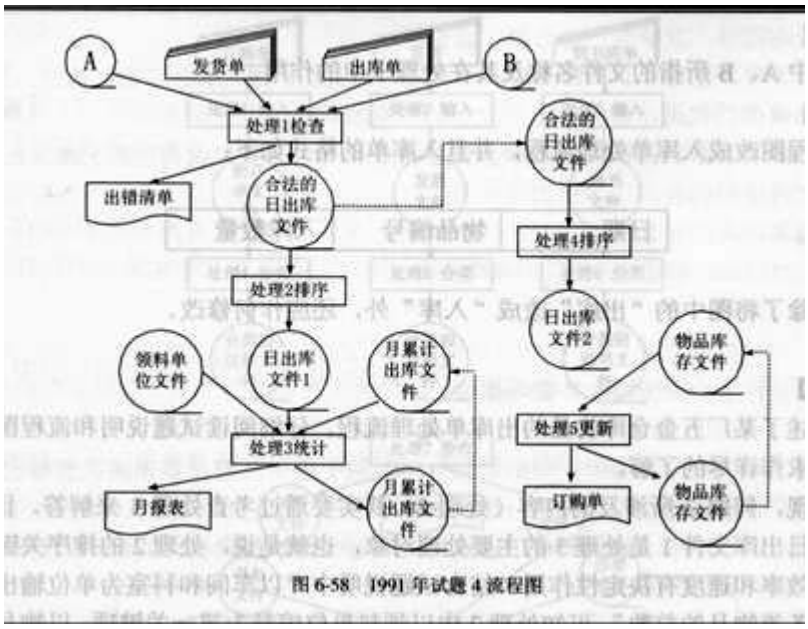


图 6-58　1991 年试题 4 流程图

[说明]

流程图描述了某厂五金仓库物品的出库存单处理流程。该厂有 50 个车间和科室，五金仓库内有近万种物品。仓库管理人员凭入库单收进物品，凭出库单发出物品。系统要求每天根据出库存单更新物品库存文件。当某物品更新后的库存量低于该物品的最低库存量时，输出订购清单。每月还要以车间和科室为单位输出该部门一个月内领取各类物品的总数（称为月报表）

规定每张出库单由两位操作员分别输入，以检查操作员的输入错误。

物品库存文件按物品编号排序，其格式如下：

物品编号	名称	规格	库存量	最低库存量	最高库存量

其中"最高库存量"是指该物品允许存放在库中最大值。

领料单位文件按领料单位编号排序，其格式如下：

领料单位编号	领料单位名称

出库格式如下：

日期	物品编号	出库数量	领料单位编号

[问题 1]

指出处理 2 排序时的第一关键项和第二关键项。

[问题 2]

指出图中 A、B 所指的文件名称及其在处理 1 中作用。

[问题 3]

若把流程图改成入库单处理流程，并且入库单的格式如下：

日期	物品编号	入库数量

那么，除了将图中的"出库"改成"入库"外，还应作任何修改。

[解析]

本题描述了某厂五金仓库物品的出库单处理流程。仔细阅读试题说胆和流程图，对系统的功能和要求作详尽的了解。

不难发生，问题 2 所涉及的内容（处理 2）其实要通过考查处理 3 来解答，因为处理 2 排序产生的日出库文件 1 是处理 3 的主要处理对象，也就是说，处理 2 的排序关键项的设置来处理 3 的效率和速度有决定性作用。结合试题说明中"以车间和科室为单位输出该部门一个月内领取各类物品的总数"，可知处理 2 中以领料单位编号为第一关键项，以物品编号为第二关键项符合处理 3 的要求，能够提高处理 3 的处理速度和处理效率。

处理 1 要对出库单进行检查，为保证数据的正确性，应该对出库存单的每一数据项时行检查，日期的检查是很容易的。但物品编号、出库数量、领料单位编号合法性，必须使用物品库存文件，要难出库存数量、领料单位编号如何检查，显然应该借助文件 A 和文件 B 进行，考查试题说明所给出的文件结构，要验证物品编号的合法性，必须使用物品库存条件，要验证出库数量合法性（如是否大于库存量），必须使用物品库存文件，要验证领料单位编号的合法性，必须使用领料单位文件。综合以上分析，文件 A 和文件 B 所指的对象应该是领料单位文件和物品库存文件。

如果把本流程图改成入库存单处理流程，由于入库过程不中不存在领料行为，所以有关领料和领料单位的操作都应该删除或改成入库操作。处理 3 可以改成月入库统计、报表。另外，处理 5 产生 的订购单显然是不应该在入库过程中出现的。考试到试题说明中"最高库存量"是指该物品允许存放在库中最大值。应该将此处改成超过是高库存量物品报告。

[答案]

[问题 1]第一关键项："领料单位编号"；第二关键项"物品编号"。

[问题 2]A 所指的是领料单位文件，其作用是检查出库单中是否有非法物品编号和非法出库数量（例如出库量>库存量）。

[问题 3]删除图中所有领料单位文件，并把处理 5 输出的订购单改为"报告超过最高库存量信息"。

试题 5 （1991 年试题 5）

阅读下列说明和流程图，如图 6-59 所示，回答问题 1 至问题 3。

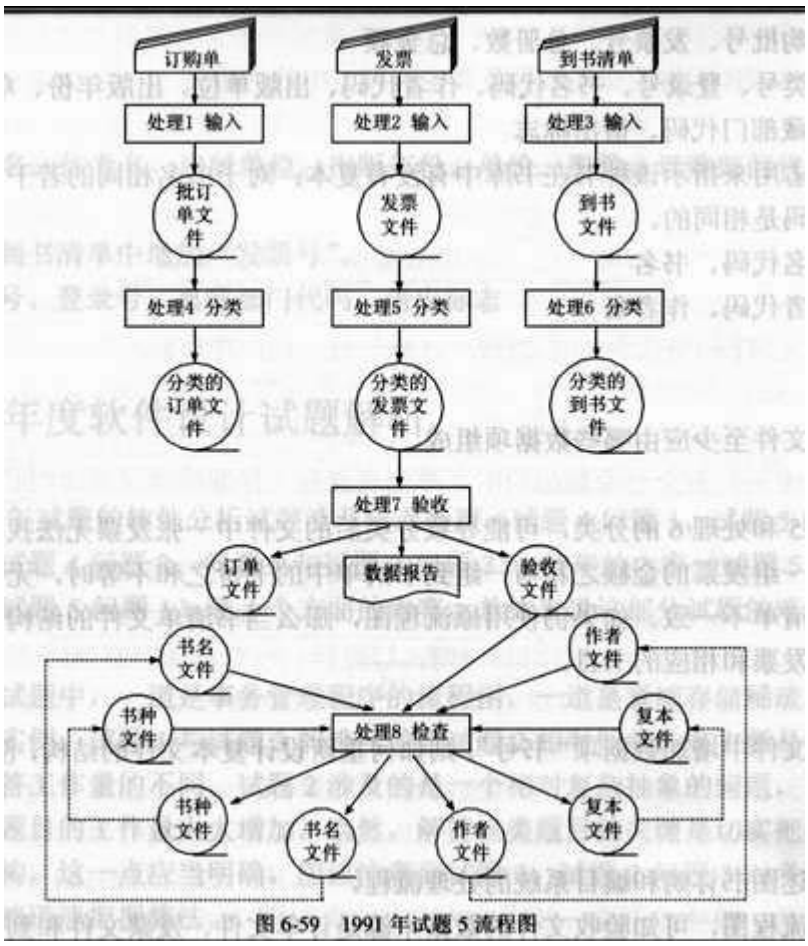


图 6-59　1991 年试题 5 流程图

[说明]

流程图描述了某高校图书订购与编目系统的处理流程。全校的图书典藏在校图书馆和各系的资料室中。学校每年分若干批向出版单位订购图书，同一批订购的图书将陆续邮寄到学校。出版单位在寄出图书的同时附上到书清单和发票，发票上仅给出一份到书清单中书的总册数和总金额。学校收到图书和发票后，先参照订购单验收，然后进行编目，并把有关信息存放在书种文件、书名文件、作者文件和复本文件中，以供读者检索。

书种文件记录了每种书的有关信息。所谓一种书是指同一作者、同一书名、同一出版单位和同一出版年份出版的书。例如，1990 年王某在高等教育出版社出版了《高等数学》（印数 8000 册）和《高等数学》（印数 5000 册），则王某在 1990 年出版了两种书。在全校的藏书中，如果一种书只有一册，则该书的信息存放在书种条件中；如果一种书有多册，则其中一册书的信息存放在书种文件中，其余的书作为复本将信息存放在复本文件中。复本文件的结构一与书种文件的结构相同。每种书都有一个记号，书号唯一地标识了一种书。在书库中，每册书有一个登陆号，登陆号唯一地标识了一册书。此外，为了图书检索的方便，将图书按学科分类，分类号用来标识不同的学科领域。

各类单据和文件的结构如下：

订购单：订购批号、书名、作者名、出版单位、出版年份、单价、订购册数、订购部门代码、订购日期

到书清单：订购批号、书名、作者名、出版单位、出版年份、单价、册数

发票：订购批号、发票号、总册数、总金额

书种文件：分类号、登录号、书名代码、出版单位、出版年份、单价、复本标志、典藏部门代码、借出标志。

其中，复本标志用来指示该种书在书库中没有复本；对于书名不同若干种书，书名的代码是相同的。
书名文件：书名代码、书名
作者文件：作者代码、作者名
[问题 1]
指出验收文件至少应由哪此数据项组成。
[问题 2]
由于处理 5 和处理 6 的分类,可能导致分类在后的文件中一张发票无法找到与定它对应的那些书,从而当一组发票的金额之和与一组到书清单中的书价之和不等时,无法知道是哪一张发票和哪一份清单不一致。如果仍使用原流程图，那么当书清单文件的结构作何改动后，能找出不一致的发票和相应的书目。
[问题 3]
若在书种文件中增加数据"书号"，则如何重新设计复本文件的结构，使数据冗余最小。
【解析】
本试题描述图书订购和编目系统的处理流程。
仔细研究流程图，可知验收文件的数据来源是订单文件、发票文件和到书文件，因此可以确定验收文件数据项的最大范围上述 3 个文件数据项的并集。考查处理 8，发现除了验收文件外，处理 8 更新了书种文件、书名文件、作者文件和复本文件 4 个文件，因此凡是处理 8 所需而又不包含在上述 4 个文件中的数据项都应该存放在验收文件中。显然，像订购批号、订购日期、总册数和金额等数据项对处理 8 是没有意义的，不应该包含在验收文件中。至于分类号、登录号、书名代码和作者单位等数据项，我们可以设想，如果验收文件中已经包含了这些数据项，是否存在编目系统的虚拟化问题，即编目完全融合在订购工作中，这是验收文件至少应该包含书名、作者名、出版单位、出版年份、单价、册数、订购部门/典藏部门代码等数据项。
阅读试题说明并结合流程图，可知出版单位在邮寄图书的同时附上到书清单和发票，发票上仅给出一份到书清单书的总册数和总金额。在处理 5 和处理 6 之后，同一张到书清单上的书必须分散在分类的到书文件中的不同位置上，可能会发生找不到与某一张发票相对应的图书的情况，从面导致发票上的金额与到书清单中的书价之和不一致时无法查对的后果。如果在到书清单中增加数据项"发票号"，则处理 7 中就能很容易地根据发票号找出与之相对应地所有图书。
根据试题说明，可知每种书都有一个书号，书号唯一地表示了一种图书。一种图书的不同复本的确良著录信息是相同的，但登录号对于每一册图书是唯一的，典藏部门代码和借出标志也是针对某一些册图书而有所不同的。所以，在书种文件中增加数据项"书号"后，复本文件中关于该种图书的出版发行信息及分类号、书名代码、作者代码等的数据项可以删除而不影响系统的功能。
【答案】
［问题 1］　书名、作者名、出版单位、出版年份、单价、册数、订购部门代码（或典藏部门代码）
［问题 2］在到书清单中增加"发票号"
［问题 3］书号、登录号、典藏部门代码、借出标志

5.11 1990 年度软件设计试题解析

1990 年度下午试题的软件分析试题涉及错误处理（试题 4 问题 1、试题 5 问题 1）、流

程图修改及评估（试题 4 问题 2、问题 5 与试题 3 题问题 3)、处理的内容（试题 5 问题 2）和文件数据项构成（试题 5 问题 1）等 4 个方面的内容。总体来说这部分试题的难度较低，侧重能力才查。

3 道流程图试题中，一道是事务管理程序的流程图，一道是紧缩存储稀疏矩阵的转置，一道是探测法的实例。试题 1 与试题 3 的难度较之试题 2 相对低些，这主要是因为它们题材类型的差异与解答工作量的不同。试题 2 涉及的是一个相对复杂抽象的问题，而且提供了两种算法，使解答题目的工作量大大增加。当然，解答这类题目的关键是切实把握试题说明和流程图的整体结构，这一点应当明确，应该注意到试题 3、度题 2 问题 3 对考生能力要求较高，要求考生能够迅速把握算法。

试题 1 （1990 年试题）

阅读下列说明和流程图，如图 6-60 所示，回答问题 1 至问题 3，把解答填入答卷的对应栏内。

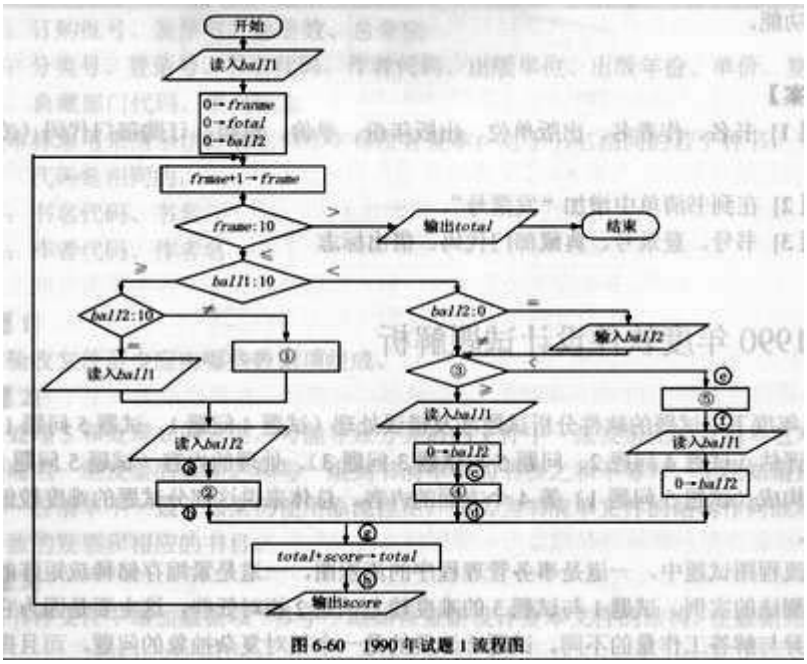


图 6-60　1990 年试题 1 流程图

[说明]

有一种游戏，是用滚球击 10 个性，比赛分 10 局，每局可滚球 1 次或多次，其规则和记分布方法如下：

（1）若一局的第一个球击倒全部 10 个柱（称为 strike)，则这局不再滚球（对球 10 局来说，还可补滚两次球)，其得分为 N 加下两次滚球所击倒的柱数。

（2）若一局数的第一个球未击倒 10 个柱，则可对剩下的柱再滚一次球。如果这局的两闪滚球击倒全部 10 个球（称为 spare)，则这局不再滚球（对第 10 局来说，还可补滚一次球)，其得分为 10 加下下一次滚球所击倒的柱数；否则，这局也不再滚球，其得分为本局两次滚球所击倒的柱数之和。

（3）总得分为 10 局得分之和。

流程图 6-60 读入每球击倒的倒数，计算产并输出每局得分 score 及总分 total 。图中 ball 和 ball2 分别存放每局第一个球和第二个球（如有的话）所击倒的柱数， frame 用于对局计数。

［问题 1］

填充流程图 6－60 中的圆圈，使之成为完整的流程图。

［问题 2］

若要把每球击倒的柱数记录在一个一维数组中，这个数组最少要有几个元素，最多要有几个元素。

［问题 3］

若计算每局得分的规则增加一条，当前面各局累积得分。超过 100 分时，每取得一次 strike 奖励 5 分，那么流程图 6-61 应插在流程图 6-60 中的(a)～(h)哪一个位置上。

[解析]

仔细阅读试题说明，不难发现流程图中有两个重要的判断应该注意。一是判断某局的第一次滚球结果是否为 strike ，即本次滚球击倒全部 10 个柱，二是在第一次滚球结果非 stike 时判断某局的前两次滚球是否将全部 10 个柱击倒。这两个判断在记分中有重要作用。

粗读流程图，会很快发现第一次滚球结果是否为 strike 的判断，而且流程图在此处分成两个分支，左边的分支处理第一次滚球结果为 strike 的情况，对应试题说明的（1）：右边的分支处理第一次滚球结果不是 srtike 的情况，对应试题说明的（2）。两个分支的会合处是 total+score→total 可以大致地判断这是将该局得分记入总分。

我们先看流程图右边的分支，该分支处理 strike 情况，从 ball 为第一次滚球得分来看，baal12 也应该是滚球得分。 判断 ball2 是否为 0 的作用何在？两个分支中都有这个判断，都是在 ball2 为 0 的时候读入 ball2，这里我们可以推断出 ball2 为 0 表示某次球是否滚过。在 ball2 不为 0 情况下，ball1 和 ball2 分别表示下两次滚球的得分。我们看到，在 ball2 是否为 0 的后面，一个分支在 ball2 为 0 的情况下读入 ball1,这说明另一个分支上的①应该是对 ball1 的操作。很明显，在 ball2 不为 0 的情况下，ball1（值为 10）已经失去了意义，应该用来记录下次的得分，所以①处应该填写 ball2→ball1。在 ball1 和 ball2 都准备就绪以后，按照流程图说明，"其得分为 10 加下两次滚球所击倒的柱数"，所以②处就应该填写 10＋ball1＋ball2→score

结合试题说明的（2），从 3 的判断处开始，流程图右边的分支又可分为两具小分支。从试题说明可以推断出判断的某局前两局滚球结果是否为 spare，即两次滚球的结果是否击倒全部 10 个柱，这个判断是容易构造的。借用以上的分析结果，在 ball1 和 ball2 准备就绪的情况下，只要判断 ball1 和 ball2 之和与 10 敦大即可，所以我们在 3 处填写 ball1+ball2:10 来完成这个判断，这样，从判断结果的两种不同情况可以把两个小分支的作用明确下来，即左边的小分支处理结果为 spare 的情况，右边的上分支处理非结果 spare 的情况，结合试题说明与流程图结构，在这两个小分支里应该各有一个记分操作，否则在左右两个大分支的会合处 total+score→total 将不能正常工作，所以 4 处的答案应该是 10+ball1→score （"其得分为 10 加上下一次滚球所击倒的柱数"）5 处的答案应该是 ball1+ball2→score("其得分为本局两次滚球所击倒的柱灵敏之和")。

结合试题说明，不难看出，除第 10 局外，每局最多滚两次球（结果非 strike 时），最少滚 1 次球（结果为 strike 时）。这样记录前 9 局的得分在滚球次数最多的情况下需要一维数组有 18 个元素，在滚球次数最少的情况下需要一维数组有 9 个元素。第 10 局最少要滚两次球（结果非 strike 时），最多要滚 3 次球（结果为 Strike 时），这样，对一维数组的要求就是最少 11 个元素，最多 21 个元素。

问题 3 涉及的内容是在总分超过 100 时，每取得一次数 strike 奖励 5 分，因为这里是处理问题，结合以上对流程图整体结构的理解，我们就先排除了将所给流程图插入(c)处至(h)处的可能性，注意(g)和(h)的排除， 这里不详细说明排除的过程。②的解答我们已经给出，如果在之前插入给出的流程图，操作将会使奖励分丢失，权衡所有因素，给出的流程图应该插入②(b)处。

因为不是在考场上，我们再对流程图进行一点评论。本题目来判断某次球是否滚过。若某次滚球一无所获呢？如果用一个更为特殊的值。如负数、来表示某次球尚未滚过，就可以解决

这个问题。

从上述的解析过程我们可以得出一点方法性的结论，即我们不一定能够对流程图的每一语句框都透彻理解，只要对流程图的结构有一个宏观把握，结合试题说明，那么解答问题的逻辑严密性和正确性还是保障的。如果忽视了对流程图整体结构的把握，一味地研究每一个语句框，将会陷于细微之处，最终是事倍功半。

【答案】

［问题 1］

①ball2→ball1 ②10＋ball1＋ball2→score ③ball1＋ball2：10

④10＋ball1→score ⑤ball1＋ball2→score

［问题 2］ 最少 11 个，最多 21 个

［问题 3］ 流程图 6－60 插在 (b) 点

试题 2 （1990 年试题 2）

阅读下列说明和流程图，如图 6－62 所示，回答问题 1 和问题 2。

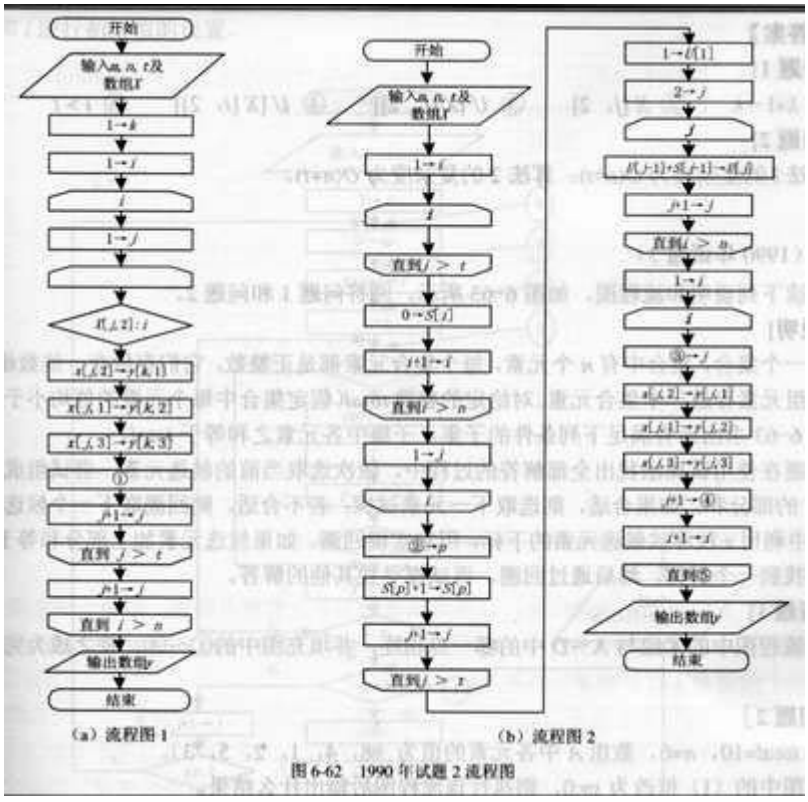


图 6-62　1990 年试题 2 流程图

［说明］

将一个 m × n 的矩阵 X 转置后存放到矩阵 Y 中，其计算复杂度为 O（m*n)。对稀疏矩阵来说，可以用紧凑的存储方式来减少所需的存储量，并降低计算复杂度。

已知一个有 t（t＞0）个非零元素的 m×n 稀疏矩阵 W（每行每列至少有一个非零元素）以紧凑方式存放在数组 X[1：t,1:3]用中。X 中某行的 3 个值为 时表示在 W 的第 行第 列有一个非零元素 。假定 X 中的元素已按行号列号递增排序。现要求将 X 转置后以紧凑表示形式存放在数组 Y[1：t,1:3]中，并且 Y 也按行号列号递增排序。

下面描述了两种紧凑的稀疏矩阵的转置算法。

算法一见流程图 6－62 的图（a)。

算法二见流程图 6－62 的图（b)。图中，数组元素 S[i]用来存放 X 中列号为 的元素个数，数组元素 U[j]用来计算 X 中第 列元素在 Y 中的行号。

［问题 1］
填充流程图 6－62 的图（a）和图（b）中的①～⑤，使之实现相应的算法。
［问题 2］
分别写出算法一和算法二的计算复杂度。
【解析】
流程图 6－62 的图（a）和图（b）实现了两种将稀疏矩阵 X 转置后存放到 Y 中的算法，其中 X 和 Y 都是以三元组形式存储的稀疏矩阵，都按行号和列号递增排序。结合我们对稀疏矩阵知识的了解，三元组形式存储的稀疏矩阵的转置，实际上就是将三元组中的行号与列号对换。但本题的难度在于怎样使转置后的矩阵 Y 也按行号列号递增排序。
仔细阅读流程图 1，连续的 3 个语句框 X[j,2]→Y[k,1],X[j,1]→Y[k,2]和 X[j,3]→Y[k,3]揭示了本流程图的核心所在。我们可以从这 3 个语句框推断出本流程图所使用的算法，即依次将 X 中第 列的所有元素送到 Y 中作为 Y 的第 行的元素。显然，这里的 应该是递加的，所以我们在①处填写 。
流程图 6－62 的图（b）比较复杂，需要解答的问题也比较多。但粗读流程图之后，不难发现该流程图由 4 个循环组成，让我们通过对 4 个循环的解析来解答题目。
显然，第一个循环将一维数组 S 的所有元素（n 个）置 0。至于 S 的如何使用，我们先放在一边。
由程序说明可知数组 S 中的元素 S[I]的用途是统计 X 中列号为 I 的元素个数，第 2 个循环中使用发数组 S。②将某个值赋给 p，从后面的使用来看 p 的值显然是一人列号。根据本循环执行 t 次的特点，我们推断②是斜某元素的列号（在循环中用 X[j,2]表示)，赋给 P。
数组 U 中的元素的用途是统计 X 中第 j 列元素在 Y 中的行号。第 3 个循环执行了 n 次，循环结束后数组元素 U[j]指示了 Y 中第 j 行的第 1 个元素在 Y 中的超始位置。
考查第 4 个循环，该循环的作用是将 X 中所有三元组送到 Y 的相应元素中。结合以上对数组 U 的作用及元素值特点的了解,根据 X 元素中的列号(x[i,2])，在 U 中即可查找到该列元素在 Y 中的相应位置。所以在 3 和 4 处我们都填写 U[x[i,2]]。
本循环的作用已经明确，循环次数控制也就明确了，该循环的执行次数由 X 中三元组的个数决定。我们在 5 处填写，控制循环执行 t 次。
流程图 1 所描述的算法是由一个二重循环构成主体。其内循环次数为 t，外循环次数为 n。故二重循环的循环次数为 n*t。所以其计算复杂度为 0。
流程图 2 所描述的算法由 4 个单循环构成，它们的循环次数分别为 n、tn-1t,所以其计算复杂度为 0（n+t)。
[答案]
[问题 1]
①k+1→k ② X[j,2] ③ U[X[i,2]] ④ U[X[i,2]] ⑤i>t
[问题 2]
算法 1 的复杂度为 O(n*t);算法 1 的复杂度为 O(n+t).

试题 3 （1990 年试题 3）

阅读下列说明和流程图，如图 6-63 所示，回答问题 1 和问题 2。

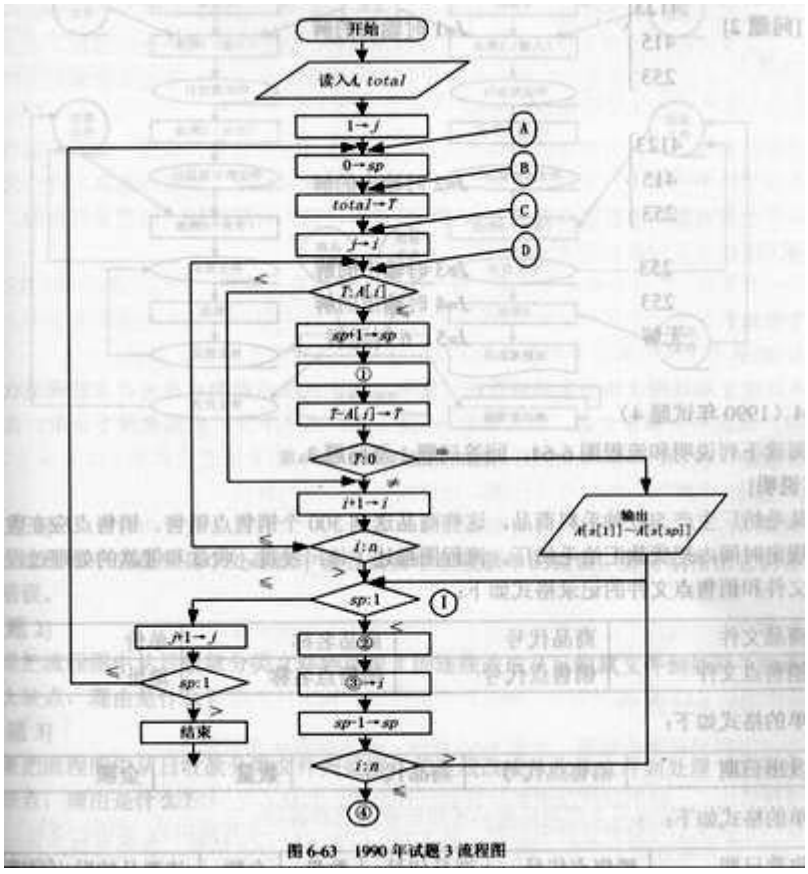


图 6-63　1990 年试题 3 流程图

［说明］

有一个集合，集合中有 n 个元素，每个集合元素都是正整数，它们存在一维数组 A 中，每个数组元素存放一个集合元素。对给定的整数 total（假定集合中每个元素的值均小于 total），流程图 6-63 求出所有满足下列条件的子集，子集中各元素之和等于 total。

本题在使用试探险去找出全部解答的过程中，依次选取当前的候选元素，尝试组成一个小于 total 的部分和。如果合适，则选取下一元素试探；若不合适，则回溯取下一人候选元素尝试。题中利用 S 栈存放候选元素的下标，用它实现回溯。如果候选元素加上部分和等于 total，则表示找到一个解答，然后通过回溯，再试探寻找其他的解答。

[问题 1]

问流程中的 4 应与 A～D 的哪一点相连，并填充图中的 1～3，使之成都市完整的流程图。

[问题 2]

设 total=10 n=6，数组 A 中元素的值为（8，4，1，2，。5，3）

若图中的(I)框改为 sp:0，则执行该流程图后输出什么结果。

[解析]

该题描述了一种用法试控法求解问题的处理流程，其核心是回溯技术。

我们先解决 4 与保处连接的问题，从 1→j,j→i,j+1→j 及两句 i:n 等语句来看，该程序的主体是一个二重循环，显然，4 是内循环的最后一句，应该连到内循环的第一句，结合我们的编辑经验，内循环的第一句不应在之前，因为我们看到变量 sp,T 和 i 都将在内循环中使用，每次内循环都要修改这些变量将使流程图的功能无法实现，所以 4 处应该连到 D 处。当然，这个判别断有很大的主观性，我们应以下的流程图中予以证实。

我们研究从 D 处以下到 T：0 的语句，由于 T 的初值已经是 total，从及此后的 T：0，这里应该是考虑某个元素是否包含进子集中，从试题说明中可以知道，考虑某一元素时，用堆栈

S 存放该元素的下标，显然，从来看，这里的元素下标就是 I，所以我们在处填写。
研究从 1 处到 4 处的内容，可以推断出这里处理一次探测到达集合最后一个元素及仍旧没有找到解而回溯的情况，回溯的方式是退栈，退栈的同时还要恢复 T 的值。然后从 s[sp]所指示的下一个元素寻找解答。从该部分语句的功能可以推出 2 和 3 的解答，即 T+A[s[sp]]→T 和 s[sp]+1。
结合以上对流程图的理解，我们印证了 4 处连到 D 处的正确性，因为每次外循环都要对 SP、T 和 i 进行初始值的设置。
解答问题 2 设计没有特殊的好方法，该题的解答是建立在对算法和流程图的正确理解之上的。根据流程图可知，流程图中每次执行内循环，都能找出 A[j]～A[n] 中包含 A[j]且满足条件的解 sp:n 和 sp:1 的区别在于回溯的范围。若(I)处改为 sp:0。很明显，在 SP=1 时仍要回溯。即堆栈中只包含 A[j]时仍旧要回溯。所以，如果 sp:1 改为 sp:0,每次内循环都能找出 A[j]～A[n] 之间所有的解。
[答案]

[问题 1] ①$i \rightarrow s[sp]$　②$T + A[s[sp]] \rightarrow T$　③$s[sp]+1$　④D

[问题 2]

82
4123
415
253　　J=1 时输出的解

4123
415
253　　J=2 时输出的解

253　　J=3 时输出的解
253　　J=4 时输出的解
无解　　J=5，6 时无解

试题 4 （1990 年试题 4）

阅读下列说明和流程图 6－64，回答问题 1 至问题 3。

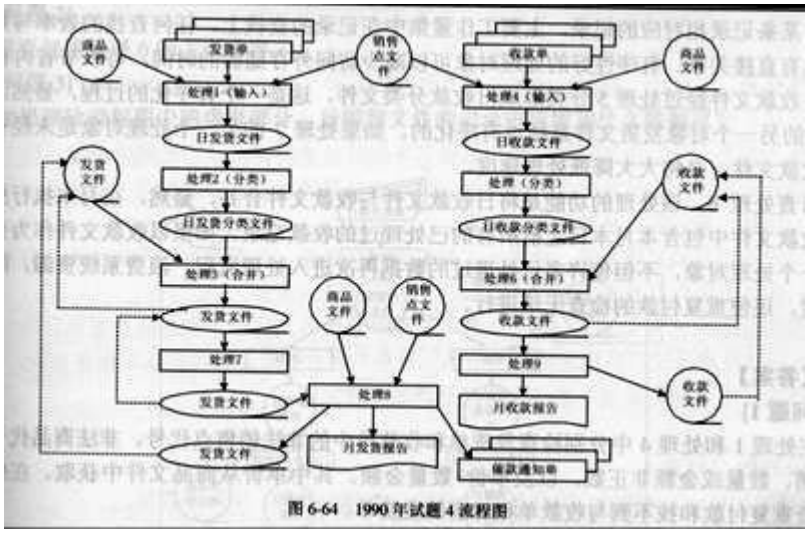


图 6-64　1990 年试题 4 流程图

［说明］

某毛纺厂生产 500 种毛料商品，这些商品送到 300 个销售点销售。销售点应在收到商品后的规定时间内把货款汇给毛纺厂。流程图描述了该厂发货、收款和催款的处理过程。其中商品文件和销售点文件的记录格式如下：

商品文件	商品代号	商品名称	单价
销售点文件	销售点代号	销售名称	地址

发货单的格式如下：

发出日期	销售点代号	商品代号	数量	金额

收款单的格式如下：

收款日期	销售点代号	商品代号	数量	金额	该商品的发出日期

处理 1～3 把当天的发货单合并到发货文件。处理 4～6 把当天的收款单合开到收款文件。每天在处理 3 和处理 5 做过之后，由处理 7 在发货文件中当天已收款的记录上加上已收款标记，处理 8 在月末执行一次，它有 3 个功能（1）汇总输出本月发货清单；（2）删除发货文件中已收款的所有记录，形成一个新的发货文件，作为下月初处理时的初始文件；（3）产生催款能知单，以便对那些一个月以前已发货但至今仍未收到货款的售点催款。处理 9 也每月末执行一次，除输出本月收款报告外，还删除收款文件中的所有记录。现假定不会有完全要同的发货单。

［问题 1］

指出流程图中应在哪几个处理框中检查发货单和收款单的错误，并分别指出它们各能指出什么错误。

［问题 2］

如果把流程图中从日收款分类文件到处理 7 的连线改成从日收款文件到处理 7 的连线，则有什么缺点，理由是什么？

［问题 3］

如果把流程图中从日收款分类文件到处理 7 的连线改成从收款文件到处理 7 的连线，则有什么缺点，理由是什么？

【解析】

阅读试题说明的目的是理解流程图的功能、某些文件的结构和某些处理的功能，这些内容为我们解答问题提供了丰富而有效的信息，不可忽视。

问题 1 要求我们回答在检查何处检查发货单与收款单中的错误，能够检查出什么错误。首先，检查数据的错误应该是数据进入系统后的第一道手续，否则，就不能保证后面的处理使用的是正确的数据。我们可以看到，处理 1 是输入发货单，处理 4 是输入收款单，则必须在处理 1 检查发货单的错误，在处理 4 检查收款单的错误。一般来说，错误检查应该考虑错误的客观存在而不能考虑错误的主观不存在，输入的任何数据项都存在出错的可能性，因此我们可能检查到非法的销售代号、非法的商品代号、非法的日期、非法的数量和非法的金额。由于数量和金额具有依赖关系，因此数量价格（从商品文件取得）金额也是可能检查到错误，另外，处理 7 要将收款单和发货单对应起来，我们不能排除重复付款和找不到与收款单对应的发货单的可能性。

我们首先考查处理 7 的功能，因为问题 2 和问题 3 都是围绕处理 7 出现的。处理 7 在发货文件中当天已收款的记录上加已收款标记，其内部处理应该是在发货文件中查找出与收款文件中某条记录相对应的记录，主要工作量集中在记录的查找上，任何查找的效率与数据的有序性有直接关系。有序性好的处理对象可以减少访问外存储器的时间，可以节省内存。

日收款文件经过处理 5 分类形成日收款分类文件，这是一个有序化的过程，显然，处理 7

处理的另一个对象发货文件提经过有序化的。如果处理 7 的另一个处理对象是未经有充化的日收款文件，必将大大降低处理速度。

考查处理 6，该处理的功能是将日收款文件与收款文件合并，显然，在月末执行处理 9 之前收款文件中包含本月本日之前所有的已处理过的收款记录。如果以收款文件作为处理 7 的另一个处理对象，不但使许多已处理过的数据再次进入处理流程，浪费系统资源，降低处理速度，还使重复付款的检查无法进行。

【答案】

［问题 1］

在处理 1 和处理 4 中分别检查发货单和收款单中的非法销售点代号、非法商品代号、非法日期、数量或金额非正数，以及单价＊数量金额。其中单价从商品文件中获取。在处理 7 中检查重复付款和找不到与收款单相对应的发货单。

［问题 2］

因为文件未分类，所以处理时要增加访问外存的时间或需要大量内存，从而降低了处理速度。

［问题 3］

收款文件中包含了已处理过的记录，从而降低了处理速度。

试题 5 （1990 年试题 5）

阅读下列说明和流程图，如图 6-65 所示，回答问题 1 至问题 3。

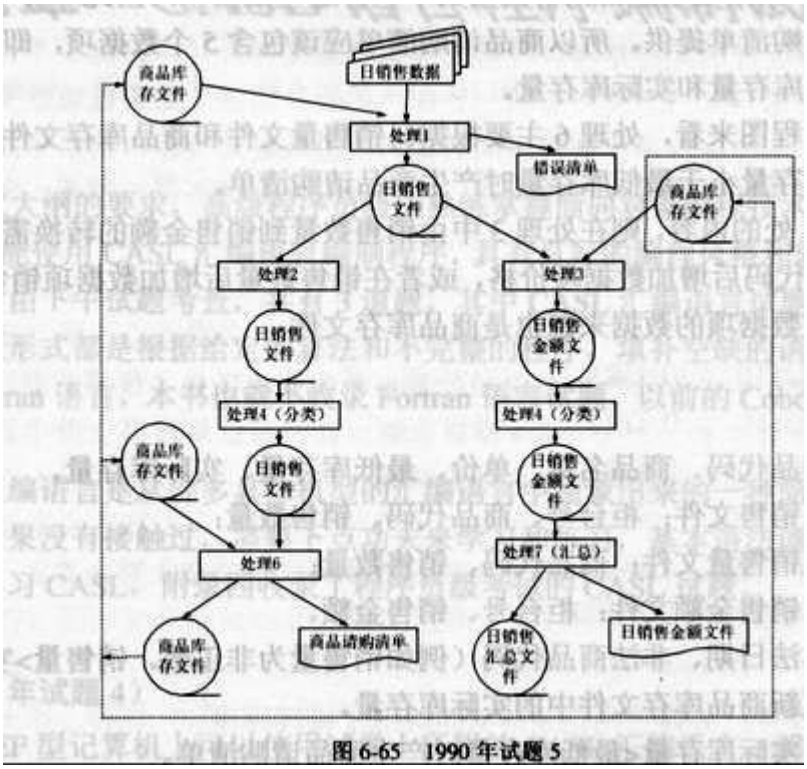


图 6-65　1990 年试题 5

［说明］

（1）流程图描述某大型商店商品销售的数据处理流程。

（2）商店设有若干柜台，同一种商品可能在几个柜台上销售，各柜台每天提供一组日销售数据，其格式如下：

日期、柜台号、商品代码、销售数量、商品代码、销售数量……

（3）数据处理系统每日产生一份反映各柜台当日销售金额和商店日销售金额的"日销售金额报告"，必要时还产生一份"商品请购报告"，给出那些低于最低库存量的商品代码、商品名

称、最低库存量和实际库存量。处理过程中产生存档的"日销售文件"和临时工作文件"日销售文件"和"日销售金额文件"。

（4）系统中所用到的数据均来自数据文件。

（5）流程图中的商品库存文件的记录已按关键字"商品代码"排序。

［问题 1］

①指出商品库存文件的记录中必须包括哪些数据项？

②分别指出在日销售文件，日销售量文件和日销售金额文件的记录中至少应包括哪些数据项，同时不产生数据冗余？

③错误清单可能指出哪些错误？

［问题 2］

简要叙述处理 6 的主要内容。

［问题 3］

如果删除流程图中的虚框部分，日销售文件的记录中应增加什么数据项？

【解析】

本题描述的是商品销售的数据处理流程。阅读试题说明，注意说明中的（3）和（4），对系统的功能和数据来源有一个大体把握。粗读流程图在处理 1 产生日销售文件后大致分为两个分支，左边的的分支处理日销售量及商品请购事务，右边的分支处理销售金额事务。其中在处理 1、处理 3 和处理 6 都要用到商品库存文件。

从流程图的处理 1 开始读起，处理 1 的作用输入日销售数据，生成日销售文件和错误清单。问题 1 的问错误清单可能指出哪些错误，答案的范围是明确的。凡是输入的日销售数据都有可能发生错误，如非法日期、非法商品代码（商品库存文件中无对应记录）和非法销售数量。处理 1 通过输入日销售数据，与商品库存文件相配合，产生日销售文件。除日期外，日销售数据包括柜台号、商品代码和销售数量。如果商品库存文件中的数据项在这里已经包括进日销售文件，那么在处理 3 和处理 6 处应该不再用到商品库存文件，所以我们断定日销售文件的数据全部严自日销售数据，即它最多包括柜台号、商品代码和销售数量 3 个部分。考查这 3 个数据项以后的用途，未发现数据冗余。

日销售文件通过处理 2 和处理 3 生成日销售量文件和日销售金额文件，其中日销售量文件用作销售量处理，因此至少应该包括商品代码和销倍数量。日销售金额文件要产生反映各柜台当日销售金额和商店日销售金额的日销售数量，日销售金额文件要产生反映各柜台当日销售金额和商店日销售金额报告，所以至少应该包括柜台号和日销售金额。数据项销售金额可以借助商品库存文件得到。

商品库存文件必须包含哪些数据项？首先是试题说明中提到的关键字商品代码；其次，在处理 3 中，商品销售数量到销售金额的转换需要商品价格，该数据项也必须包含在商品库存文件中；第三，处理 6 根据日销售量文件和商品库存文件产生商品请购清单，商品请购清单中的商品名称、最低库存量和实际存量 3 个数据项尚未落实来源。由于所有数据都来自数据文件，而另一个原始数据源日销售数据又不包含这些数据项，因此综合考虑，这 3 个数据项也应该由商品请购清单提供。所以商品请购单应该包含 5 个数据项，即商品代码、商品名称、价格、最低库存量和实际库存量。

从试题说明和流程图来看，处理 6 主要根据日销售文件和商品库存文件来更新商品库存文件，并在实际库存量小于最低库存量产生商品请购清单。

如果删除虚框 A 处的内容，则在处理 3 中由销售数量到销售金额的转换需要日销售文件的支持，应该在商品代码后增加数据项价格，或者在销售数量后增加数据项销售金额。当然，在日销售文件中增加数据项来源也是商品库存文件。

【答案】

［问题 1］①商品代码、商品名称、单价、最低库存量、实际库存量。
②日销售文件：柜台号、商品代码、销售数量；
日销售金额文件：柜台号、销售金额。
③非法日期、非法商品代码（例如销售量为非正数，销售量＞实际库量）。
［问题 2］ ①更新商品库存文件中的实际库存量。
②当实际库存量＜最低库存量时产生商品请清单。
［问题 3］ 在日销售文件中增加"单价"或"金额"。

六　CASL 语言程序编制度量精解

根据考试大纲的枎，高级程序员级应熟练掌握面向对象编程技术，熟练使用 C/C＋＋语言编制程序，能使用 CASL 汇编语言编制程序，具有良好的编程风格和较强的算法设计能力。程序编制能力由下午度量考查，共有 3 道题，其中 CASL 汇编语言试题 1 道，C 语言程序试题 2 道，试题形式都是根据给定的算法和不完整的程序，填补空缺的语句。从 1999 年开始，不再选考 Fortran 语言，本书也就不收录 Fortran 语言试题，以前的 Cobol 、Pascal 试题也没有收录。

CASL 汇编语言是从许多具体机型的汇编语言中抽象出来的一种简化的、专用于考试的汇编语言。如果没有接触过，需要下点功夫来学习和练习，基本语法请参见本书的附录三。为便于考生复习 CASL，附录四收录了程序员级考试的试题。

试题 1　（2000 年试题 4）

在 COMEF 型计算机上可以使用试卷上所附的 CASL 汇编语言。程序说明和 CASL 程序，将应填入 （n） 处的字句，写在答卷的对应栏内。

[程序 1 说明]

1. 本子程序根据每位职工的基本工资（非负值）和他完成产品的超额数或不足数计算职工的应发工资。

2. 主程序调用时，CRI 中给出子程序所需参数的起始地址，参数的存放次序如下表。

GR1

a1
b1
c1
a2
b2
c2
…
an
bn
cn
-1(结束标志)

其中为职工的基本工资，为职工的完成产品的超额数或不足数，为职工的应发工资数（i=1,2,…,n)。

bi 以原码形式存放（大于零为超额，小于零为不足)。基本工资与计算所得的应发工资以补码形式存放。

3．应发工资的计算规则为：

·恰好完成定额数（此时为零)，应发工资即为基本工资。

·每超额 4 件，在基本工资基础上增加 10 元（不到 4 件，以 4 计算。例如超额数为 10 时，增加 30 元)。

·每不足 4 件，在基本工资基础上减 5 元（不到 4 个，以 4 计算。例如，不足数为 5 时，减

10 元）。

[程序 1]

```
        START
BEG     PUSH 0,GR1
        PUSH 0,GR1
L1      PUSH 0.GR2
        PUSH 0.GR3
        (1)
        JEA GR0,0,GR2
        JMI FINISH
        LD GR3,3,GR1
        LEA GR2,0,GR3
        AND GR2,C7FFF
        JZE L3
L2      SRL GR3,15
        LEA GR2.-1,GR2
        (2)
L3      LEA GR2,-4,GR2
        JPZ L2
        (3)
FINISH  (4)
        (5)
        POP GR3
        POP GR2
7FFF    POP GR1
BONUS   RET
        DC #7FFF
        DC 10
        DC -5
        END
```

[解析]

阅读程序说明，特别要注意在 GRI 中给出的子程序所需要的参数。可以看出，有所给参数中包含 4 类数据：1、ai, ，职工 i 的基本工资；2、bi ，职工 i 的完成听超额数或不足数；3、ci，职工 i 的应发工资；4、－1，结束标志。

在阅读程序说明的其他部分以后，开始阅读程序。从程序开始到第 4 行是子程序的起始部分，执行保存现场的操作，将 GR1、GR2、GR3 的内容分别入栈，这一部分是容易理解的，而且也与本题的解答关系不大。

第 5 行需要填写内容，但现在程序刚刚开始，我们很难从已知的信息推断出这里应该填写的内容。那么就跳过第 5 行继续阅读程序。第 6 行的语句给我们很多想像。在这里将 GR2 的内容取到 GR0 中去了。注意，子程序开始时，只有 GR1 的内容对子程序是有用的，GR2 的内容是由调用子程序时主程序的运行善的。如果第 5 行不是对 GR2 进行操作，将某数据装入 GR2，第 6 行的操作将导致无法预测的结果，尤其时在第 7 行有一个 JMI 语句。这样，我们就可以判断第 5 行的大体内容。即在 GR2 中装入某个有用的数据。倡这里装入的是什么数据呢：仍旧需要结合后面的语句进行解答。

第 7 行的"JMI FINISH "可能导致程序的结束，问题是什么时候程序可能结束。程序当然是在遇到结束标志时结束，但在程序的开始部分就如此安排显然时不妥当的，我们再考察数据的存储。可能出现负数的只有结束标志和两类，显然遇到数额不足的职工就退出程序也是不妥当的。仍旧要从结束标志上看。现在让我们来假设，如果此处装入 GR2 的内容是 GRI 所指地址的内容，那么在处理完一个职工的工资后将 GRI 内容后跳 3 个字节，在处理完　了所有职工后，GR1 所指地址的内容巍然是结束标志。暂且作这样的假设，在填空（1）中填写"LD　GR2,0，GR1"。

继续阅读程序。第 8 行的语句告诉我们将某位职工的超额数或不足数装入 GR3，然后在第 9 行中又把 GR3 的内容拷贝到 GR2 中。第 10 行的 AND 操作究竟有什么作用？如果我们注意到是以原码方式存储的，而且注意到数据 C7FF 的特点，就不难判断这是取超额数或不足数的绝对值。当绝对值为 0 时就跳转到第 17 行的 L3。那么，从第 17 行起，必定有一个存储工资额的操作，因为在超额数或不足数为 0 时，应发工资即为基本工资。

接着阅读第 12 行。第 12 行的逻辑右移操作是简单的，可是目的难以判断。暂且在草稿纸上演示一番。当 GR3 是超额数时，经过右移，GR3 的内容就变成 0；当 GR3 是不足数时，经过右移，GR3 的内容就变成 1。这 1 和 0 怎样表示是超额数还是不足数呢？先跳过再说。第 13 行的操作更让人摸不着头脑，为什么把数额给减了一个去，我们也暂时存疑吧。

第 14 行是需要填写内容的，从下面的将 GR2 内容减 4 看，这里是将超额数或不足数的绝对值减了 4，结合后面的"JPZ　L2"，第 14 行的语句应该是将奖金或罚款与基础工资进行加减操作的语句。考察变量 BONUS 定义，如果直接引用该变量，那么它就是 10，如果引用该变量的下一个存储单元，就是－5。让我们再次提起第 12 行的逻辑右移。很、明显，如果我们这里使用 GR3 的内容来处理奖金或罚款，应该是很方便的。比如说，在这里填写"ADD GR0，BONUS，GR3"，这样，当 GR3 内容为 0（数额为超额数）时，这里回到 GR0 中的就是 10，当 GR3 内容为 1 时（数额为不足数）时，这里加到 GR0 中的就是－5。再仔细阅读从第 12 行到第 16 行的语句，这样的实现是能够满足需要的。

现在看第 16 行，JPZ 操作是在等于或大于时跳转。我们在这里可以举一个例子，当超额数为 4 时，第一次执行第 14、15 行时就完成了对应发工资的计算和对超额数进行减 4 操作，接着执行第 16 行，不免要跳转到第 14 行再进行一次奖金发放和超额数减 4，然后才能够执行第 17 行的操作。显然这是不符合道理的。结合第 13 行考虑，上述的错误就不会发生了，具体过程学员可以自己计算。

关于第 17 行的内容，在前面我们已经判断为存储应发工资额了。现在关键是看如何实现，由于是要写到某个内存地址，所以必须使用 ST。结合以上程序，可以知道应发工资额存储在 GR0 中，那么这里的第一个操作数就是 GR0 了。如何定位地址？程序中始终使用 GR1 来指示内存地址，这里最好不要例外，而且要想用别的寄存器指示内存地址还需要额外的操作，在这里是不允许的，所以我们就这样构造答案："ST GR0，1，GR1"

在第 18、19 行是在处理完一个职工的信息之后的操作，这里应该是什么操作？显然，应该是将 GR1 的内容加 3，为下一个职工的信息处理提供正确的内存地址信息，然后再强制跳转到 L1 处。

在本题的答案中，我们应注意两点，一是始终注意寄存器内容的变化，二是始终注意变量的灵活使用。注意寄存器内容的变化，不仅要求掌握寄存器的内容发生了哪些变化，而且要求对寄存器在某种情况下应该如何变化有清醒的把握。对变量的灵活使用也包括两个方面，一是明确变量使用后发生的效果，其次是如何对连续存储的变量进行操作。

[答案]

（1）LD GR2，0，GR1

（2）ADD GR0，BONUS，GR3

（3）ST GR0，2，GR1
（4）LEA GR1，3，GR1
（5）JMP L1

试题 2 （1999 年试题 4）

在 COMST 型计算机上可以使用试卷上所付的 CASL 汇编语言，阅读程序说明和 CASL 程序，把应填入（n）处的字句，写在答卷的对应栏内。

[程序 1 说明]

本子程序是对 15 位二进位串，求它的奇校验位，生成 16 位二进位串，使 16 位二进位串中有奇数个 1。

进入此子程序时，15 位二进位串在 GR1 的第 1 位至 15 位，并假定 GR1 的第 0 位是 0。求得的奇校验位装配在 GR1 的第 0 位上。

[程序 1]

```
START
BEG     PUSH 0，GR2
        PUSH 0，GR3
        LEA GR3，1
         （1）
L1      SLL GR2，1
         （2）
        LEA GR3，1，GR3
L2      JZE L3
        JMP L1
         （3）
        ST GR3，WORK
        ADD GR1，WORK
        POP GR3
        POP GR2
        RET
WORK    END 1
```

[程序 2 说明]

子程序 SUM 是将存储字 A 起的 n(n>0)个字求和，并将结果存于存储字 B 中。

调用该子程序时，主程序在 GR1 中给出存放子程序所需参数的起始地址。参数的存放次序如下：

```
（GR1）+0 A
       +1 n
       +2 B
```

[程序 2]

```
SUM     START
        LD GR2，0，GR1
L5      LD GR3，1，GR1
        LEA GR0，0
```

```
        ADD GR0，0，GR2
        LEA GR2，1，GR2
        （4）
        JNZ L5
        （5）
        ST GR0，0，GR3
        RET
        END
```

[解析]

[程序 1]

在解答试题之前我们在这里引用 1992 年程序员级考试 CASL 语言试题的第一道题。试题说明大同小异，只是现在要求的是奇校验，而 1992 年要求的是偶校验而已，我们只能说，两道试题不仅仅是相似。下面看 1992 年的试题：

```
START
BEG     （1）
        （2）
LEA     GR3，0
LEA     GR2，0，GR1
        （3）
        （4）
LEA     GR3，1，GR3
L2      （5）
JMP     L1
L3      （6）
ST      GR3，WORK
ADD     GR1，WORK
POP     GR3
POP     GR2
RET
WORK    DS 1
        END
```

首先阅读试题说明和程序 1。试题说明没有提供更多的关于算法的信息。但既然是求奇偶校验位，常用的方法是移位并逐位确认并累积"1"的个数，从程序中的"SSL GR2，1"与"LEA GR3，1，GR3"两条语句可以看到这种算法的轮廓。下面我们通过读程序来解答题目。

在填空（1）之前是一些程序初始化工作，从程序的开头和结尾处对 GR2 和 GR3 的 POP 和 PUSH 操作来看，前 3 条语句只是设置了 GR3 的初值，而填空（1）之后马上又对 GR2 进行操作，显而易见填空（1）是对 GR2 的赋初值进行操作。从程序中的"SLL GR2，1"和"ADD GR1，WORK"可以看出，真正在程序中进行移位比较的是 GR2，求校验位的 15 位二进制位串还保存在 GR1 中，没有参与移位运算。基于上述分析，（1）的内容就很明确了，即"LEA GR2，0，GR1"，完成取待处理的二进制位串道 GR2。

从标号 L1 一直到语句"JMP L1"可以看出这是一个循环，那么作为程序中唯一的循环结构，确定"1"的个数应该是在这个循环结构中完成的。"SLL GR2，1"和"LEA GR3，1，GR3"增强了我们对这个判断的饿信心。不管 GR3 的初值如何，在循环体中必须有对当前是否为"1"的

判断，综合考虑循环体，任务只能由语句（2）来完成。那么，这里应该是判断其为"0"呢，还是判断其为"1"？跳转的目的地是哪里？

假设跳转到 L2，那么我们看在这里如果 GR2 内容全为"0"的情况下要跳转到 L3。如果 GR2 的内容全为"0"，则表示移位操作已经完成，程序也就应该结束了。跳转到 L2 的假设使跳转到 L3 的假设失去了意义，因为只有程序计算出二进制位串中"1"的个数后才能跳转到 L3。由此可以断定跳转的目的地是 L2。

但我们仍旧不能确定填空（2）中的判断。我们假设是判断"1"，那么在当前位为"0"的情况下执行"LEA GR3，1，GR3"，累加"1"的个数。在全"0"的情况下也不会发生跳转，这样就多加了一个数位。如果判断填空（3）是对"0"的个数的处理，填空（3）和后续的两句实现写入校验位是不可能的，所以这里应该是判断"0"的。综合考虑，我们就填写"JPZ L2"。

现在通读程序，不难得出结论，当程序运行到 L3 时，GR3 的内容为二进制位串中"1"的个数加 1。如何使 GR3 的值满足"ST GR3，WORK"和"ADD GR1，WORK"的要求，共同完成写二进制位串的校验位的功能？首先，从上面提到的两句来看是在 GR3 中形成了一个首位为"1"或"0"（依"1"的个数而定），其余各位全部是"0"的数字。我们看，由于 GR3 的初值是 1，当二进制位串中有偶数个"1"时，GR3 中即为奇数，反之为偶数。根据奇校验的规则，当二进制位串中有奇数个"1"时保持首位为"0"，反之则将其首位变为"1"。考虑到当 GR3 内容为偶数时末位为"0"为奇数时末位为"1"，我们可以在填空（3）中填写"SLL GR3，15"来使 GR3 的末位移动到首位，而其余各位都是"0"。

[程序 2]

鉴于这是一道非常容易的试题，我们直接通过阅读程序来解答。

首先，程序的开头是两个 LD 语句 ，结合程序说明，很容易确定它们的含义，即将起始地址 A 和数字的个数分别读进 GR2 和 GR3。从此后对 GR0 的操作来看，GR0 中存储的是加法运算的结果，而语句"LEA　GR2，　1，　GR2"则实现地址指针的偏移。

那么填空（4）中的语句究竟应该是什么呢？从下面的"JNZ　L5"来看，L5 和填空（4）之间的语句构成一个完整的循环体。从整体上考虑，（4）应该是循环控制变量的修改，由于 GR2 和 GR1 都不能用作循环控制条件，结合 GR3 内容的含义，我们构造"LEA　GR3，　－1，GR3"完成循环变量的修改。

而"JNZ　L5"之后的部分，跳过填空（5），"ST　GR0,0，GR3"是将求和运算的结果存储到 GR3 所指示的地址中去。从程序说明可知 GR3 所指示的地址应该是（GR1＋2），那么（5）的内容就很容易确定了，即"LD　GR3,2，GR1"，完成将 B 传入 GR3 的操作。

在解答上面两道题的过程中，我们有 3 点体会：1、研究历年试题很有用；2、注意 LD 和 LEA；3、注意通过条件跳转构造循环的算法。

[答案]

（1）　LEA　GR2,0，GR1
（2）　JPZ　L2
（3）　SLL　GR3,15
（4）　LEA　GR3，－1，GR3
（5）　LD　GR3,2，GR1

试题 3　（1998 年试题 4）

在 COMET 型计算机上可以使用试卷上所附的 CASL 汇编语言。阅读以下程序说明和 CASL 程序，将应填入（n）处的字句，写在答卷的对应栏内。

[程序说明]

本程序是统计字符串中数字字符'0'至'9'的出现次数。

字符串中的每个字符是用 ASCII 码存储。一个存储单元存放两个字符，每个字符占 8 位二进位。

程序中，被统计的字符串从左至右存放在 STR 开始的连续单元中，并假定其长度不超过 200，字符串以'.'符作为结束。NCH 开始的 10 个单元存放统计结果。

[程序]

```
        START MIN
MIN     LEA GR2,9
        LEA GR0,0
L1      (1)
        LEA GR2,-1,GR2
        JPZ L1
        LEA GR4,0
        LEA GR1.0
L2      LD GR2,STR,GR1
        EOR GR4,C1
        JNZ RL
        (2)
PL      STL CR2,8
        LEA GR3,0,GR2
        SUB GR3,C9
        JMI L3
        JNZ L4
L3      LEA GR3,0,GR2
        SUB GR3,C0
        JMI L5
        LEA GR2,1
        (3)
        (4)
L4      LEA GR4,0,GR4
        JNZ L2
        (5)
        JMP L2
L5      SUB GR2,C
        JNZ L4
        EXIT
C1      DC 1
C       DC '.'
C0      DC '0'
C9      DC '9'
STR     DS 200
NCH     DS 10
        END
```

[解析]

仔细阅读程序说明，不难发现本题的难度在于一个存储单元存放两个字符，这样每个存储单

元必须进行再次统计。如何控制先统计存储单元的前 8 位再统计其后 8 位？让我们带着这个问题来阅读程序，通过对程序的理解来集会其控制算法。

在通读程序之后　，我们会发现一直到"JPZ　L1"之前的语句无论在位置上还是在逻辑上都属程序的前奏部分。因为此后没有出现跳转到 L1 的语句。从"LEA　GR1,0"、"L2　LD　GR2　STR，GR1"大致可以看出 GR1 是用作统计时的地址指针的，不可能再次出现 GR1 置 0 的情况，因此排队了后面 4 个空语句中存在跳转到 L1 的可能性，确定了该部分语句作为程序前奏的地位。通过"MIN　LEA　GR2,9"和"LEA　GR2，－1，GR2"及"JPZ　L1"可以断定此处语句共执行了 10 次。程序中需要进行 0 次操作的变量只有 NCH 一个，所以我们就可以大胆的在填空（1）中填写"ST　GR0，NCH，GR2"，需要注意的是，这里 GR2 既作循环计数器又作地址偏移指针。

此后开始的语句应该是本程序要完成的中心任务，一一统计各字符出现的次数。为了在理解算法的时候不出现偏差，应先通过阅读程序来确定各个寄存器在程序中的用途。GR1 在程序中出现过两次，而且是置 0 操作和用作取内存内容的地址偏移指针，不存在地址偏移指针固定不变的情况，由此可以断定在后面的 4 个填空处至少应该有一处是对 GR1 进行操作的，只是位置有待确定而已。GR4 在程序中出现了 3 次，分别是置 0、与 C1 导电异或和检测是否不为 0，考虑到异或的结果是使 GR4 的内容在 1 和 0 之间变换，我们大致可以判定 GR4 是用作某种标志值。至于 GR2、GR3 的用途，由于出现次数较多，用法灵活，因此只能在程序中结合具体情况进行分析。

从"L2　LD　GR2，STR，GR1"开始阅读程序，因为已经大致判定 GR4 是用途某种标志的，所以可以发现，在"DOR　GR4，C1"之后有一个跳转，即当 GR4 内容不为 0 时跳转到"RL　SRL　GR2,8"而 GR2 的内容是 STR 某一内存单元的值。问题就在这里，程序是否通过 GR4 的值来控制统计前 8 位还是后 8 位？如果是这样，那么在不发生跳转时，从上下语句来看，能够看到当处理后 8 位时不发生跳转。怎样保证右移 8 位之后 GR2 的内容正好是 STR 某一内存单元后 8 位的值？方法只有一个，就是先把 GR2 左移 8 位。因为仅需要 8 个二进制位的值，所以防止移位的时候出现符号位问题，结合语句"SRL　GR2,8"我们选择在这里填写语句"SLL　GR2,8"。

继续跟踪 GR4 的变化，在语句"L4　LEA　GR4,0，GR4"和"JNZ　L2"处，可以结合以上的分析结果断定此处是判别一个内存单元是否已经处理完毕。如果已经处理完毕，很明显应该进行偏移地址的递加，应该是对 GR1 的操作，于是填空（5）处就填写有关 GR1 加 1 的语句，这里语句是不唯一的，可以填写"LEA　GR1,1，GR1"，这是比较平实的方案。如果考虑到物尽其用的原则，不妨填写"ADD　GR1，C1"的语句，充分发挥 C1 的作用，这已经是个人习惯的问题了。

以上分析并未涉及比较统计操作，只是对程序中的特殊用途寄存器进行分析，判断出填空（2）和（5）的内容。现在尚未处理的两个空白应该是涉及到比较和计数两方面操作的。

阅读程序，不难发现当程序执行到"L3　LEA　GR3,0，GR2"处时，已经表明 GR2（GR3）的内容是一个小于"9"（C9）的字符。执行到"LEA　GR2,1" 处，各寄存器的内容应该是：GR3————－统计比较对象（应该是大于"0"）减去"0"（C0）的值，GR2——预备进行计数累加的数值 1。那么，在下面的两个空白处应该填写的内容就是计算累加的工作，结合对程序整体的理解，我们就在这里填写"ADD　GR2，NCH，GR3"与"ST　GR2，NCH　，GR3"两个语句。注意 GR3 的内容既然是统计比较对象（应该是大于"0"）减去"0"（C0）的值。也就应该是进行计数时相对于 NCH 首地址的偏移量。

[答案]

(1) ST GR0,NCH,GR2

(2) SLL GR2,8

(3) ADD GR2,NCH,GR3
(4) ST GR2,NCH, GR3
(5) LEA GR1,1,GR1 或 ADD GR1,C1

试题 4　（1997 年试题 4）

在 COMET 型计算机上可以使用试卷上所附的 CASL 汇编语言。阅读下列程序说明和 CASL 程序，将应填入（n）处的字句，写在答卷的对应栏内。
[程序说明]
本子程序将一个非负二进整数翻译成 5 位十进整数字符。
进入子程序时，在 GR0 中给出被　翻译的非负二进整数，在 GR2 中给出存放 5 位十进整数字字符的起始地址。
十进制数字符字用 ASCII 码表示。当结果小于位时，左边无意义的 0 用空白符替换；当二进整数为 0 时，在（GR2）＋4 中存放 0 的 ASCII 码。
数字字符 0 至 9 的 ASCII 码是 48 至 57，空白符的 ASCII 码是 32。
[程序]

```
        START
        LEA GR1,0
        LEA GR3,32
L1      (1)
        JPZ L2
        ST GR3,0,GR2
        LEA GR2,1,GR2
        LEA GR1,1, GR1
        LEA GR4,-4, GR1
        JNZ L1
L2      (2)
L3      (3)
        JMI L4
        SUB GR0,SN0, GR1
        LEA GR3,1, GR3
        (4)
L4      ST GR3,0, GR2
        LEA GR2,1, GR2
        LEA GR1, GR1
        (5)
        JNZ L2
        RET
SNO     DC 10000
        DC 1000
        DC 100
        DC 10
        DC 1
        END
```

[解析]
本题所给出的程序是对数进行形式转换，将非负二进整数转换成由 ASCII 码表示的十进制整数字符串，这里我们首选要的。另外在 CASL 语言中没有除法指令，要利用减法来实现除法。求商的过程是将被除数减去除数，得到的差再减除数，这样一直到不够减为止。减操作的次数即为商，剩下的数为余数。
阅读已经给出的 CASL 程序，我们能够看到，程序的最后定义了以 SNO 开始的 5 个数字，即 10000、1000、100、10、1。由此我们可以判断，程序中就是利用这 5 个数来分别求得十进制数的 5 位数字字符的，即将原数除以 100…….就一直十进制数的最后的一位。
5 个需要解答的问题将程序分解得支离破碎，无法从整体上进行把握，只好根据上面对程序算法的推断对程序的局部进行分析，从推断局部的功能入手来解答问题。阅读程序，发现填空（1）后是一个跳转语句，如果不跳转的话，从以后的语句看是把翻译结果从左边开始置"0"，然后 GR2 和 GR1 都加 1。这样我们就能够明确填空（1）的功能，（1）是考查待翻译的二进制整数翻译成十进制数后是否满足 5（4、3、2 等）位，所以应该是 GR0 与以 SNO 为首址的 5 个整数的比较。如何控制呢？是否可以采用"CPL　GR0　SNO　GR1"来完成任务呢？在这个循环中只有 GR1 可以担当这个任务。由此推断在此后的程序中，GR1 一直担当指示 SNO 偏移位置的指针，而且还能够推断 GR3 存放商，GR4 的作用好象是与 GR1 配合，记录已经翻译的倍数。
填空（2）和（3）的连续出现让我们摸不着头脑，但考查（3）和（4）之间的语句，不难尽责这是循环减的语句段，所以（4）应该是一个跳转语句。由于 GR3 的功能所限，不能用 GR3 来控制一个条件跳转，只能构造一个无条件跳转，但跳转的结果是 L2 还是 L3，现在还不能明确。分析这个循环的功能，不难发现，控制循环结束的语句（3）和其后续语句。那（3）应该是一个比较语句，因为如果用减法将影响此后的递减，考虑到处理对象是非负整数，选用 CPL 来构造解答，结合以上对寄存器使用情况的理解，不难构造出（3）的解答为"CPL　GR0，SNO，GR1"，同时还可以确定填空（4）跳转的目标应该是 L3。填空（2）先放在一边。
L4 直至"JNZ　L2"完成向内存中写入一位翻译结果的任务，结合解答（1）时对 GR4 作用的理解，大致可以判定 GR4 的作用是记录已经翻译的位数。依照上面的语句，在这里填写"LEA GR4，－5，GR1"。
填写完了 4 个答案以后，通读程序，填空（2）的作用和答案就非常明显了，在每一次循环减操作之前一定要给 GR3 清 0，否则商的叠加会使翻译结果莫名其妙，所以在这里填写"LEA GR3,48"。
[答案]
（1）　CPL　CR0，SN0，G,1
（2）　LEA　GR 3,48
（3）　CPL　GR0，SN0，GR1
（4）　JMP　L3
（5）　LEA　GR4，－5，G,1

试题 5（1996 年试题 4）

在 COMET 型计算机上可以使用试卷上所附的 CASL 汇编语言，阅读下列程序说明和 CASL 程序，将应填入（n）处的字句，写在答卷的对应栏内。
[程序说明]
子程序 OFFSET 用二分法，查找无符号整数 M 在一个长度为 N 的有序（升序）无符号整数列表 NTABLE 中的位置。

程序中标号为 LOW 和 UP 的两个存储字分别用作存放查找区间的上下限。
进入子程序时，在 GR1 中给出存放子程序所需参数的起始地址。参数的次序如下：

(GR1)+0	M
1	N
2	NTABLER 的首址

从子程序返回时，GR0 中存放查找结果，即 M 在此有序表中的位置序数。如果表中找不到 M，则 GR0 返回 0，其他寄存器的内容保持不变。
[程序]

```
        START
OFFSET  PUSH 0, GR2
        PUSH 0, GR3
        LD GR0, 0, GR1
        LEA GR2, 0
        ST GR2, LOW
        (1)
        (2)
        ST GR2, P
LOOP    ADD GR2, LOW
        SRL GR2, 1
        LEA GR3, 0, GR2
        (3)
        (4)
        JZE FOUND
        JPZ INCLOW
        LEA GR2, -1, GR2
        ST GR2, UP
        JMP CMPLU
INCLOW  LEA GR2, 1, GR2
        ST GR2, LOW
        (5)
CMPLU   CPL GR2, LOW
        (6)
        (7)
FOUND   LEA GR0, 1, GR2
        POP GR3
        POP GR2
        RET
LOW     DS 1
UP      DS 1
        END
```

[解析]
试题说明没有提供更多的信息，在这里我们只能了解程序中一些变量的含义。所以问题只能通过两点来解答，一是对程序结构的理解，一是对二分法的掌握程序。二分法的查找过程为：

先将待查记录所在的范围（区间）分成两个区间，然后确定待查的记录所在的区间。如此循环往复，直到查找成功或失败。这样，无论如何实现二分法，有些工作是必须的，例如确定查找的上限和下限，确认待查询记录所在的区间。

考察试题所提供的程序，可以推断出程序的对二分法的具体实现过程：首选设指针 LOW 和 UP 分别指向 NTABLE 的第一个元素和最后一个元素，比较 M 与表的第[（LOW＋UP）/2]个元素（[ ]）表示取整），即与区间的蹭元素相比较。若相等，则找到；若 M 大，则说明要找的元素在区间的上半部，所以下一次比较将 LOW 值置为[（LOW＋UP）/2]＋1，UP 值不变；若 M 小，则说明要找的元素在区间的下半部，所以下一次比较将 LUP 值置[（LOW＋UP）/2]－1，LOW 值不变。循环的结束条件是 LOW＞UP，此时查找失败。

结合算法来阅读程序。程序在完成了一些例行工作之后，语句"LD　GR0，0，GR1"将要查找的值 M 放入 GR0 中，接下来是设置查找的下限 LOW，显然下面该初始化指针 UP 了。由以上 LOW 的初值为 0,可以推断 UP 的正确初值应该是 N-1。从填空(2)的下一行"ST GR2，UP"来看，UP 的初值是经 GR2 传入的，而且填空（1)、(2）完成对 UP 的赋初值操作，这样填空（1)、(2）应设为 GR2 等于 N-1，所以（1)、(2）两行应为：

LD GR2，1，GR1；把 N 放入 GR2 中

LEA GR2，-1，GR2；把 N-1 放入 GR2 中

接下来就开始二分法查找，首先求得 UP+LOW 的值，进而求得[（UP+LOW）/2]的值。然后将算出的中间位置值赋给 GR3。显然这里的中间位置只是相对于 NTABLE 的首址而言的。如果在程序中真正用来与 M 进行比较，必须换算成真正的逻辑地址，所以填空（3）的答案为"ADD GR3，2，GR1"，将 NTABLE[（UP+LOW）/2]所在地址赋给 GR3。从接下来的几个跳转语句来看填空（4）的功能应该是进行 M 与 NTABLE[（UP+LOW）/2]比较，然后才有此后根据比较结果进行的跳转。观察"JPZ　INCLOW"我们知道，应将　M　值在前，NTABLE[（UP+LOW）/2]在后比较，所以（4）的答案应为"CPL GR0，0，GR3"。

若比较结果相等，程序跳转至 FOUND，这时将位置信息存入 GR0 返回。若 M 大，则跳转至 INCLOW，将 NTABLE[（UP+LOW）/2]+1 的值赋给 LOW。因为 LOOP 循环开始时已假定 GR2 中为 UP 值，所以填空（5）的答案为"LD GR2，UP"，用来恢复 GR2 的值，使 GR2 的值位 UP 的值。若 M 小，则将 NTABLE[（UP+LOW）/2]-1 的值赋给 UP，然后应当判断循环是否结束，即判断 UP 是否小于 LOW，程序中 CMPLU 开始的几句完成这一判断。在 UP>=LOW 时,继续查找,否则应该返回,所以填空(6)的答案应为"JPZ LOOP"。当 UP>=LOW 时，继续查找，否则 UP<LOW，循环结束，返回值应当为 0，以示查找失败。考虑到填空（7）下面的一条语句是将 GR2 加 1 后传送到 GR0，所以（7）的答案应该是"LEA GR2，-1"，与下一条语句结合来完成设置查找失败标志的功能。

[答案]

（1）　LD　GR2,1，GR1 或 ADD　GR2,1，GR1

（2）　LEA　GR2，－1，GR2

注：(1)、(2）两条指令在 GR2 中形成 N－1 的解答均正确，如：

（1）　LD　GR3,1，GR1

（2）　LEA　GR2，－1，GR3

或

（1）　LEA　GR2，－1

（2）　ADD　GR2,1，GR1

若在 GR2 中碰 N 可以给 1 分

（3）　ADD　GR3,2，GR1

（4）　CPL　GR0,0，　GR3

（5） LD　GR2，UP
（6） JPZ　LOOP
（7） LZA　GR2，－1 或 SUB　GR2，LOW
注：若（6)、(7）的解答为：
（6） JMI　FOUND
（7） JPZ　LOOP 或 JMP　LOOP　亦可给 2 分

试题 6 （1995 年试题 6）

在 COMET 型计算机上可以使用试卷上所附的 CASL 汇编语言。阅读下列程序说明和 CASL 程序，把应填入（n）处的字句，写在答卷的对应栏内。
［程序说明］
本程序是按某种规律自动生成 16＊16 单色点阵图形。点阵图形中每行 16 点的信息排列在一个存储字中，16＊16 点阵图形可以用连续的的 16 个存储字来表示。
程序中标号为 PTN 开始的 16 个存储字连续存放自动生成 16＊16 点阵图形，点阵图形中的第一行作为已知数据给定，其余 15 行按下述规则自动生成：
1． 每个存储字的第 0 位和第 15 位（即边界点）恒为 0。
2． 一个存储字的第 n 值 0＜n＜15 取决于前一存储字的第（n-1）位和第(n+1)位的值是否相同。若这两位的值相同，则第 n 位为 0，否则为 1。
3． 例如，点阵图形第一行的存储字内容为＃35B4，按上述规则生成点阵图形第二行的存储字内容为＃71B2，其余类推。
0011010110110100 #35B4
0111000110110010 #71B2
0…… 0 ……..
[程序]

```
        START
        LEA GR1, 0
        LD GR0, PTN, GR1
LOOP    SLL GR0, 2
        (1)
        (2)
        (3)
        LEA (4)
        (5)
        LEA GR2, -15, GR1
        JMI LOOP
        EXIT
PIN     DC #35B4
        DS 15
        END
```

[解析]
粗读程序，应该发现需要解答的问题非常集中。研究程序结构，不难确定需要解答的部分大致覆盖了程序的核心部分，生成并存储各位。这样，解答的关键就是如何从有限的语句中推断程序所使用的算法。
从试题说明中不难看出，点阵的每个存储字的第 0 位和第 15 位为 0。除了第 1 个存储字外，

每个存储字的第 n 位取决于上一个存储字的第 n-1 位的和 n＋1 位的值是否相同，也即由上一个存储字的第 n－1 位的和第 n＋1 位异或而得。

阅读程序，可以推断出在标号 LOOP 和语句"JMI　LOOP"之间的语句完成生成一个存储字的操作，因为作为最外层的循环其循环体执行了 15 次。是否在需要解答的部分还有一个循环需要构造？从语句数量上来看是不够的，因为 5 条语句除了循环变量的初始化、循环变量的修改和跳转之外剩余的两条语句已经难以完成取两位异或并存储的任务，看来程序采用的是另一种算法。

考察语句"LD　GR0，PTN，GR1"和语句"SLL　GR0,2"的功能。由于需要生成的存储字第 0 位和第 15 位的值是固定的，我们需要解决的是中间的 14 个二进制位的值，把当前存储字左移 2 位并与当前存储字异或应该是比较简便的方法。我们姑且认定程序采用的就是这种算法。

如果采用上述算法，现实的问题就是在新生成的存储字中，第 0 位到第 13 位是符合要求的存储字的第 1 位到第 14 位，而最后两位的值是不确定的。为达到第 0 位和第 15 位为 0 的目的，我们可以采用逻辑移位的方法，可以采用的方法是先逻辑左移 2 位，再逻辑右移 1 位。把上述算法用 CASL 语句表达，可以填写在填空（1）、（2）和（3）处。仔细推敲答案，上述语句在功能实现上没有问题，这也印证了我们上面的推断。

由于在填空（5）之后有通过对 GR1 减 15 以确定是否结束循环的语句，在（4）和（5）中应该有一个对 GR1 进行操作的语句，否则不但循环无法结束，就是以后的循环也难以进行正确的 LD　操作；在上面已经生成了一个符合要求的存储字，应该将其存储进 PTN 之后的连续存储空间内，这项任务也应该在填空（4）和（5）中完成。老鸦到（4）所在的语句是一个 LEA 操作，我们在 (4) 中填写"GR1,1GR1"以完成第 1 项任务，在 (5) 中填写"ST　GR0，PTN，GR1"以完成第 2 项任务。

在所有需要解答的问题都解答完毕之后，我们再从整体上阅读程序，以确定解答的正确性。本题给我们的启示就是在无法看出程序所使用的算法时，应该通过各种迹象来推断程序所使用的算法，以这个推断为出发点解答题目。推断算法的关键就是对程序中问题前后的语句的功能进行充分的理解，交考察它们对解答题目的直接意义。

［答案］

(1) EOR GRO,PTN,GR1

(2) SRL GRO,2

(3) SLL GRO,1

(4) GR1,1,GR1

(5) ST GRO,PTN.GR1

试题 7（1994 年试题 6）

在 COMET 型计算机上可以使用试卷上所附的 CASL 汇编语言。阅读程序说明和 CASL 程序，把应填入（n）处的字句，写在答卷的对应栏内。

［程序说明］

子程序 DIVIDE 将真分数 a/b(0<a<b)转换成 N 位小数(0<N≤20)，并输出结果。

子程序从栈中接受参数。参数的入栈顺序为：分子（a)、分母（b）和小数位数（N)。进入子程序时，已保证 0<a<b 且 0<N≤20 子程序不再对参数作合法性检查。

由于需要将结果输出，所以子程序中计算出每一位小数后直接用字符方式顺次存放在标号 OUTPUT 之后的存储区域中。数字字符 0~9 的 ASCII 编码依次是 48~57。

[程序]

```
        START DIVIDE
```

```
DIVIDE  LEA GR3, 2
        LD GR1, 1, GR4
        LEA GR1,2 GR1
        ST GR1, OUTLEN
        LD GR0, (1)
DIGIT   LEA GR2, 48
        SLA GR1, 1
        ST GR0, WORK
        SLA GR0, 2
        ADD GR0,WORK
SUBST   CPA GR0, (2)
        JMI GOOD
        (3)
        (4)
        JMP SUBST
GOOD    ST   GR2,   OUTPUT,
        GR3
        LEA GR3, 1 GR3
        SLL (5)
        JZE OUT
        CPA GR3, OUTLEN
OUT     JMI DIGIT
        ST GR3,OUTLEN
        OUT          OUTPUT,
OUTPUT OUTLEN
        RET
OUTLEN DC '0'
WORK    DS '20'
        DS 1
        DS 1
        END
```

[解析]

阅读试题说明，首先确定子程序从栈中接收参数时各个参数的地址，我们能够看出，a、b、N 的地址分别是 GR4＋3、GR4＋2、GR4＋1。

仔细研究程序，发现 N 加 2 之后赋给 OUTLEN，这是因为输出时需要输出"0"。填空（1）所在的语句究竟要给 GR0 中装入什么内容呢？应该是 GR4＋3 或 GR4＋2。继续阅读子程序，发现 GR0 的内容在以下的语句中被乘以 10，从一般的编程常识来看，在求真分数　时被乘以 10 的大多是分子，当然，这仅仅是常识而已，至于是否在本题中适用，还要继续推敲才能够确定。我们暂且认为填空（1）的解答是"3，GR4"，看下面的程序是否能够证明这种推断的正确性。

填空（2）究竟将 GR0 与什么进行比较？从后续语句来看，如果 GR0 的值小于被比较的值，就跳转互标号 GOOD。仔细阅读标号 GOOD 及其以后的语句，不难发现这是存储一个小数位交判断是否结束子程序进行输出的语句段。那么（2）中被比较的内容应该是分母（GR4

＋2），所以填空（2）的解答就是"2，GR4"。
从程序结构来看，（3）和（4）应该完成求得一位小数的操作。CASL 中没有除法指令的，一般采用减法来实现除法操作。从后面的语句"格 ST　GR2，　OUTPUT　，GR3"来看，试题中求得的小数位存储在 GR2 中，这样我们就不难构造（3）和（4）中的解答来完成除法操作了，即"SUB　GR0，　2，　GR4"和"LEA　GR2,1，GR2"。填空（5）所在的语句是判断当某个值为 0 是结束子程序，那么这个值是什么呢？显然是当分子为 0 时应该结束子程序，所以应当填写"GR0,0"。
［答案］
（1）　3，GR4
（2）　2，GR4
（3）　SUB　GR0,2，GR4
（4）　LEA　GR2，1，GR2
（5）　GR0,0

试题 8（1993 年试题 6）

在 COMET 型计算机上可以使用试卷上所附的 CASL 汇编语言，阅读程序说明和 CASL 程序，把应填入（n）处的字句，写在答卷的对应栏内。
［程序说明］
子程序 MOVE 是将地址为 A 开始的 N 个存储单元移动到地址为 B 开始的 N 个存储单元中，对于两个区域互相重叠的情况也能正确处理。
主程序在 GR1 中给出存放子程序所需参数的起始地址，参数的存放形式为：

```
(GR1)+    A
0         B
+1        N
+2
```

［程序］

```
MOVE    START
        LD GR2,1, GR1
        LD GR3, 0, GR1
        CPL GR3, 1, GR1
        JZE ENDMOV
        (1)
        (2)
        JMP SAVE
LT      (3)
        LEA GR3, (4)
        (5)
        LEA GR3, (6)
        (7)
SAVE    ST GR0, WORK
        LD GR1, 2, GR1
LOOP    LD GR0, 0, GR3
        ST GR0, 0, GR2
```

```
        ADD GR2, WORK
        ADD GR3, WORK
        LEA GR1, -1, GR1
ENDMO   JNZ LOOP
V       RET
WORK    DS 1
        END
```

[解析]

细阅读程序说明，注意程序说明中对两个区域重合情况的特别强调，注意 GR1 中给出的是子程序所需参数的地址，而非参数本身，理解了这一点对整个子程序的理解非常重要。

从子程序的开头部分阅读，"START"后面 4 条语句的作用是易于理解的，前前句把 B 的起始地址装入 GR2，把 A 的起始地址装入 GR3，第 3 句比较 A、B 起始地址的大小，第 4 句处理 A、B 起始地址重合的情况，在这种情况下可以不必移动。我们可以先不考虑接下来的填空，先对程序的整体结构进行分析。很明显，在"JZE　ENDMOVE"之后到标号 SAVE 之前没有发生移动存储单元的动作，因为"LD　GR1,2，　GR1"到"JNZ　LOOP"之间完成了一次 N 个存储单元的传送过程，由此我们推断

本题的 7 个需要填写都集中在处理 A、B 地址不同时的传递控制参数设置上。

要处理好 A、B 地址重叠的情况，必须考虑两种情况，即当 A 大于 B 时，作顺序传送，这样即使 B＋N＞A，也能保证传送处理的正确性；当 B 大于 A 时，作反序传送，这样即使 A＋N＞B，也能保证传送处理的正确性。

我们先看填空（1）和（2）之后的"JMP　SAVE"，考查在直接跳转到标号 SAVE 时的操作。跳转后的第一条语句是"ST　GR0，　WORK"，本子程序中尚无对 GR0 操作的语句，如果此前没有对 GR0 的操作，程序的结果将是不可预知的。要保证程序的正常运行，在"JMP SAVE"之前必须有对 GR0 进行操作的语句，现在的问题是在什么位置进行何种操作。继续阅读子程序，易知"JMP　SAVE"发生在 A 大于 B 的情况下。标号 SAVE 后的语句完成从 A 到 B 的搬运，" WORK"（GR0）是作为传送地址移动而准备的。由此推断从标号 LT 到标号 SAVE 之间的语句是为反序传送进行传送参数设置。

有了以上的分析，（1）和（2）的答案就是显而易见的了，填空（1）应该是在 A 小于 B 时的跳转，跳转的目标应该是"LT"。而填空 (2) 的作用就应该是对 GR0 的操作，从"ST　GR0"、"WORK"及"WORK"作为地址修改参数两点来看，这里应该是"LEA　GR0,1"，以保证顺序传送的进行。

既然标号 LT 和标号"SAVE"之间是为反序传送设置传送参数，反序传送时传送的源数据区是从 A＋N－1 开始（注意存储区的地址计数是　1 从 0 开始的），而传送的目的数据区则是从 B＋N－1 开始，地址修改参数应该是－1。结合以上分析，在程序执行到标号 SAVE 之前时，GR3 的内容应该是 A＋N－1，GR2 的内容应该是 B＋N－1，GR0 的内容应该是－1。在填空（3）到填空（7）之间完成上述操作解决方案是灵活的，我们可以在（3）或（7）处填写"LEA　GR0，－1"以完成对 GR0 的设置，但考虑程序的可读性，将"LEA　GR0，　－1"填写在（7）处比较好些。蓁其余的 4 条语句就是完成对 GR2 和 GR3 进行设置的 A＋N－1 和 B＋N－1，结合程序说明，完成该操作的语句应该很明确了。

本题解答关键在于理解填空（1）到（7）之间语句的作用和反序传送控制参数的主。设置，如有一方面的内容不能正确理解，就不能得出正确的解答。本题需要解答的部分比较集中，关键是我们要结合程序的整体结构，确定需要填写的部分所承担的功能，从实现程序功能入手，确定需要填写的语句。如果不结合程序整体结构进行考虑，面对填空多于语句的试题，我们就会无所适从。

［答案］
（1） JMI　LT
（2） LEA　GR0,1
（3） ADD　GR3,2　GR1
（4） －1　GR3
（5） ADD　GR2,2，GR1
（6） －1　GR2
（7） LEA　GR0，－1

试题 9（1992 年试题 6）

在 COMET 型计算机上可以使用试卷上所附的 CASL 汇编语言，阅读程序说明和程序，回答问题，把解答写在答卷的对应栏内。

［程序说明］

本程序根据输入的姓名（字符串），在已有的线性璷中查找其相应的通讯地址（字符串），并予以输出。

标号为 DATPTR 的存储字存放着线性表第一个结点的地址。结点的数据结构定义如下：

第 K 个结点

+0	→指向第 K+1 个结点
+1	存放姓名字符串
+2	存放通讯地址字符串长
+3	度
	自此存放姓名和通讯地址

最后一个结点的指针字段为空指针（内容为 0）。

［程序］

```
          START GEGIN
OTLONG    DS 1
OTTEXT    DS 80
NOLONG    DC 10
NNOTEXT DC 'NOT FOUND'
INTEXT    DS 80
INLONG    DS 1
DATPTR    DC FIRST
BEGIN     IN INTEXT, INLONG
          LEA GR0,0
          CPA GR0, INLONG
          JZE ENDSCH
          LEA GR3, DATPTR
NEXTME    (1)
M         (2)
          JZE NOTFND
          LD GR0, 1, GR3
          CPA GR0,INLONG
```

```
        JNZ NXMEM
        (3)
        LEA GR2, INTEXT
        CALL CMPSTR
        (4)
        LD GR0, 2, GR3
        ST GR0, OTLONG
        (5)
        (6)
        LEA GR2, OTTEXT
        CALL CPYSTR
        OUT      OTTEXT,
        OTLONG
NOTFND  JMP BEGIN
        OUT      NOTEXT,
ENDSCH  ONLONG
CMPSTR  JMP BEGIN
CMPNXT  EXIT
        PUSH 0, GR3
        LD GR3, 0, GR1
        CPL GR3, 0, GR2
        JNZ CMPEND
        LEA GR1, 1, GR1
        LEA GR2, 1, GR2
CMPEND  SUB GR0, CONST1
        JNZ CMPNXT
CONST1  POP GR3
CPYSTR  RET
CPYNXT  DC 1
        PUSH 0, GR3
        LD GR3, 0, GR1
        ST GR3, 0,GR2
        LEA GR1,1,GR1
        LEA GR2, 1,GR2
        SUB GR0, ONE
        JNZ CPYNXT
        POP GR3
        RET
        ONE DC 1
        END
```

[解析]

钥匙首先要弄清楚程序中所使用的数据结构，这是理解程序处理流程的基础。由程序说明我们应该能够发现该程序使用的是链表。链表的表头由 DATPTR 指示，每个结点存储 5 部分信息，即第 K＋1 个结点的存储地址、姓名字符串长度、通信地址字符串长度、N1 个存储

字存放姓名、N2 个存储字存放通信地址，而最后一个结点的第一个字节的内容是 0，这是链表未尾的标志。程序中进行的比较查找就是以这个链表为基础的，理解了链表结点的结构，理解程序的处理流程就会比较容易了。然后再对程序的整体结构进行考查，发现程序由主程序和两个子程序组成，我们要解答的问题都在主程序内。从子程序的名字上大致可以判断它们的功能，仔细考查，确认 CMPSTR 完成字符串的比较，CPYSTR 完成字符串的拷贝，一定要雇它们都使用哪些寄存器工作，各寄存器的作用是什么。

阅读程序，在明显的 4 组变量定义之后，程序开始进行输入操作及输入后简单的输入内容检查，这是一些无关大局的细节问题。接下来的"LEA　GR3，DATPTR"让我们得知 GR3 的内容就是链表表头的地址。

标号 NXTMEM 后面的两填空让人感到题目的难度，但结合变量的含义仔细推敲，"JZE NOTFND"的含义是，如果上个语句的结果使 FR 值为 00，则宣布没有查找到符合要求的结点。查找链表则只能在查找到链表未尾时才可以宣布结束，而在这里链表结束的标志是结点的第一个字节内容为 0。经考查，断定此处两空白应该完成装入结点首字节和判断结点首字节的任务，但目前几乎每个寄存器都是可用的，究竟使用哪个寄存器，我们暂且放下，继续阅读程序。后续的语句是比较输入字符串的长度与该结点的长度是否相等，此处使用 1，GR3 来表示姓名长度，则易知上面的操作应该使用 GR3。这里需要注意的是语句"GR3，SATPTR"有稳操胜券不敢轻易使用 GR3，但结合后续语句可以判定不使用 GR3 就难以实现结点指针的移动。

在姓名长度相等的情况下，应该调用 CMPSTR 比较咀输入字符串相等，很明显，CMPSTR 需要用 GR0、GR1 和 GR2 传递参数，其中 GR0 传递需比较的字符串的长度，GR1 和 GR2 分别传递需要比较的字符串的地址。GR0 是无须再设置的，GR2 的内容由"LEA　GR2，INTEXT"来设置，且明确了 GR2 传递的是输入字符串的地址，那么填空（3）处就应该是设置 GR1 的内容为结点姓名字符串的地址，答案是明显的。CMPSTR 的调用后面紧跟着一个填空，我们可以暂且忽略它，继续阅读下面的程序。下面的程序先是调用 CPYSTR，然后双输出字符串，这是找到符合条件的结点之后的工作。由此我们判断，填空处理 CMPSTR 认定当前结点并非查找结果的情况，应该是跳转到 NXTMEM，但如何表示呢？考察子程序 CMPSTR，发现跳转受"SUB　GR0，CONST1"控制，因为若认定当前结点符合条件，则一定是比较到字符串的末尾，"SUB　GR0，　CONST1"应该使 FR 的值为"00"，则 FR 值不为"00"时发生跳转，所以此处应该填写"JZN　NXTMEM"。

填空（5）和（6）后面是对子程序 CPYSTR 的调用，结合我们对 CPYSTR 的理解，易知该子程序使用 GR0、GR1 和 GR2 传递参数，GR－传递需拷贝的字符串的长度，GR1 拷贝对象的地址，GR2 传递拷贝目的字符串地址。很明显这里要拷贝的应该是通信地址，GR0 和 GR2 的设置已经完成，需要把通信地址字符串的地址放进 GR1，　考查链表结点的结构，不难发现通信地址字符串的确切地址必须由姓名地址加姓名长度求得，所以就很难用一个语句来完成工作，并且此处出没有其他得的工作要做，所以我们在这里填写"LEA　GR1，　3，GR3"和"ADD　GR1,1，GR3"，　结点的通信地址字符串首地址送到 GR1。

由以上的分析解答可以总结出两点可以称作经验的东西：1、要切实理解试题说明；2、要切实理解程序流程和结构。

［答案］

(1) LD GR3, 0, GR3

(2) LEA GR3, 0, GR3

(3) LEA GR1, 3, GR3

(4) JNZ NXTMEM

(5) LEA GR1,3,GR3

(6) ADD GR1,1,GR3

试题 10（1991 年试题 6）

在 COMET 型计算机上可以使用试卷上所附的 CASL 汇编语言，阅读程序说明和程序，回答问题，把解答写在答卷的对应栏内。

［程序说明］

本程序将存储字 A 中的第 n 位起的 n 位二进制位搬至存储字 B 的第 b 位起的 n 位中。

1、 主程序在 GR1 中给出子程序所需的参数起始地址。

2、 子程序所需的参数个数和含义如图 7－1 所示。

3、 0≤a≤15; 0≤b≤15;1≤n≤16; a+n≤16;b+n≤16。

4、 存储字 B 中，除中开始的 n 位外的其他二进位应保持原值。

［问题］

在程序中（1）～（8）处各填入一条正确的指令，以完成此程序。除非必要，标号栏还要填写。

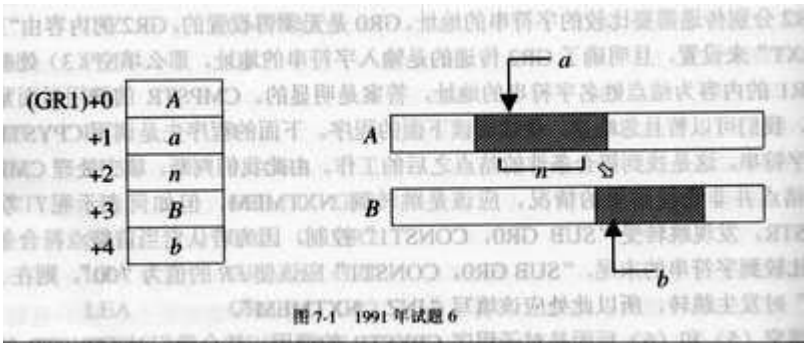


图 7-1　1991 年试题 6

［程序］

```
MOVBIT  START
        LD GR2, 2, GR1      SAVE1 中形成
          (1)               11…00……0
          (2)                   ↓
        ST GR2,   SAVE1 n   个
        LD GR2,1，GR1
          (3)
        LD GR2,0，GR1
        AND GR0,0，GR2
        LD GR2,4，GR1
          (4)
        LD GR2,1，GR1
        SUB GR2,4，GR1
          (5)
        JMP SHFTE
SHFTR   SUB GR2,1，GR1
          (6)
SHFTE   ST GR0，SAVE2
        LD GR2，SAVE1
```

```
        LD GR3,4，GR1
        （7）
        （8）
        LD GR3,3，GR1
        AND GR2,0，GR3
        OR GR2，SAVE2
        ST GR2,0，GR3
C8000   RET
CFFF    DC  #8000
SAVE1   DC  #FFFF
SAVE2   DS 1
        DS 1
        END
```

［解析］

仔细阅读程序说明，把握程序要实现的功能和试题所提供的信息。重点是要理解 GR1 中给出了子程序所需参数的起始地址，并从 GR0 开始的各内存单元所存储的内容。然后考查子程序的变量定义部分，根据变量值的特点估计它们的用途。

子程序的开始部分需要填写的内容比较多，虽然有一句注释，但对我们理解程序帮助不大。在这种情况下我们应该跳过这些语句，直接阅读填空（3）之后的部分，看看（3）之后进行了什么操作，则易知在（3）之前已经进行了哪些操作。从"LD　GR2，0，GR1"读起，推断到该句和"AND　GR0,0，GR2"两句，程序应该是已经完成了从 A 中取出从 a 位起的 n 位，因为在此之后的工作已经是比较 a 和 b 的大小　，这是进行对准数位的前奏，一旦对准数位，搬运工作就开始了。这样，到填空（3）处，GR0 中的内容应该具有这样的特点，即从该字的第 a 位起，有连续 n 个 1，其余的数位全部是 0。考察已有的 3 条语句，没有完成此项功能的相应语句，不过"LD　GR2,2，GR1"和"LD　GR2,1，GR1"两条语句也能给我们许多有用的信息，要想在 GR0 中形成符合要求的内容，必须先生成一个在开头或结尾具有 n(GR1+2)位 1、其余位全部为 0 的字，然后再将其逻辑左移或右移若干位（由 GR1＋1 决定）。综合考虑到填空（4）之前没有其他要完成的任务，我们推断（1）至（3）完成上述操作，即在 GR0 中形成从第 a 位起有连续 n 位的 1 的存储字。结合对变量的，我们在 3 处分别填写"LD GR0，C8000"、"SRA　GR0，－1，GR2"和"SRL　GR0,0，GR2"。

跳过"CPA　GR2,1，GR1"后面的空白，继续阅读后面的语句，不难确定到标号 SHFTE 之前完成的是对位工作，即将从 A 中取出的从第 a 位起有连续 n 个 1 的字变成从第 b 位起有连续 n 个 1 的字。很明显，填空（4）之后直到"JMP　SHFTE"完成的是 a 大于 b 的对位工作，而标号 SHFTE 及后续的一句完成 a 小于 b 的对位工作。两条 SUB 语句已经在 GR2 中生成了需要移动 GR0 的内容的位数。当 a 大于 b 时，GR0 内容左移，否则右移，填空（5）和（6）的答案非常明显。现在回头去考察（4），（4）的跳转功能已经非常突出，即在 b 大于 a（a 小于 b）时跳转到标号 SHFTR 所在语句。

要想把 GR0 的内容搬运到 B 中，还必须将 B 从第 b 位起的连续 n 位都置 0，否则后面的"OR GR2，SAVE2"就难以完成任务，结合"OR　GR2，SAVE2"所提供的信息，可知将 B 从第 b 位起的连续 b 位都置 0 的字应该生成在 GR2 中。由此上溯，易知填空（7）和（8）完成的是在 GR2 中形成的除第 b 位起的 b 位为 0 外，其余数位全部为 1 的字。综合考虑用过的和未用过的变量及应该填写的语句数量同，此处的答案为"SRL　GR2,0，GR3"和"EOR　GR2，CFFF"。

总结解析该题的经验，当我们无法按语句顺序理解程序时，可以先阅读后面的语句，确定不易理解的部分应该完成哪些任务，然后结合实际情况，综合考虑程序的整体，排队错误判断之后解答问题，结合上下文在规定的位置填写语句，完成后续工作需要完成而又没有完成的任务。本题中的（1）、（2）、（3）与（7）、（8）的解答过程就是这样的，我们可以称其为"倒卷珠帘"。

［答案］

(1)LD GR0, C8000

(2)SRA GR0，－1，GR2

(3)SRL GR0,0，GR2

(4)JPZ SHFTR

(5)SLL GR0, 0, GR2

(6)SRL GR0, 0, GR2

(7)SRL GR2, 0, GR3

(8)EOR GR2, CFFF

试题 11（1990 年试题 7）

在 COMET 型计算机上可以使用试卷上所附的 CASL 汇编语言，阅读程序说明和程序，回答问题，把解答写在答卷的对应栏内。

［程序说明］

本程序完成两个 4 位十进制数相加，并输出与会代表两数之和。

例：输入"5794＋6438"输出"12232"。

1、 按上述要求输入，否则输出"INPUT　ERROR"，信息，并重新输入。

2、 从低位开始，逐位进行十进制数相加。

3、 若输入长度为 0 时，本程序结束。

［问题］

在程序中（1）～（7）处逐个填入一条正确指令，以完成此程序。除非必要，标号栏还要填写。

［程序］

```
标号      指令码  操作数
          START
BEGIN     ST      GR4, SPW
RETRY     IN1     INBUF, LENG
          LEA     GR1,0
          CPA     GR1, LENG1
          JZE     HALT
          LEA     GR2, 9
          CPA     GR2, LENG1
          JNZ     ERROR
          LEA     GR3, 4
          (1)
          CPL     GR0, SING
          JZE     PASS1
ERROR     OUT     INERR, LENG2
                  RETRY
```

```
PASS1   JMP     GR1, SM
        LD      GR1, INBUF, GR3
        ST      GR1, 0
LOOP1   LEA     GR2, INBUF, GR1
        LD      GR1,1,GR1
        LEA     GR3, SM
        CPL     ERROR
        JM1     GR3,LM
        CPA     ERROR
        JPZ     GR3, BCD
        AND     0, GR3
        PUSH
        (2)     LOOP1
        JNZ     GR2, CY
        ST      GR1,4
        LEA
        (3)     GR3, (4)
        ADD     GR3, CY
        ADD     GR3, TEN
        CPA     LAB1
        JMI     GR3,SIX
        ADD     GR3,BCD
        ADD
        (5)     LAB2
LAB1    JMP     GR0,0
LAB2    LEA     GR0,CY
        ST      GR3, SM
        OR      GR3,OUTBUF,GR1
        ST
        (6)     LOOP2
        JNZ     GR0,SM
        OR
        (7)     OUTBUF, LENG3
        OUT     RETRY
HALT    JMP     GR4,SPW
        LD
SPW     EXIT    1
INBUF   DS      80
LENG1   DS      1
INERR   DS      'INPUT ERROR'
LENG2   DC      11
SING    DC      '+"
SM      DC      '0'
LM              ':'
```

```
BCD     DC    #000F
CY      DC    1
TEN     DC    10
SIX     DS    6
OUTBUF  DC    5
LENG3   DC    5
        DS
        DC
        END
```

[解析]

首先阅读程序说明，程序说明提供的有效信息不多；然后阅读程序后半部分的变量定义，切实各变量的含义。粗读程序，发现必须在某处标号栏填写标号 LOOP2。从 GEGIN 开始读起，很快就会发现一直到"JPZ　RETRY"是关于输入合法性检查的部分。"LEA　GR3,4"后面是一个我们必须填写的填写，我们暂且跳过两行，看填后面的语句"CPL　GR0，SING"很明显，在检测输入的第 5 个字符是否为"＋"，而且该字符已经被装入 GR0。仔细推敲，可以确定将输入的第 5 个字符装入 GR0 的操作只能是由"CPL　GR0，SING"前面的两行语句完成，因此，"LEA　GR3,4"的作用也就很明显了，这是在为取内存的内容作地址偏移准备。至此，填空（1）的答案也就可以确定了。

继续阅读程序，不难推断从"LEA　GR1,0"到"JNZ　LOOP1"构成一个循环，在跳转之前修改循环计数器属于常识，这里重要的是判断哪个寄存器或变量循环是计数器，循环步长是多少。对程序进行分析，发现该循环仍旧是进行输入合法性检查的，它检查输入缓冲区的每一位是否为"0"－"9"，并将其转换为数字，按输入顺序入栈。应该注意到，标号 PASS1 处已经把输入"＋"变成"0"，所以这里的合法性检查能够顺利进行。既然是对输入缓冲区的每一位进行检查，步长只能为 1；从"JNZ　LOOP1"来看，循环计数器应该是递减的。结合以上对程序的理解，这个递减的变量必须是合法的输入长度，这个长度现在存储在 GR2 中，所以这条修改循环条件的语句就应该是"LEA　GR2，－1，GR2"。

在上面的循环完成以后，两个 4 位数已经按输入顺序入栈，接下来程序的最后一部分应该是进行两数相加。从"LEA　GR1,4"到"JNZ　LOOP2"又是一个循环。但 LOOP2 这个标号到底放在哪儿呢？填空（3）、（5）、（6）都有可能，但仔细分析，产生排除了放在（6）处的可能性。（5）后面是一个无条件跳转语句，以该语句为循环体的第二条语句是不合情理的。这样，标号 LOOP2 只能放在（3）处。分析从（3）到"ADD　GR3，CY"，可以推断它们的任务是完成对应数位的相加。由于前面我们已经把输入内容压栈，而（3）后的语句又有 GR3 参加，我们就可断定（3）是对 GR3 的操作。因为对 GR3 的操作不足以使其承担这样的任务。所以（3）的答案应该是"LOOP2　POP　GR3"。后续语句完成数位的相加，如果在（40 中也使用 CY 的话，共同性恐怕就不是几条语句所能完成的任务了。由于上面已经将所有输入压栈，我们可以通过地址偏移来实现对堆栈内容的操作。从前面压栈过程易知，对应的数位在栈中相隔 5 位，似乎应该在（4）填空"5，GR4"，但如果注意到 POP 指令对堆栈指针的修改，我们就会将"5"改为"4"，这是比较容易理解但比较容易忽略的事情。

接下来的进位检查和进位处理，当和大于 10 时，要产生进位，参考"LAB1 LEA GR0,0"和"ST GR0,CY",易知（5）处应该填写"LEA GR0,1".

一直读到"JNZ LOOP2",未发现循环体中修改循环条件的语句，GR2 的值为 0，是不能和 JNZ 相互配合的，GR3 和 GR4 各胡用途，只有 GR1 尚在闲置，而且其值为 4，正符合我们进行的 4 位数相加的要求，通过对其递减和"JNZ　LOOP2"的判断，完成循环控制是非常方便的。

考察前后语句，未发现与其他工作的冲突，所以我们就确定 GR1 是作为循环计数器的。在循环 LOOP2 结束后，两个 4 位十进制数的和可能会在最高位产生进位，与上面的操作相同，我们要把它转换成字符形式准备输出，理解最后几条语句的含义后，（7）完成将进位存入 LUTBUF 的功能应该是很明显的，所以这里我们填写"ST　GR0，OUTBUF"。

［答案］

（1）　LD　GR0，INBUF，GR3

（2）　LEA　GR2，－1，GR2

（3）　LOOP2　POP　GR3

（4）　4，GR4

（5）　LEA　GR0,1

（6）　LEA　GR1，－1，GR1

（7）　ST　GR0，OUTBUF

七 C 语言程序编制度量精解

高级程序员级 C 语言编程试题有一定的浓度和难度。根据新的考试大纲，每年有两道必考的 C 语言编程试题，考试试题更注重于算法理解和实现方面，在解答试题的过程中，一定要记住要严格按照试题说明提供的算法展开思路，而不是在了解了程序的功能后按照自己推断的算法去理解程序。这一点有必要在复习备考过程中加以重视，具体的方法就是在阅读试题说明过程中不产生自己的算法中，以理解并接受试题说明的内容为目标。

试题 1（2000 年试题 5）

阅读下列程序说明和 C 代码，就填入（n）处的字句在答卷的对应栏内。

［程序说明］

下列文法可用来描述化学分子式的书写规则（例如，AL2（CO3）3、C u （OH）2）：

λ→β|βλ

β→δ|δn

δ→ξ|ξθ|(λ)

其中 λ 是一个分子式；δ 或是一个元素，或是一个带括号的（子）分子式；元素或是一个大写字母（记为 ξ），或是一个大写字母和一个小写字母（记为 ξθ）；β 或是一个 δ，或是在 δ 之后接上一个整数 n；δn 表示 β 有 n 个 δ 的元素或（子）分子式。一个完整的分子式由若干个 β 组成。

当然一个正确的分子式除符合上述文法规则外，还应满足分子式　本身的语义要求。

下面的程序输入分子式，按上述文法分析分子式，并计算出分子式的分子量。例如：元素 H 的原子量是 1，元素 O 的原子量是 16。输入分子式 H2O，计算出它的分子量为 18（1＊2＋16）。程序中各元素的名及它的原子量从文件 atom.dat 中读入。

［程序］

```
#include <stdio.h>
#include <string.h>
#define MAXN 300
#define CMLEN 30
struct elem {char name[3]; /*  元素名  */
Double v; /*  原子量  */
}nTbl [MAXN];
char cmStr [CMLEN], *pos;
int c; FILE *fp;
double factor();
double atom() /*  处理文法符号  */
{ char W[3];int i; double num;
while ((c=*pos++)= =''||c= ='\t'; /*  掠过空白字符  */
if (c= ='\n')
return 0.0;
if(c>='A'&& c<='z') { /*  将元素名存入 W    */
w[i=0]=c; c=*pos++;
If (c>='a'&& c<='Z') w[++i]=c;
```

```
Else pos--;
W[++i]='\0';
For (I=0;nTbl[i],v>0.0;i++)
If(strcmp(w,nTbl[i],name)= =0)
return nTbl[i],v;
Prinft("元素表中没有所输入的元素：\t%s\n",w);
return -1.0;
} else if (c= = '(') {
if((num= (1) )<0.0)
return -1.0; /*  包括可能为空的情况  */
if(*pos++!=')')
{ printf ("分子式中括号不匹配！\n"); return -1.0
return num;
}
prinft("分子式中存在非法字符：\t%c\n",c);
return -1.0;
}
double mAtom() /*  处理文法符号 β */
{
double num; int n=1;
if((num= (2) )<0.0)
return -1.0;
c=*pos++;
if(c>='0'&& c<='0')
n=0;
while 9c>='0'&& c<='9')
{
n= (3) ;
c=*pos++;
}
}
pos--;
return num *n
}
double factor() /*  处理文法符号 λ */
{
double num=0.0,d;
if((num=mATom())<0.0)
return -1.0;
while)*pos>='A'&& pos<='Z'||pos= ='('
{
if((d= (4) )<0.0)
return -1.0;
(5) ;
```

```
}
}
void main()
{
char fname[ ]="stom.dat"; /*  元素名及原子量文件*/
int i;double num;
if((fp=fopen(fname,"r"))= =NULL) /*  以读方式打开正文文件  */
{
prinft("Can not open %s file.\n",fname);
return; /*  程序非正常结束  */
}
i=0;
while (i<MAXN && fscanf (fp,"%s%1f",nTb1[i].name,& nTb1[i].v= =2)
i++;
fclose(fp);
nTb1[i].v=-1.0;
while(1)
{ /*  输入分子式和计算分子量循环，直至输入空行结束    */
prinft("\n 输入分子式！（空行结束）\n";gets(cmStr);
pos=cmStr;
if(cmStr[0]= ='\0')
break;
if(num=factor())>0.0)
if(*pos!='\0')
printf("分子式不完整！\n");
else
printf("分子式的分子量为 f\n",num);
}
}
```

[解析]

首先要仔细阅读并理解程序说明中所提供的文法规则。只有充分掌握这些规则，解答才会有规律可循。在下面的解析中我们将始终结合方法规则来解答试题。

粗读程序，不难发现程序中对 3 个函数的调用都很简单，由于字符串指针 pos 是全局变量，各函数在进行处理时直接对 pos 进行处理，然后返回自己所计算的那一部分分子式的分子量。

对程序的阅读要从主函数 main()开始。从第 51 行到第 64 行都很好理解，一直到第 65 行才能看到对函数 factor()的调用。很显然，在调用函数 factor()后，返回值 num 即是分子式，当然，在分子式不合乎文法规则时， num 是一个负数。那么我们就开始研读函数 factor()。

我们先看关于的文法规则：λ→β\βλ。这就是说，λ 或者是由 β 构成，或者是由构成。处理的时候应该区分这两种情况，即处理完了 β 之后应该判断其后是否有 λ，如果有 λ，应对其进行处理。

那么程序中是如何表述的呢？很明显，第 45 行处理了 λ 起始部分的 β。在没有涉及到函数 mStom() 更多内容时，我们只要知道 mAtom()的返回值（分子量或者是表示分子式有问题的－1.0）赋给 num 就可以了。第 46 行的 while 循环条件如何理解？结合以上对文法规则：、

的分析，我们不难判断，这是判断其后是否还有一个 λ 或 β。第 47 行显然应该是一个函数调用语句（比较第 65 行和 45 行，不难发现本程序中调用函数的常用语句形式）。这里应该调用什么函数？处理 λ 和 β 的过程基本相似，这里可以调用函数 factor()，也可以调用函数 mAtom()。

基于以上的理解，在函数调用成功的情况下，整型变量 d 的值应该是函数调用返回的分子量。从第 49 行我们可以判定，函数 factor()中整型变量 num 是用来存储分子量的。那么第 47 行的函数调用返回值就应该累加到 num 中，这是一个简单的操作，其 C 语言实现也很容易。

在研读函数 mAtom()之前，我们先看看关于 β 的文法规则：β→δ|δn。这就是说，β 或者由 δ 构成，或者由 δ 和一个整数 n 构成。这样，在处理完了 δ 后，一定要确认其后是否跟着一个整数 n。

不难判断，第 32 行的填空（3）处需要填写的是对某个函数的调用（与第 47 行的判断文法相同）。但是这个函数是谁呢？如果我们注意文法的相同之处，结合函数 factor()的结构，不难判断这里是调用处理 δ 的函数，即函数 atom()。将 δ 的分子量计算出来赋给 num。注意，这里的判断还需要在以后进行确认，目前我们暂且这样认定。

从第 34 行开始，熟悉 C 语言的程序员都能够看出这是在判断 c 是否为一个数字字符。显然，如果是一个数字字符，就应该把它转换成数字。从第 41 行可以看出，转换以后的数字存储在变量 n 中。这样，第 36 行的语句就是完成字符到数字的转换的。而且从此处使用一个循环结构可以看出这里可以处理若干位的数字。从数字字符向数字的转换应该是很容易的，在这里，就应该描述为"n*10+c-'0'"。

回过头来仔细阅读这个函数，在实现程序功能方面没有潜在的逻辑错误，我们能够确认这样填写是符合要求的。

现在看函数 atom()。该函数是处理文法符号 δ 的。文法符号 δ 的文法规则是 δ→ζ|ζθ|(λ)。不妨借助解读上两个函数的文法来解读该函数。从第 15 行到第 21 行处理了前两种情况，即 ζ 是一个大写字母或一个大写字母和一个小写字母。到第 22 行，从判断 c 是否为'（'，可以看出这是处理第 3 种情况了。如何处理呢？第 23 行是一个函数调用（判断文法同上），由于在'（'之后是一个 λ，显然，此后的函数调用应该是处理 λ 的函数，即函数 factor()。

这里的解题过程非常简单，即将程序与文法规则紧密结合，通过文法符号的处理来解题。首先要将程序中的语句与文法规则对应起来，然后就能够理解程序中的处理流程了。

［答案］

(1) factor()

(2) atom()

(3) n*10+c-'0'

(4) mAtom()或 factor()

(5) num+=d

试题 2 （2000 年试题 6）

阅读下列程序说明和 C 代码，将就填入（）处的字句在答卷的对应栏内。

［程序说明］

某城市有 n 个车站，并有 m 条公交线路连接这些车站，设这些公交车都是单向的，这 n 个车站被顺序编号为 0 到 n－1。本程序，输入该城市的公交线路数、车站个数和各个公交线路上的各站编号，求得从 0 站出发乘公交车到车站 n－1 的最少换车次数。

程序利用输入信息构建一张有向图 G（用邻接矩阵 g 表示），有向图的顶点是车站，若有某条公交线路经 i 站能到达 j 站，就在顶点 i 到顶点 j 之间设置一条为权 1 的有向边＜i,j＞。如果是这样，从站点 x 到站点 y 的的最少上车次数便对应图 G 中从点 x 到点 y 的路径长。而

程序要求的换车次数就是上车次序减 1。

［程序］

```
#include <studio.h>
#define M 20
#define N 50
int a[N+1]; /*  用于存放一条线路上的各站编号  */
int g[N][N]; /*  存储对应的邻接矩阵  */
int dist [N]; /*  存储站 0 到各站的路径  */
int m,n;
void buildG()
{
int i,j,k,sc,dd;
prinft("输入公交线路数\n");
scanf("%d%d",&m,&n);
for(i=0;i<n;j++) /*  邻接矩阵清 0 */
for(j=0;j<n;j++)
g[i][j]=0;
for(i=0;i<m;i++)
{
prinft ("沿第条公交线路并进方向的各站编号
(0<=编号<=%d,-1 结束);\n",i+1,n+1);
sc=0; /*  当前线路站计算器  */
while(1)
{
scanf("%d",&dd);
if(dd= =-1)
break;
if(dd>=0 && dd<n)
(1)
}
a[sc]=-1;
for(k=1;a[k]>=0;k++) /*  处理第 i+1 条公交线路  */
for(j=0 ; j<k ; j++)
g (2) =1;
}
}
int minLen()
{
int j,k ;
for (j=0; j<n; j++)
dist [j]=g[0][j];
dist[0]=1;
do{
for (k=-1,j=0;j<n;j++) /*  找下一个最少上车次数的站  */
```

```
if (dist[j]>0 && (k= =-1 | | dist[j]<dist[k])) k=j;
if(k<0 || k==n-1) break;
dist[k] = -dist[k]; /*设置 k 站已求得上车次数的标志*/
for(j=1;j<n;j++)/*调整经过站能达到的其余各站的上车次数*/
if( (3) && (dist[j] == 0 || dist[k]+1 < dist[j]))
dist[j] = (4) ;
}while(1);
j=dist[n-1];
return (5) ;
}
void main()
{
int t;
buildG();
if((t=minLen()) < 0) printf("无解!"\n);
else printf("从 0 号站到%d 站需要换车%d 次\n",n-1,t);
}
```

[解析]

阅读程序说明，其中的第二段说明了两件事情：（1）有向图 G 中的数据是如何说明两站之间有无公交线路的，"若有某条公交线路经 i 站能达到 j 站，就在顶点 i 到顶点 j 之间设置一条权为 1 的有向边<i,j>"；（2）如何确定最少换车次数，"从站点 x 至 y 站点的最少上车次数便对应图 G 中从点 x 至点 y 的最短路径长度。而程序要求的换车次数就是上车次数减 1"。

综合程序结构，可以看到程序由 4 各部分组成：（1）文件包含和全局数据定义；（2）用于建立有向图 G 的函数 buildG()；（3）用于求出从站点 0 到站点 n-1 最少换车次数的函数 minLen()；（4）主函数 main()。

结合注释，我们可以从第 1 部分中明确 3 个数组的含义，由此确定他们在程序中的应用。数组 a[N+1]用于存放一条公交线路上的个站编号；数组 g[N][N]存储对应的邻接矩阵；dist[N]用于存储站点 0 到各站点的最短路径。下面我们来分析函数 buildG()。

在函数中用到的变量有 m、n、i、j、k、sc、dd 等。由第 10 行和第 11 行可以看出，m 表示公交线路数，n 表示公交站数；由第 17 行可以看出，sc 为当前线路站点计数器；其余 i、j、k 的不过用作循环变量而已。至于 dd 的用途，一时还难以确定。

结合注释部分阅读程序，从第 10 行到第 13 行都是很好理解的。从 14 行起的 for 循环，结束于第 27 行，构成本函数的主体，而且此后即是函数的结束，由此可以判定该循环完成输入各条线路并建立有向图的任务。

建立有向图，应该包括两个步骤，一是输入一条路径上的各个顶点，二是将输入路径的信息反映到邻接矩阵中去。将以上两个步骤套在一个循环中，反复执行即可完成有向图的建立。

研读第 14 行至第 27 行的 for 循环，可知在处理完 m 条公交线路后结束循环。分析循环体，这部分的结构是清晰的，除第 15、16、17 和 23 行外，由两个内层循环构成，即第 18 行至第 22 行的 while 循环和第 24 行到 26 行的 for 循环。

先研究 while 循环。由第 15、16 和 17 行与输入语句可知该循环完成一条公交线路上各个站点的输入。从第 19 行可知，dd 表示的是一个公交站点的编号。跳过第 20 行，我们看到这里有的"dd>=0 && dd<n"条件，这进一步印证了上面我们关于 dd 的判断。那么，既然 dd 是一个站点编号，在输入站点编号之后，我们首先要将其记入公交线路 a[ ]中，这里只有填空（1），别无语句，我们只好在这里填写完成该任务。同时还要注意到，作为当前线路站计数

器，在 while 循环中没有对其进行修改的语句，所以在记入一个公交站点时，应相应的修改 sc。鉴 sc 的初值是 0，我们必须在这里填写"a[sc++]=dd"，如果使用"++sc"将会使 a[0]无值，也就是说公交线路的第一个站点将被遗漏。

借助注释，不难确定从第 24 行到 26 行的 for 循环完成将该条公交线路站点信息反映到邻接矩阵中的任务。考虑到公交线路是单向运行的，所以这里的操作也就是使公交线路上任意站点到其后任意站点之间存在一条权值为 1 的有向边。这样就需要使用一个二重循环，通过外层循环依次处理公交线路上的各个站点，通过内层循环将在站点前各站点与该站点之间设置一条为权 1 的有向边。

很明显，第 26 行就是完成上述内层循环任务的核心语句，这里应该怎样表述呢？从试题中我们可以看到填空（2）需要填写的应该是数组 g[ ] [ ]的下标。结合上面的分析，我们在这里填写"[a[j]] [a[k]]"是很自然的。这里一定要注意公交线路是单向运行的。

可能有的读者要对第 24 行的"k=1"表示困惑，因为这样便失去了处理公交线路的第一个站点 a[0]的机会了。这里仍旧要强调公交线路的单向运行。因为在单向运行的公交线路上，通过该线路上的任何站点都不能到达该线路的第一个站点，所以在循环中就直接从"k=1"开始，跳过线路的第一个站点。

下面我们来研读函数 minLen()。该函数中包含了 3 个填空，值得我们投入更多的精力来解答。关于求有向图顶点到顶点之间最短路径，Dijkstra 有一个标准的算法，我们先在这里描述这个算法，然后结合这个算法来理解程序。

设指定的顶点为 v，把图中顶点集合分成两组，已求出最短路径的顶点集合为第一组，设为 S，其余的尚未最短路径的顶点集合为第二组。按最短路径长度递增次序逐个把第二组中的顶点移入 S 中，直至从顶点 v 出发可以到达的顶点都在 S 中。在这个过程中，需要始终保持从 v 到 S 中各个顶点的最短路径都不大于从 v 到第二组中任何顶点的最短路径长度。另外，为了便于程序处理，要为每个顶点保存一个距离。S 中的顶点距离就是从 v 到此顶点的最短路径长度，第二组中的顶点的距离就是从 v 到此顶点的只包括以 S 中的顶点为中间顶点的那部分还不完整的最短路径长度。

根据以上描述，结合对程序的理解，就不难理解第 38 行到第 40 行的循环所处理的内容了，即调整经过站能到达的其余各站的上车次数。具体分两种情况：1、当存在从站 k 到站 j 的直通公交线路时，如果已经暂时确定了从站 0 到站 j 的最少上车次数(dist[j]>0)，在这种情况下，如果从站 0 经站 k 换车到站 j 的上车次数小于已经确定的从站 0 到站 j 的最少上车次数，则应该将经过站 k 到站 j 的上车次数(-dist[k]+1)确定为目前从站 0 到站 j 的最少上车次数。2。当存在从站 k 到站 j 的直通公交线路时，如果从站 0 到站 j 没有直通公交线路(dist[j]= =0)，应该取经过站 k 到站 j 的上车次数(-dist[k]+1)为目前从 0 站到站 j 的最少上车次数。

将以上的分析翻译为 C 语言语句，那么填写（3）和填空（4）的解答就非常明显了，它们分别为"(dist[j]>=0 && g[k][j]= =1"和"-dist[k]+1"。细心的读者要问，在第 39 行中已经给出"dist[j]=0"的判断了，在填空（3）中还有必要再反映这一点吗？这是有必要的，如果我们在填空时不反映这一点，那么当 dist[j]==0 时将不进行调整处理，显然是不符合要求的。既然有了要填空的"dist[j]==0"，第 39 行中已经有的那个"dist[j]==0"还有存在的必要吗？答案是肯定的，这里就不再详细解释了，读者可以自己理解。

至于填空（5），不过是函数最后的回送返回值的语句而已。结合第 48 行和第 49 行，发现返回值有两种情况：1、无解（t<0）；2、有解（t>0）。显然在填空（5）中要处理好这两种情况。

要处理好这两种情况，必须从 k 的值入手。我们看第 36 行，可知在循环结束时 k 的值有两种情况，一是 k<0，二是 k=n-1。显然，k<0 是程序无解的标志。那么我们在这里就应该在 k<0 让函数返回一个负数，否则就返回从站 0 到站 n-1 的 0 最少换车次数（dist[n-1]-1）。因

为在第 42 行中已经将 dist[n-1]赋给 j，所以我们可以构造填空（5）的答案为"k<0?-1:j-1"。解答完本题，我们应该吸取的经验就是要切实掌握各种算法，同时要掌握各种算法在实际中的应用，通过程序语句与算法来解答题目。在过去几年中，最后一道 C 语言程序填空试题几乎都是关于回溯法的程序，其实也是关系到算法的掌握和应用。切实掌握了算法的实现，就可以非常容易地将程序（或函数）划分为几个部分，各部分都实现算法中的某一步骤。这样，结合试题的具体情况就能够快速写出正确的答案。

[答案]

（1）a[sc++] = dd

（2）[a[j]][a[k]]

（3）dist[j] >=0 && g[k][j] == 1

（4）-dist[k]+1

（5）k<0?-1:j-1

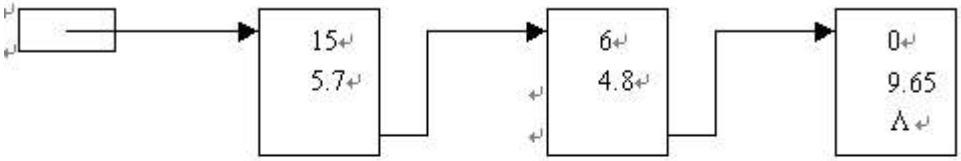
试题 3（1999 年试题 5）

阅读下列程序说明和 C 代码，将应填入（n） 处的字句写在答卷的对应栏内。

[程序说明]

本程序实现两个多项式像乘。多项式用链表表示，链表上的各表元按多项式的幂指数降序链接。例如：

$$f(x)=5.7x^{15}+4.8x^{6}+9.65$$



设两个多项式 $f(x)$和 $g(x)$分别为：

$$f(x)=f_n x^n+\Lambda+f_1 x+f_0 \qquad g(x)=g_m x^m+\Lambda+g_1 x+g_0$$

其积多项式为 $s(x)=f(x)g(x)=s_k x^k+\Lambda+s_1 x+s_0$

其中 $k=n+m, si=\sum_{u+v=i} f_u * g_v$　　（$0\leqslant i\leqslant k$）

[程序]

```
#include <stdio.h>
#include <malloc.h>
typedef struct elem{int index;double coef;struct elem *next
}POLYNODE;
void write(POLYNODE *g)
{ POLYNODE *p=g;
while(p){ printf("%8.4f",p->coef);
if(p->index) printf("*^%d",p->index);
if(p->next && p->next->coef>0) printf("+");
p = p->next;
```

```
}
printf("\n\n");
}
main()
{ POLYNODE *f,*g,*s,*inpoly(),*polymul();
f=inpoly();g=inpoly();s=polymul(f,g);write();
}
POLYNODE *reverse(POLYNODE *g)
{ POLYNODE *u=NULL,*v=g,*w;
while(v){ w=v->next;v->next=u;u=v;v=w;}
return u;
}
POLYNODE *polymul(POLYNODE *f, POLYNODE *g)
{ POLYNODE *fp,*gp,*tail,*p=NULL,*q;
int i,maxindex; double temp;
maxindex = f->index+g->index; g=reverse(g);
for(i=maxindex;i>=0;i--){
fp=f;gp=s;
while(fp!=NULL &&fp->index>i) fp = fp->next;
while(gp!=NULL && gp->index<i-fp->index) gp = gp->next;
temp = 0.0;
while(fp && gp)
if(fp->index + gp->index ==i){
temp +=fp->coef*gp->coef; fp=fp->next; gp=gp->next;
else if( (1) ) fp=fp->next;
else gp=gp->next;
if(temp!=0.0){
q=( POLYNODE*)malloc(sizeof(POLYNODE));
q->index=i; q-coef=temp; q-next=NULL;
if( (2) ) p=q;
else (3) ;
tail =q;
}
}
g=reverse(g);
return p;
}
POLYNODE *inpoly()
{ POLYNODE *u,*v,*h=NULL,*p;
int index; double coef;
printf("Input index(<0 for finish)"); scanf("%d",&index);
while(index >=0){
printf("input coef"); scanf("%lf",&coef);
p=( POLYNODE*)malloc(sizeof(POLYNODE));
```

```
p->index=index; p->coef = coef;
v=h;
while(v!=NULL &&index<v->index) { nu=v;v=v->next;}
if(v==NULL || index > v->index)
{
p->next=v; if(v==h) (4) ;else (5) ;}
else v->coef +=coef;
printf("Input index(< 0 for finish)"); scanf("%d",&index);
}
return h;
}
```

[解析]

这道试题看起来很麻烦，其实不然，程序中科学地使用 3 函数，使程序的结构非常清晰。函数 inpoly()完成输入一个多项式各项的指数和系数，并建立链表的任务。函数 polymul()完成两项式的相乘的任务。现在我们就来分别研究这两个函数。

先看函数 inpoly()，它在输入多项式每一项指数和系数的过程中建立链表。语句"returnu;"完成回送返回值的操作，所以要特别函数中对 h 的操作。

一直到语句"p->coef;"是完成一个结点的输入，此后的"v=h"，因为"v=NULL"，所以这以后又有许多关于 v 是否为 NULL 的判断，足见 v 和 h 的变化对程序功能实现有极大关系。

接下来是一个循环体，如果仔细理解"index<v->index"这个循环条件的含义，就不难确定这是在"v= =NULL"的时候在中寻找当前输入结点的合适位置，以建成一个指数从大到小的链表形式的多项式。那么"v= =NULL"又代表什么？从程序中看，"v= =NULL"有两种情况：1、第一次输入，链表还没有建立；2、在寻找该结点的合适位置时遍历鲐链表的最后一个结点，即需要插入的结点应该在链表的末尾。

有了以上的理解，不难确定，以下 if-else 体中"index>v->index"的含义就是要在当前结点之前插入该结点。而"v->coef +=coef"则是完成在指数相等时的系数相加。仔细研究 if-else 体中的 if 部分，不难发现，该部分完成一个结点的插入，或者是在第一个结点输入后的处理。

先考虑第一次输入的情况，"p->next = v;"使 p->next 置空。填空（4）显然是处理第一次输入的情况(也包括当前输入结点指数大于链表中任何结点指数,该结点需要插入到链首的情况，读者可自己思考)。我们不管填空（5）的内容，当再次执行到"v=h;"和其后的循环体时，不难发现在（4）中要将当前链表的首指针赋给 h，否则这个 while 循环就不能完成任务。那么（4）的内容显然是"h=p"。权衡上述各种情况，可以确定即使在当前输入结点指数大于链表中任何结点指数的情况下，程序仍能够完成任务。

基于上述的分析，不难确定填空（5）应完成在链表中或链表尾插入结点的功能。已经有了"p->next =v;"，这里只要填写"u->next=p"就可以完成任务了。

现在回到函数 polymul。首先，程序取两个多项式中最大指数的和为乘积的最大指数，然后调用函数 reverse()将链表 g 反转，使 g 的各个结点指数从链首到链尾由小到大进行排列。关于函数 reverse()的理解，这里就不多说了。

进入 for 循环之后的两个 while 循环让人摸不着头脑，但如果将 i=maxindex 和 gp->index<i-fp->index 以及 fp->index+gp->index>=i 联系起来则不难看出,这是在两个多项式 gp 和 fp 中寻找两个结点，这两个结点相乘后的指数（index）先于或大于当前正在处理的指数（当然不会有大于 maxindex 的，但在程序处理过程中两结点指数和大于当前正在处理指数的机会会有很多)。

显然，在找到一对指数和先于当前正在处理指数的时候，这是通过指数的相加和系数的相乘

来生成一个由 q 来表示的新结点。这个结点应该插入链表中去，这就是填空（2）和（3）所在的 if-else 体应该完成的任务了。
这里的插入操作由一个 if-else 体和语句"tail=q"来完成。显然，插入时 p 有空和非空两种可能的状态，如果在（2）填写"p!=NULL"，那么"p=q"的结果是非常危险的，因为这将丢掉以前的运算结果，使我们最后只能悼念到运算结果的最后一项。因此这里只能填写"p==NULL"。
由此就可以确定（3）处理 p!=NULL 时的插入操作。由"tail=1"来看，tail 永远指向链尾的结点，那么这里应该先将 q 接到 tail 后，于是我们确定在（3）中填写"tial->next=q"。
现在只剩下填空（1）没有解决了。只有在两种情况下才能进入由"while(fp&&gp)"引导的循环：1、"gp->index+fp->index>i"；2、"gp->index+fp->index==i"。这里应该处理这两种情况，但为什么还要用一个 if-else 体呢？如果考虑 gp->index+fp->index==i 在进的处理我们可以看到 ，fp 和 gp 各自抽后移动了一个结点。fp 是按结点指数从大到小连接的，而 gp 是从小到大连接的，在处理了一次"gp->index+fp->index==i"的情况之后，很容易出现 gp->index+fp->index>i 或 gp->index+fp->index<i 的情况。当 gp->index+fp->index>i 时，为使其趋向 gp->index+fp->index == i，应该使 fp 向下移动一个结点反之使 gp 向下移动一个结点（为什么这样，请读者自己思考）。有了以上的理解，填空（1）的解答就显得很容易了，即"gp->index+fp->index>i"。
[答案]
（1）fp->index+gp->index>i
（2）p == NULL
（3）tail->next=q;
（4）h=p
（5）u->next =p

试题 4（1999 年试题 6）

阅读下列程序说明和 C 代码，将应填入 （n） 处的字句写在答卷的对应栏内。
[程序说明]
本程序从 n 种不同重量、不同价值的物品中选取一部分物品。要求在不超过限定重量 limw 的前提下，使被选取的那些物品的总价值较大。这里约定 limw 不超过 n 种物品的重量总和，也没有一种物品的重量超过 limw，并且各物品的价值都大于 0。
程序中，n 种物品被顺序编号为 0、1、2、.....、n-1。
【程序】

```
#include <stdio.h>
#define N 100
double limw;
int opts[N]; /* 存储临时最佳的选择方案，当 opts[i]为 1，物品 i 在解中 */
struct elem { double weight;
double value;
} a[N]; /* 物品的重量和价值信息 */
int k, n ;
struct { int flg; /* 物品的考虑状态：0：不选，1：将被考虑，2：曾被选中 */
double tw; /* 已达到的总重量 */
double tv; /* 期望的总价值 */
} twv[N]; /* 当前候选解中各物品的考虑状态，以及候选解的状态 */
```

```
main()
{ double maxv, find();
printf("Enter number of matter. "); scanf("%d", &n);
printf("Enter limit of weight. "); scanf("%1f", &limw);
printf("Enter weight and values of matters. ");
for (k = 0; k < n; k++) scanf("%1f%1f", &a[k].weight, &a[k].value);
maxv = find(a,n);
for(k = 0; k < n; k++) if(opts[k]) printf("%4d", k);
printf("\nTotal value = %1f\n", maxv);
}
next(int i , double tw, double tv) /* 将考虑 i 号物品 */
{ twv[i].flg = 1; twv[i].tw = tw; twv[i].tv = tv; }
look(int i, int *f, double *tw, double *tv) / * 取 i 号物品在解中的状态信息 * /
{ *f = twv[i].flg; *tw = twv[i].tw; *tv = twv[i].tv; }
double find (struct elem *a, int n )
{ int i, k, f;
double maxv, tw, tv, totv = 0.0;
maxv = 0;
for(k=0; k < n; k++) ____(1)____;
next(0, 0.0, totv);
i = 0;
while(i >= 0) {
look(i, &f, &tw, &tv);
switch (f) {
case 1: twv[i].flg++; /* 先考虑被选中 */
if(____(2)____ <= limw) /* 选中可行吗？ */
if(i < n-1) { /* 后面还有物品吗？ */
next(____(3)____); /* 置 i+1 物品的状态 */
i++;
}
else if (tv > maxv) { /* 是一个更好的候选解吗？ */
maxv = tv;
for(k = 0; k < n; k++)
opts[k] = twv[k].flg != 0;
}
break;
case 0: i--; break; /* 回退 */
default: /* f == 2 */
twv[i].flg = 0;
if (____(4)____) /* 不选 i 号物品可行吗？ */
if(i < n-1) { /* 后面还有物品吗？ */
next(____(5)____);
i++;
}
```

```
else {
maxv = tv - a[i].value;
for(k = 0; k < n; k++)
opts[k] = twv[k].flg != 0;
i--;
}
break;
}
}
return maxv;
}
```

[解析]

仔细阅读程序说明和程序，不能发现试题中需要解答的问题全部集中在函数 find( )中。在理解了函数 next( )和函数 look( )之后，我们开始阅读函数 find( )。

首先，由于后面有关于 next( )调用时参数表的填空，应该研究函数 next( )的 3 个参数的含义。阅读的 next( )函数体，能够确认：

参数 i 代表物品编号；

参数 tw 代表当前已经达到的总重量；

参数 tv 代表期望的总价值。

填空（1）之前有"totv=0.0"，之后 totv 作为函数 next( )的一个参数使用。通过研究函数 next( )，发现 totv 的值传给了 tw[i].tv，其含义应该是期望的总国。，显然 totv 不能是 0.0，那么（1）应该是对 totv 的值操作。究竟应该怎样赋值呢？根据说明，我们能够理解期望的价值应该人总价值开始试探，那么作为 for 循环的循环体，（1）应该主用来计算总价值的。所以我们在这里填写"totv+=a[k].value"。

现在看下面的程序。填空（2）所在的语句是通过 if 语句来判断选中某个物品是否可行，这里是考虑选中 a[i]是否可行。不可行的条件只有一个，即 a[i]的 weight 与当前已达到的总之和超过了重量限制 limw。通过对函数 look( )的解读，可以知道程序中 tw 即表示当前已经达到的总重量，那么结合后面的"v<=limw"，这里应该填写"tw+a[i].weight"。继续往下阅读程序，包含填空（3）的语句及注释和此前的语句表明这里对函数 next( )的调用是在第 i 个物品已经被选中后，考虑第 i+1 个物品。

由于是第 i+1 个物品，因此这里的第 1 个参数就是 i＋1。第 2 个参数，因为第 i 个物品已经被选中，所以已经达到的总重量就应该是"tw+a[i].weight"；第 3 个参数是期望的总价值，在经过函数 next( )和 look( )的操作之后 ，tv 的值即为期望达到的总价值。在没有超过 limw 的情况下还可以继续选择物品，这表明该期望值是合理的，所以应该维持原来的数值，即 tv。所以（3）的正确答案为"i+1, tw+a[i].weight,tv"。

由"if(tv>maxv)"引导的 if 条件句是处理这样一种情况，即选中当前物品会使总重量超过 limw，从而在不选择物品 i 的情况下得到一个解。由此不难确定，maxv 记录了程序处理过程中达到的最大价值的数值。有必要说明一下，因为得到的解的总价值是各不相同的，所以最大的值就被记录下来，这里被选中物品的编号也就被记录在数组 ops[ ]中作为最佳方案。

现在看 CASE 结构的 default 分支。在这里，填空（4）所在的 if 语句判断不选物品 i 是否可行。若选中物品 i 也不能够使总价值达到期望总价值，就可以不选物品，那么（4）的解答可以这样构造，即"tv-a[i].value>maxv"。同时，在这种情况下应该修改期望的总价值，上述情况表明当前期望的价值是不合理的。

填空（5）要求解答的是在不选物品 i 的情况下考虑下一个(i+1)物品。参照填空（3）的解答，

第 1 个参数就是 i＋1。那么第 2 个参数呢，这里决定不选物品 i 了，已经达到的总重量中就不能将物品 i 的重量包括进来，仍旧是 tw。第 3 个参数比较麻烦，物品 i 未选中，而且即使加上 i 的价值，当前最大价值仍旧不能够达到期望价值，就只好考虑修改期望价值，合理的修改方式是 tv-a[i].value。

另外，由于函数 next( )和 look( )使变量 tw 和 twv[i].tw、tv 和 twv[i].tv 具有等值关系，那么在解题时凡是使用的，都可以替换为 twv[i].twltwv[i].tv。

[答案]

（1） totv+=a[k].value

（2） tw+a[i].weight

（3） i+1,tw+a[i].weight,tv

（4） tv-a[i].value>maxv

（5） i+1,tw,tv-a[i].value

(注：解答中的 tw 可答为 twv[i].tw，tv 可答为 twv[i].tv).

试题 5 （1998 年试题 5）

阅读以下程序说明和 C 程序，将应填入(n)处的字句，写在答卷的对应栏内。

【程序说明】

这里给出的程序逐一从指走课程成绩文件中读入学生的学号和成绩，对同一学生汇总他的总成绩，并按以下格式输出名次（按总成绩由高到低的顺序）、总成绩、同一名次的学生人数、同一名次学生的学号（按学号由小到大的顺序）。

程序约定学生学习课程不超过 30 种，课程成绩文件的第一个数字就是课程号，统计过程中，同一课程号的成绩文件不能重复输入。

程序采用链表结构存储学生有关信息，链表中的每个表元对应一位学生。程序在数据输入过程中，形成一个按学生学号从小到大顺序链接的有序链表。当数据输入结束后，程序按总成绩从高到低，学号从小到大的顺序对链表排序。程序最后按指定格式输出链表中的信息。程序的输出格式如下例所示：

名　次	总成绩	人　数	学　号
1	470	2	1225
3	450	3	1524
6	430	1	14
7	401	3	1318

【程序】

```
＃INCLUDE〈STDIO.H〉
＃DEFINEM30
＃DEFINENLEN10
TYPEDEFSTRUCTNODE{INTCUR_S／*最近输入成绩的科目*/
                  CHARNO[NLEN]；INTSCORE；
                  STUCTIONNODE*NEXT；
                    ｝ NODE；
N0DE*BUBBLESORT(N0DE*HEAD)
{NODE*Q，*TAIL，*P＝(MODE*)MALLOC(SIZEOF(NODE)；
  P→NEXT＝HEAD；HEAD＝P；TAIL＝NULL；
  WHILE(TAIL!=___(1)___)
  {P=HEAD;Q=P→NEXT;
```

```
    WHILE(Q→ENXT！=TAIL)
    {IF(P→NEXT→SCORE→〈Q→NEXT→SCORE||
        P→NEXT→SCOR＝＝Q→NEXT→SCORE＆＆
        STRCMP(P→NEXT→NO，Q→NEXT→NO)〉0)
      {P→NEXT＝_(2)_，/*两相邻表元链接关系前后颠倒*/
        _(3)_＝Q→NEXT→NEXT；P→NEXT→NEXT＝Q；
      }
      P＝P→NEXT；/*调整 P 和 Q*/Q＝_(4)_；
    }
    TAIL＝Q；
  }
  P＝HEAD→NEXT；FREE(HEAD)；RETURNP；
}
INTS[M]，SP；
MAIN()
{FILE*FP；
  NODE*H，*U，*V*P；
  INTSS,，MARK，ORDER，C；
  CHARFNAME[80]，NO[NLEN]，ANS；
  FOR(H＝NULL，SP＝0；；)
  {PRINT("输入科目成绩文件名(输入 AAAA 表示强行结束)。\N")；
    WHILE(1)
    {SCANF("%S",FNAME)；
      IF(STRCMP(FNAME，"AAAA")＝＝0)BREAK；
      IF((FP＝FOPEN(FNAME，"R"))＝＝NULL)
        PRINTF("不能打开文件%S，请重新输入科目文件名。\N，FNAME)；
      ELSEBREAK；
    }
    IF(STRCMP(FNAME，"AAAA")＝＝0)BREAK；
    FSCANF(FP，"%D"，＆SS)；/&MIDDOT;输入科目号&MIDDOT;/S[SP]＝SS；
    FOR(I＝0；S[I]！＝SS；I＋＋)；
    IF(I<SP)
    {PRINTF("该科目的成绩已输入，请输入别的科目成绩文件。\N")；
      CONTINUE;
    }
    SP＋＋；
    WHILE(FSCANF(FP,"%S%D"，NO，＆MARK)＝＝2)
    {/*在链表中寻找最近输入的学号*/
      FOR(V＝H;V!＝NULL,＆＆STRCMP(V→NO,NO)＜0;U＝V,V＝V→NEXT)；
      IF(V！＝NULL＆＆STRCMP(V→NO，NO)＝＝0)
      {IF(V→CUR_S！＝＝SS)
        {V→SCORE＋＝MARK；V→CUR_S＝SS；
        }/*同一科目成绩的重复输入，后输入成绩被忽略*/
      }ELSE{P＝(NODE*)MALLOC(SIZEOF(NODE)；/*一位新的学生*/
```

```
            STRCPY(P→NO，NO)；P→SCORE＝MARK；P→CUR_S＝SS；
            P→NEXT＝V；
            IF(V＝＝H)H＝P；ELSEU→NEXT＝P；
    }FCLOSE(FP)；
    PRINTF("还有科目成绩文件要输入吗？(Y/N)")；SCANF("%C",&ANS)；
    IF(ANNS＝＝'N'||ANS＝＝'N')BREAK；
  }
  H＝BUBBLESORT(H)；
  PRINTF("名次总成绩人数学号\N")；/*以下按格式要求输出*/
  V＝H；ORDER＝1；
  WHILE(V！＝NULL)
  {FOR(C＝1,U＝V→NEXT;U!NULL&&U→SCORE==V→SCORE;C++,U＝U→NEXT)；
    PRINTF("%4D%7D%8D"，ORDER，V→SCORE，C)；
    FOR(ORDER＋＝C，I＝1；_(5)_；V＝V→NEXT，I＋＋)
    {IF(I＞1＆＆I%5＝＝1)PRINTF("\N%23C"，")；
      PRINTF("%S"，V→NO)；
    }PRINTF("\N")；
  }
}
```

[解析]

本题考查的重点是数据结构和排序算法。虽然数据结构中的链表和排序算法中的冒泡排序属于比较容易掌握的内容，但把两部分结合起来运用应有一定的难度。这道题目给我们启示就是在复习中不要把各个知识点孤立起来，要达到融会贯通的程度，这样在考试中就不会发生费很多力气接受程序算法的现象。我们看到，本题的题干部分给出了很多注释，使我们能够比较轻松地理解程序。

不要为题干的篇幅所迷惑，如果借助注释阅读程序，就会发现一直到语句"h=bubblesort(h)"都出现需要填写的空白。本题需要填写的空白集中在链表排序和统计结果的打印上。

遵循程序的模块划分，首先进行填空（5）的填写。结合对以上程序的理解，易知需要打印的对象是一个有序链表，先按成绩排序，在成绩相同的情况下按学号排序。打印的格式要求已经给出，关键是如何实现按格式要求输出。语句"while(v!=NULL)"引导程序进入一个大循环，由循环条件"v!=NULL"和该循环的结束即程序的结束可以断定所有的输出均在循环体内完成。

仔细推敲循环体，发现循环体是由 3 个部分构成的。第 1 部分是一个 for 循环，适用于计算取得相同成绩的学生的个数，把计算结果存放在整型变量中，对这个循环的理解，关键是还要忽视了语句最后有一个"；"，如果忽视了这个"；"，对程序的理解就会出现根本性的错误。第 2 部分是一个打印语句，包括条打印名次(order)、总成绩(v-score)和取得该成绩的学生人数(c)。第 3 部分也是一个 for 循环，打印取得该成绩的学生的学号，我们重点研究这一部分。从对第 1 部分的理解我们可以知道，c 为取得该成绩的人数，v 为取得该成绩的第一个学生结点，u 为取得该成绩的最后一个学生结点。理解了上述 3 个变量的含义，填空（5）循环条件的解答也就非常容易了，即尚未打印完毕取得该成绩的所有学生。表达方式是多种多样的，可以通过对取得该成绩的学生人数(c)递减来判断。这种判断的构造也是多种多样的，可以通过循环计数变量 i（在程序中主要用来控制打印学号的换行）和变量 c 的关系来构造，也可以通过判断结点 v 和结点 u 是否重合来构造，但最简单的特征条件应该是通过变量 i 和变量 c 的比较来构造。

现在回过头来看链表的排序。如果我们对"bubble"这个英语单词不陌生的，就应该马上意识到这是一种冒泡排序。函数中的第一句注释"两相邻表元链接关系前后颠倒"也显示出很强的冒泡色彩。函数的重心是一个二重循环，在这个二重循环中集中了我们需要解答的 4 个填空语句，所以对该循环的理解就是对本题的解答。

在几个变量初始化语句之后，进入循环体刚进入循环就遇到第 1 个填空，需要我们填写循环条件，显然在我们对该循环的循环体没有充分理解之前是难于完成这个任务的，我们暂且跳过该语句，因为该循环是洽谈室要继续下去的，否则本题所提供的程序是无法工作的。循环体的前两条语句使我们得到这样的信息，结点 p 在前，结点 q 在后，每次执行这两条语句问题使结点 p 和结点 q 处于链表的前两个结点上。

接下来是内循环。内循环的一开始是一个 if 语句，该语句的判断条件很长，但仔细阅读，发现这是排序的条件———成绩高靠前，相同成绩时学号靠前。借助注释，我们能够与日俱增当相邻表元链接关系前后颠倒的条件成立时，需要颠倒的两个表元是 p->next 和 q->next 而不是 q 和 p。所以 p->next 就应越过 q，指向 q->next，q->next 应该指向 q->next->next。在 p->next->next 指向 q 之后，原来的 q 已经变为 p->next->next，原来的 q->next 已经变为 p->next。在一次比较（交换）之后，应该调整 q 和 p，在"p=p->next"语句之后，是对 q 的调整，显然，根据我们以上的理解，q 应该是 p 的后续结点，所以填写"q=p->next"是非常轻松的事情。

内循环结束之后，我们看到调整语句是"tail=q"，而每次比较（或交换）时，head 和 tail 是不参加的，这就意味着每次循环使比较（或交换）的对象减少一个。当若干次循环之后，和成为相邻表元，中间再也没有未经排序的表元之后，表示排序工作已经结束，循环也就应该结束，所以填空（1）的循环条件也就非常容易构造，即"tail!=head->next"。

[答案]

（1）head->next

（2）q->next

（3）q->next

（4）p->next 答 q->next 给 1 分

（5）c-- !=0 或 c-- 或 u!=v 或 i<=c 或 c-- >0

试题 6（1998 年试题 7）

阅读以下程序说明和 C 程序，将应填入(n)处的字句，写在答卷的对应栏内。

【程序说明】

本程序的函数 SUM(INT，IINTTOTAL，INTSIGMA，INTREAR，INTD[]，INTN)用来从已知数组 D 的前 N 个元素中找出所有部分元素序列之和等于 TOTAL 的元素序列，约定数组 D 的元素都是正整数，且都小于等于 TOTAL。

函数 SUM 使用递归方法找出全部解答。参数 I 表示递归函数当前考虑元素 D[I]，参数 SIGMA 是调用前已选取的部分序列的元素和，参数 REAR 是后面还未考虑的那部分元素的元素和。函数对元素 D[I]有两种可能的选择方案：

1．考虑元素 D[I]被包含在新的部分元素序列中的可能性。如果在当前部分元素序列之后接上 D[I]，新序列的元素和不超过 TOTAL，则函数将 D[I]包含在当前部分元素序列中。如果新的部分元素序列的元素和等于 TOTAL 时，新的部分元素序列就是一个解答，函数将其输出；否则，若继续考虑后面的元素还有可能找到解答时，函数就递归去考虑后面的元素，寻找解答。最后，函烽就恢复原来部分元素序列中不包含 D[I]的状态。

2．考虑元素 D[I]不被包含在新的部分元素序列中的可能性。如果继续向 D[I]之后考虑还是有希望能得到和为 TOTAL 的部分元素序列时，函数将新序列不包含 D[I]也作为一种可能的选择，并递归去考虑后面的元素，寻找解答。

【程序】

```
＃INCLUDE〈STDIO.H〉
＃DEFINE N100
INT A[N]；
INT FLG[N]；
SUM(INTI，INTTOTAL，INTSIGMA，INTREAR，INTD[]，INTT)
{INTJ；
  /*考虑元素 D[I]被包含在新的部分元素序列中的可能性*/
  IF(SIGMA＋D[I]TOTAL/*如果 D[I]与当前序列的和不超过 TOTAL*/
    {FLG[I]＝1；/*D[I]被考虑在被部分元素序列中*/
     IF(_(1)_＝＝TOTAL)
     {/*输出解*/
       FOR(J＝0；FLG[J]＝＝0；J＋＋)；
       PRINTF("%4D＝%D"，TOTAL，D[J])；
       FOR(J＋＋；＜＝I；J＋＋)
            IF(FLG[J])
                 PRINTF("＋%D",D[J])；
       PRINTF("\")；
     }
     ELSE/*并且继续考虑后面的元素有可能找到解答时*/
     IF(I＜N－1&&REAR－D[I]＋SIGMA＞＝TOTAL)
          SUM(I＋1，TOTAL，_(2)_，REAR－D[I]，D，N);
     _(3)_；
  /*考虑元素 D[I]不被包含在新的部分元素序列中的可能性。*/
  IF(I＜N－1&&REAR－D[I]＋TIGMA＞＝TOTAL)
            SUM(I＋1，TOTAL，_(4)，REAR－D[I]，D，N)；
}
MAIN()
{INTI，J，N，TOTAL，S，D；
  PRINTF("输入 TOTAL！/N")；SCANF("%D",&TOTAL)；
  PRINTF("输入 N！/N")；SCANF("%D",&N)；
  FOR(S＝I＝0；I＜N；)
  {PRINTF("输入第%D 个元素＞0 且＜＝%D)\N"，I＋1，TOTAL)；
    SCANF("%D"，&D)；
    IF(D＜1||D＞TOTAL)
      {PRINTF("出错，请重新输入！\N")；
           CONTNUE；
      }
      S＋＝A[I＋＋]＝D；
  }
  SUM(0，TOTAL，0,_(5)_，A，N)；
  PRINTF("\N\N")；
}
```

[解析]

如果不仔细阅读程序说明，可能会话中有话话中有话误以为本题考查的重点是探测法，但在切实理解了程序说明之后，我们能够认识到本题的考查重点是函数的递归调用。我们仍旧强调要结合程序说明来解答题目，因为题干所提供的程序是对程序说明的踏实实现，或者说题干所提供的程序就是程序说明的另一种表达方式。

在仔细阅读程序说明之后，从 main( )开始读起。在读完程序的 for 循环之后，可以得到如下信息；整型数组 a 中有个 n 元素，是选取符合要求的元素序列的范围；s 为数组 a 中所有元素的和；total 为我们选取的特定元素序列的和，凡是和为 total 的元素序列都将被视为符合条件的元素序列加以输出。for 循环之后是对函数的调用，其中第 4 个实际参数是要解答的问题。参照程序说明，发现该参数为后面还未考虑的那部分元素的元素和。在这里，函数是第一次调用，函数当前考虑的元素是 a[0]，调用前已选服部分序列的和为 0，则该填空处应该填写的内容就是数组 a 所有元素的和。结合对 for 循环的理解，正确答案应该是。此处易犯的错误是填写"s-a[0]"。一定要记住，该参数是尚未包含进序列后面的那些元素的和。这一点也可以从填空（2）和填空（4）所在的两次函数调用得到佐证，因为我们看到只有在确定了 d[i]是否包含进序列中之后才在 rear 中减去 d[i]。很明显，此处尚未将 a[0]包含进序列，故只能填写"s"。

下面我们对本题的主体部分——函数 sum( )进行解析，解答包含在函数体内的 4 个填空。函数体的主要内容是两个 if 语句，仔细阅读选择数组元素的方案，发现第一个 if 语句对应第 1 种选择方案，第 2 个 if 语句对应第 2 种选择方案。现在就可以把整个函数体分成两个互不相干的两部分，分别加以考查。

第 1 个 if 语句的内容饱含个部分，即 1、将 d[i]包含进部分元素序列中；2、if-else-if 语句，该语句实现了选择方案 1；3、需要解答的填空，该语句在 if-else-if 语句条件不成立时执行。很明显，填空（1）处判断某个值是否等于 total，结合下面的注释和对程序的理解，能够很容易持理解在该判断成立时输出一个符合条件的元素序列，因此就不难确定这里判断考虑进 d[i]之后元素序列的和是否等于 total。未考虑进 d[i]之前的元素序列的和是 sigma，那么考虑进 d[i]之后的元素序列之和就应该是 sigma+d[i]了，我们应该毫不犹豫地将"sigma+d[i]"填写到答题纸上。

填空（2）是函数调用　的第 3 个实际参数，结合注释和程序说明，很容易就能推断出该函数调用实现了在 d[i]被包含进元素序列之后而元素和尚小于 total 的处理。根据心目对程序的理解，易知第 3 个实际参数应该是包含进 d[i]之后的元素序列和，未包含 d[i]之前的序列的和是，那么包含进 d[i]之后的元素序列之和就应该是 sigma+d[i]，因此此处应该填定与填空（1）相同的内容。

如果你觉得想像不出填空（3）应该填写什么答案，那么就再回过头去读程序说明中的选择方案。这里有一句话就是对填空（3）的表述——"函数应恢复名不包含的状态"。把 d[i]包含进序列的操作是把数组 flg 的第 i 个元素赋值为 1，而且在上面打印符合要求的元素序列时跳过在 flg 中对应值为 0 的元素。易知某个元素不包含在序列中时，在 flg 中对应的元素值应该为 0，即恢复序列中不包含 d[i]的状态，就应该是 flg[i]的重新置 0 了，所以此处应该填写"flg[i]=0"。

函数 sum( )主体的第 2 个 if 语句要实现的功能是，当元素 d[i]不被包含在序列中时，继续求解。既然元素序列中没有包含 d[i]，则元素序列的和就没有发生变化，仍旧是上次调用时的值，即 sigma。至此我们已经解答了全部问题。

回过头来看看本题的解答，我们应该坚定这样的钥匙思路，把程序划分成相对独立的几部分，给各部分在程序说明中找到对应的描述。但难度也就在这里。如果程序划分或与程序说明对应时发生失误，可能会导致我们的解答全车覆没，只要大家在平时的练习中注意这方面的锻炼，发生错误的可能性还是比较小的。

[答案]
（1）sigma+d[i]
（2）sigma+d[i]
（3）flg[i]=0
（4）sigma
（5）s

试题 7（1997 年试题 6）

阅读以下程序说明和 C 程序，将应填入_(n)_处的字句，写在答卷的对应栏内。
【程序说明】
某系统由 N 个部件组成，这些部件被物理地分成若干个分离的部件组。同一组内的两件部件 I 和 J，它们或直接相连，或间接相连（部件 I 和部件 J 间接相连是指在这两件部件之间有一个部件相连序列，其中部件 I 和 J 分别与这相连序列中的某个部件直接相连）。系统的 N 个部件被统一编号为 0，1，…，N-1。本程序输入所有直接相连的部件号对，分别求出系统各分离部件组中的部件号并输出。
程序根据输入的直接相连的两件部件号，建立 N 个链表，其中第 I 个链表的首指针为 S[I]，其结点是与部件 I 直接相连的所有部件号。
程序依次处理各链表。在处理 S[I]链表中，用 TOP 工作链表重新构造 S[I]链表，使 S[I]链表对应系统中的一个部件组，其中结点按部件号从小到大连接。
【程序】

```
#INCLUDE
#DEFINE N 100
TYPEEFSTRUCTNODE{
INTDATA;
STRUCTNODE*LINK;
}NODE;
NODE *S[N];
INT I,J,N,T;
NODE *Q,*P,*X,*Y,*TOP;
MAIN()
{PRINTF("ENTERNUMBEROFPARTS.");
SCANF("%D",&N);
FOR(I=0;IDATA=J;P->LINK=S[I];S[I]=P;
P=(NODE*)MALLOC(SIZEOF(NODE));
P->DATA=I;P->LINK=S[J];S[J];=P;
}
FOR(I=0;IDATA]!=NULL)
{/将 J 链表也移入工作链表*/
FOR(P=S[J];P->LINK！=NULL;P=P->LINK);
P->LINK=TOP;TOP=S[J];__(3)__;
}
/*在重新生成的第 I 链表中寻找当前结点的插入点*/
FOR(Y=S[I];__(4)__;X=Y,Y=Y->LINK);
IF(Y!=NULL&&Y->DATA==Q->DATA)
```

```
FREE(Q);/*因重新生成的第 I 链表已有当前结点，当前结点删除*/
ELSE{/*当前结点插入重新生成的 I 链表*/
__(5)__；
IF(Y==S[I])S[I]=Q;
ELSEX->LINK=Q;
}
}
FOR(I=0;IDATA);
Q=P->LINK;FREE(P);P=Q;
}
PRINTF("\N");
}
}
```

[解析]

本题考查的知识要点是链表操作。链表的基本操作是简单的，仔细阅读试题说明，结合程序中的说明文字，不难发现本题的难度主要在最后一段中，"在处理链表 s[i] 表中，用 top 工作链表重新构造 s[i] 表，使 s[i] 链表对应系统中的一个部件组，其中结点按部件号从小到大连结。"

阅读试题所提供的程序，我们能够从输入部分的构造链表算法中推知，这里建立的是循环链表，即链表的尾结点的指针指向链表头。结合程序说明，5 个需要回答的问题集中在顺序处理各个链表部分，所以这里的关键是要结合程序理解程序中使用的算法。

仔细推敲程序，可以断定处理各链表的过程应该为：对由 s[i]所指向链表，将该表移到由指针 top 所指的工作链表。对 top 链表的各结点进行如下处理：从 top 链表上取出一个结点，根据该结点所指出的相连部件，将由 s[j]所指向的链表也移入 top 链中，并将取出的结点按部件递增顺序重新构造由 s[i] 所指向的链表，如此重复，直到 top 链表为 NULL。这样由 s[i]所指向链表的重新构造就结束了。所有链表处理完后，重新构造好的各非空链表，与系统中的各部件组一一对应。

仔细阅读填空（1）所在的语句，还要忽视填空（1）前是"，"，而不是"；"，可以推知，这里应该是将 s[i] 置空，因为该循环的任务是重新构造链表。结合心目输入部分的链表构造及以后对由 top 所指向的链表的使用，不难推知由 top 所指向的链表也是循环链表，结合上下文，填空（2）的答案应该是" top=top->link "，实现链表尾结点的指针指向链表头的操作。考查填空（3）及其前面的两条语句，它们所承担的任务与" top=s[i] "和填空（1）所承担的任务应该是相同的，由彼及此，得出（3）的应该是很容易的。

填空（4）所在的语句组完成在重新链表中寻找当前结点的插入点的任务。根据试题说明，链表是按部件号递增顺序构成的，结合基本的链表遍历知识，（4）的答案应该是"y!=NULL && y->data<q->data "。根据填空所在语句及程序说明，结合链表插入操作的基本流程和程序对链表构造的要求，答案应该是"q->link=y"。

[答案]

（1）S[i] = NULL

（2）Top = top->link

（3）S[j] = NULL

（4）y !=NULL && y->data <q->data 或 y && y->data <q->data

（5）q->link = y

试题 8（1997 年试题 8）

阅读以下程序说明和 C 程序，将应填入_(N)_处的字句，写在答卷的对应栏内。

【程序说明】

一个相连的区域被不规则地分割成 N 个不同的小区域；每个小区域与若干其它小区域相邻接。现用 CN 种不同颜色为该区域着色，要求每个小区域着同一种颜色，相邻小区域着不同颜色。

设小区域被顺序编号为 0，1，…，N-1。每个小区域与其它小区域的邻接关系用两维数组 BORDERING 表示，元素 BORDERING[I][J]表示 I 号小区域与 J 号小区域之间的邻接关系：

BORDERING[I][J]=0 J 小区域与 I 小区域不邻接

BORDERING[I][J]=1 J 小区域与 I 小区域相邻接

程序中，把计算结果存入于两维数组 COLORED 中，颜色编号为 0,1,…,CN-1,元素 COLORED[COLER][J]的含义是

COLORED[COLOR][J]=0 J 小区域不用颜色 COLOR 着色

COLORED[COLOR][J]=1 J 小区域用颜色 COLOR 着色

函数 COLORCOUNTRY(BORDERING,COLORED,N,CN)根据所给的小区域邻接关系数组 BORDERING、小区域个数 N、颜色数 CN，将找到的着色方案记录在数组 COLORED 中。函数采用试探法找解。首先从第一个小区域着第一种颜色开始顺序为各小区域找着色方案。对某个小区域，当为它找到一种未被它的相邻小区域着色的颜色时，就用该颜色对该小区域着色，并准备处理下一个小区域。当不能为某个小区域找到一个未被它的相邻小区域着色的颜色时，就回溯。如最终为全部小区域找到着色方案，函数返回 1；否则，函数返回 0。

程序假定小区域个数不超过 20，颜色数为 4。

【程序】

```
#INCLUDE
#DEFINE N 20
#DEFINE CN 4
INT
COLORCOUNTRY(INTBORDERING[][N],INTCOLORED[][N],INTN,INTCN){INTCOLOR,USED,I,C;
FOR(COLOR=0;COLORN;I++)COLORED[COLOR][I]=0;
C=0;/*从第 1 个小区域开始*/
COLOR=0;/*从着第 1 种颜色开始试控*/
WHILE(CC;I++)
IF(__(2)__)USED=1;
IF(!USED)BREAK;/*当前颜色未被相邻小区域着色*/
COLOR++
}
IF(!USED)
{/*找到一种可用颜色，用此色着色，并准备处理下一个小区域*/
__(3)__=1;COLOR=0;
}ELSE{/*未找到一种可用颜色，回溯*/
C--;IF(C<0)RETURN0;/*　发　现　没　有　解　的　情　况*/FOR(COLOR=0;__(4)__;COLOR++);__(5)__=0}}RETURN1;}PRINT(INTCOLORED[][N],INTN,INTCN)/*　输　出　结　果*/{CHAR*COLORT[]={"RED","BLUE","GREEN","YELLOW"};INTCOLOR,I;FOR(COLOR=0;COLORN;I++)
```

```
IF(COLORED[COLOR][I])PRINTF("\T%D",I);
PRINTF("\N");
}
}
INTCOLORED[CN][N],BORDERING[N][N];
MAIN()
{INTC,I,J,N;
PRINTF("ENTERNUMBEROFAREAS.");SCANF("%D",&N);
PRINTF("ENTERBORDERING:\N");
FOR(I=0;IN;J++)BORDERING[I][J]=0;
FOR(I=0;I0TONEXT).\N",I};
SCANF("&D",&J);
WHILE(J>=0)
{IF(I!=J)BORDERING[I][J]=BORDERING[J][I]=1;
SCANF("%D",&J);
}
}
IF(COLORCOUNTRY(BORDERING,COLORED,N,CN))
PRINT(COLORED,N,CN);
ELSEPRINTF("NOSOLUTION.\N");
}
```

[解析]

该题考查的仍然是回溯法的问题。试题说明非常详细，对算法的注释也非常详尽。关键是我们是否能够将试题说明和程序结合起来读，将试题说明中的每一部分与程序中的语句段建立联系。程序中有许多注释，这使试题说明与程序的联系非常明显。

需要解答的问题全部集中在函 colorcountry( ) 数中，我们就集中精力解读这个函数。首先结合程序注释明确各个变量的含义，这是对以后程序理解的基础。然后从宏观上把握函数　结构，不难发现，函数体的主体部分是一个 while 循环。循环由一个 while 循环和 if-else 体构成。结合试题说明与程序注释，可以推知内层 while 循环对每一个小区进行着色，if-else 体将处理找到合适颜色和找到合适颜色的情况，把握了程序的结构，我们甚至不用对整修程序有透彻的理解，只要结合变量含义、试题说明与程序注释就可以解答问题。

首先，我们由填空（1）后面的注释可以推知这里应该是 color 与 cn 的比较，以控制循环的结束。既然是顺序对每种颜色作试探，则在每次执行到这里时，应该递加。因为循环体中已经实现了 color 的递加，所以（1）的解答就可以简单地构造为"color<cn"。填空（2）所在循环完成的任务是检查当前颜色是否已被某相邻小区域着色，结合循环中两个变化的变量 i、c 以及对试题说明的理解，当前颜色已被某相邻小区域着色，应该表示为 bordering[c][i]不为 0，colored[color][i] 不为 0，所以填空（2）的答案就可以表示为"bordering[c][i] && colored[color][i]"。

填空(3)、(4)、(5)集中在处理找到或没有找到的体中。找到了如何处理？首先要在 colored[]中登记，即 colored[colored][c]将置 1。如果单从这一个处理来看，这样的答案也算是正确的结合注释中"准备处理下一个小区域"的语句，通观全局，准备工作自然不能在处理没有找到合适颜色时进行，那么我们在这里实现对 c 的递加，即将答案改为 colored[colored][c++]，注意 c++与++c 的区别，体会在这里这样填写的理由。

显然，填空（4）和（5）完成没有找到合适颜色时的处理。结合心上着色的算法推算回溯的

算法，不难推出回溯的基本算法，即回溯过程中不但要回到前一步，还要能将当前已设置的信息恢复到它原告的状态，然后，以便可以根据前一步的状态继续试探其他方案。我们知道，数组 colored 中已经记录了将当前颜色要重新返回到上一个区域的颜色值。程序中（4）的解答为"colored[colored][c]"。回溯的过程除了恢复状态外，还要取消原有的选择，继续进行选择，得到下一个可能的值。因此填空（5）的内容应该为"colored[colored++][c]"。通过该语句，就未完成了回溯过程，可以继续篁，向前继续试探。

我们可以看到，本题的 5 个问题中只有（4）和（5）才涉及到回溯法的构造，其余的 3 个问题都是可以通过对试题说明和程序结构、功能的把握来解答的，这也从另一个方面说明了在钥解题时仔细阅读试题说明和从结构上把握程序的重要性。

[答案]

（1）color < cn 或 color < 4

（2）bordering[c][i] && colored[color][i] 或 bordering[c][i] == 1&&colored[color][i]==1 或 bordering[c][i] * colored[color][i] == 1 其中 bordering[c][i]可写成 bordering [i][c]

（3）colored[color][c++]

（4）colored[color][c]=0 或！colored[color][c] 或 colored[color] != 1

（5）colored[color++][c]

试题 9（1996 年试题 5）

阅读以下程序说明和 C 程序，将应填入程序中_(n)_处的字句，写在答卷的对应栏内。

【程序说明】

本程序是一个简单的计算器模拟程序。对任意给定的正确四则运算表达式，程序计算其结果值并输出。表达式中运算分量为无正负号整数，运算符为+、_、*、/，圆括号按常规配对，表达式以字符"="结束。

函数 getach()为获取表达式的一个合法自负，并将字符存入变量 curch；函数指针树组 func[]是为了统一加减乘除计算而设置的。

【程序】

```
#include <stdio.h>
int add(int x,int y){return x+y;}
int sub(int x,int y){return x-y;}
int mul(int x,int y){return x*y;}
int div(int x,int y){return x/y;}
int (*func[])()={add,sub,mul,div};
int num,curch;
char chtbl[]="+-*/()=";
char corch[]="+-*/()=0123456789";
int getach()
{ int i;
while(1)
{ curch=getchar();
if(curch==EOF) return -1;
for(i=0;corch[i]&curch!=corch[i];i++);
if(i<strlen(corch)) break;
}
return curch;
```

```
}
int geid()
{ int I;
if(curch>='0'&&curch<='9')
{ for(num=0;curch>='0'&&curch<='9';getach())
num=_①_;
return -1;
}
else{for(i=0;chtbl[i];i++)
if(chtbl[i]==curch) break;
if(i<=5) getach();
return i;
}
}
int cal()
{ int x1,x2,x3,op1,op2,i;
i=getid();
if(i==4) x1=cal(); else x1=num;
op1=getid();
if(op1>=5) return x1;
i=getid();
if(I==4) x2=cal(); else x2=num;
op2=getid();
while(_②_)
{ i=getid();
if(i==4) x3=cal(); else x3=num;
if((op1/2==0)&&(op2/2==1))
x2=(*func[op2](x2,x3));
else{ x1=_③_;
x2=x3;
_④_;
}
op2=getid();
}
return _⑤_(x1,x2);
}
void main()
{ int value;
printf("Please input an expression:\n");
getch();
while(curch!='=')
{ value=cal();
printf("The result is : %d\n",value);
printf("Please input an expression:\n");
```

```
getach();
}
}
```

[解析]

本程序是一个简单的计算器模拟程序，对值得任给的正确的四则运算表达式，程序计算其结果值并输出。程序由一个主程序和数个子程序组成。主程序的功能比较简单。输入一个表达式，计算其结果并输出，然后输入下一个表达式。4 个计算子程序 add、sub、mul、div 和输入字符子程序 getach 都很简单，这里不再浪费笔墨。

getid 子程序的功能是判断刚才输入的字符是操作数不是运算符。程序使用了一个 if 语句，若读入的字符是'0'到'9'之间的，则是操作数，那么继续将该操作数全部读进来（因为操作数也是以字符方式读进来的，因此一个操作数需要分为多个字符读进来，然后再装配成一个操作数），并将其值保存至全局变量 num 中。读入字符的顺序是先高位后低位，因此读入的数字字符要按照由高位到低位的顺序依次转化为一个数，所以填空（1）的答案为"num*10+curch-'0'"。若读入的字符不是数字，那么要判断一个是哪个运算符，不同的运算符返回不同的值。

cal 子程序是本题的关键所在。它比较复杂，使用了递归算法。它的功能是：首先判断已经输入的字符，若是'（'则根据括号优先的原则，先递归调用自己，求出括号里的表达式的值，否则是操作数（本程序无查错功能，假定所输入的都是正确的四则运算表达式，这样一个表达式的开头不是括号就肯定是操作数），将其值从全局变量 num 赋给 x1。接着读入一个运算符，若运算符是'）'或'＝'，则完成一级运算，返回结果 x1。读入第 2 个操作数，同上，若读入的字符是'（'，则根据括号优先的原则，先递归调用自己，求出括号里的表达式的值，否则将其值赋给操作数 x2。然后读入第 2 个运算符。下一步的操作应当是先读入第 3 个操作数，再根据 op1 和 op2 的优先级决定进行运算 op1 还是 op2，并将计算结果作为操作数 x1 或 x2，继续后计算。循环的结束条件是遇到'）'或'＝'，说明一对括号内的计算完成整个计算结束。因此，填空（2）的答案为"op2<5"。在循环内部，先读入第 2 个操作数，同上，若是'（'则先求括号里的表达式值。如果第一个运算符 op1 为'+'或'-'，第 2 个运算符 op2 为'*'或'/'，那么先进行 op2 并将计算结果记入 x2 中；否则先进行 op1 并将结果放入 x1 中，x3 放入 x2 中，op2 放入 op1 中，以进行下一步运算。因此，填空（3）的答案为"(*func[op1])(x1,x2)"，（4）的答案为"op1=op2"。整个循环结束后，只剩下 x1、x2 和 op1，那么最后计算一下 x1、op1 和 x2，作为整个表达式的结果，因此填空（5）的答案为"(*func[op1])(x1,x2)"。

[答案]

（1）num*10+curch-'0' 其中'0'可答成 48 或 0x30 或 corch[7]

（2）(op2>=0)&&(op2<5) 或 op2<5 或 op2<=4 或 !(op2>=5)

（3）(*func[op1](x1, x2)

（4）op1=op2

（5）(*func[op1])

试题 10（1996 年试题 7）

阅读下列程序说明和 C 程序，将应填入程序中(n)处的字句，写在答卷纸的对应栏内。

【程序说明】

本程序先从文件读入各考生的准考证号（设为整型数）及成绩，并将其存放在一棵检索二叉树上，二叉树的结点的键值是成绩，二叉树每个节点带一链表，链表结点存放取得该成绩的考生的准考证号。然后，程序按中序遍历检索二叉树，从高到底分输出结果，使每行输出某成绩及其取得该成绩的各考生的准考证号。

【程序】

```
#include <stdio.h>
typedef struct idnode{
int id;
struct idnode *next;
}IdNode;
typedef struct marknode{
int mark;
IdNode *head;
struct marknode left,right;
}MarkNode;
char fname[]="sp07.dat"'
main()
{ int id,mark;
MarkNode *root=null;
FILE *fp=fopen(fname,"r");
if(!fp){
printf("file %s open error.\n",fname);
exit(0);
}
while(!feof(fp)){
fscanf(fp,"%d%d",&id,&mark);
btree(&root,id,mark);
}
fclose(fp);
print(root);
}
btree()
{ IdNode *ip;
MarkNode *mp=mpptr;;
if(_①_){
if(mark==mp->mark) addIdNode(_②_,id);
else if(mark>mp->mark) btree(&mp->left,id,mark);
else btree(&mp->right,id,mark);
}else
{ mp=(MarkNode *)malloc(sizeof(MarkNode));
mp->mark=mark;
mp->left=mp->right=NULL;
_③_
addIdNode(&mp->head,id);
_④_
}
}
addIdNode(IdNode **ipp,int id)
```

```
{ IdNode *ip=ipp;
if(_⑤_) addIdNode(_⑥_,id);
else{
ip=(IdNode *)malloc(sizeof(IdNode));
sp->id=id;
ip->next=NULL;
_⑦_
}
}
print(MarkNode *mp)
{ IdNode *ip,ip0;
if(mp){
print(mp->left);
printf("%6d: \t",mp->mark);
ip=mp->head;
while(ip){
printf("%6d: \t",ip->id);
ip0=ip;
ip=ip->next;
free(ip0);
}
printf("\n");
print(mp->right);
free(mp);
}
```

[解析]

本程序着重考查对检索二叉树、链表的基本操作及双重指针的用法等知识点的掌握。

二叉树或者为空，或者由一个根结点加上左子树和右子树（互不相交的两棵二叉树）组成。检索二叉树左子树的每一个结点的值都小于（或大于根结点的值，右子树的每一个结点的值都大于（或小于）根结点的值，其左子树和右子树也是检索二叉树。双重指针是指向指针的指针，它的值是一个指针的地址。

大致浏览一个程序，程序主要由一个主程序和 3 个函数 btree、addidNode、print 组成。主程序的功能比较简单。打开一个文件，循环读出文件中的每一条记录，并将该记录的内容存放在一棵检索二叉树上，所有的记录读完以后，再将该检索二叉树打印出来。

程序最重要的地方是在函数 btree 中，btree 的功能是将一个新的标点插入检索二叉树中，它有 3 个参数，mpptr 是检索二叉树的根结点，id 和 mark 是从数据文件中读入的准考证号和成绩。

检索二叉树的插入算法是若二叉树的根结点为空，则创建一个根结点，给其赋上相应初值，否则判断根结点的键值与所给的值的大小。结果可能有 3 种：所给值大小根，则递归检索根的左子树，即在左子树上插入新的结点；所给值等于根结点值，结点已存在；所给值小于结点值，则递归检索根的右子树，即在右子树上插入新的结点。

根据检索二叉树的插入算法，分析 btree 的算法。题中 if(1) { }else { }的结构，对应于检索二叉树的插入算法中判断根结点是否为空的算法。因而填空（1）应为"mp!=NULL"。若根结点不空，则：若所给 mark 与根结点的 mark 值相等，就需要在根结点的 id 链表上增加一个

结点，调用函数 addidNode 实现这一功能，因此，填空（2）所在的语句为"addidNode(&mp->head,id)"；若所给 mark 大于根结点的 mark 值，则需要将该记录插入在根结点的左子树上，递归调用函数 btree 实现这一功能；若所给 mark 小于根结点的 mark 值，则需要将该记录插入在根结点的右子树，递归调用函数 btree 实现这一功能。若根结点为空，则创建一个根结点，给其赋上相应初值，赋给其 mark 值，初始化其左右子树为空，初始化其 id 链表为空，所以填空（3）答案为"mp->head=NULL"。此时，插入工作 id 尚未完成，该考生尚未保存，再调用函数 addldNode 将该考生的 id 插入到该结点的 id 链表中。最后，应将临时指针 mp 赋回给根结点，所以填空（4）的答案为"*mpptr=mp"。

函数 addidNode 与 btree 有些类似，该函数也是一个递归函数，它与 btree 的不同在于它是在一个线性链表上操作，而 btree 是在一个检索二叉树上操作。addIdNode 的功能是在一个链表尾部插入一个结点。它的算法也是先判断结点是否为空，若不空，则需递归在其 next 结点上插入，所以函数 addidNode 中填空（5）的答案为"ip!=NULL"，（6）的答案为"addIdNode(&p->next,id)"，即将结点插入到当前链表的下一结点所带链表的后面；若为空，则需要新建一结点，初始化其 next 为空，赋给其 id 值。若 ip 为空，并将临时指针 ip 赋回给原来的指针，所以填空（7）的答案为"*ipp=ip"。

程序最后根据已经建好的检索二叉树打印排序好的结果，方法是中序遍历树上每个结点，结每个结点依次访问结点 id 链表中的每一个结点，打印出其 id 值。

[答案]

（1）mp 或 mp !=NULL 或 *mpptr 或 *mpptr !=NULL\

（2）&mp->head 或 &(mp->head)

（3）mp->head=NULL

（4）*mpptr=mp

（5）ip 或 ip !=NULL 或 *ipp 或 *ipp !=NULL

（6）&ip->next 或 &(ip->next)

（7）*ipp=ip

试题 11（1995 年试题 7）

阅读下列程序说明和 C 程序，将应填入程序中 (n) 处的字句，写在答卷纸的对应栏内。

【程序说明】

本程序用回溯算法来产生由 0 或 1 组成的 2^m 个二进位串，使该串满足以下要求。

视串为首尾相连的环，则由 m 位二进制数字组成的 2^m 个子序列，每个可能的子序列都互不相同。例如，如果 m=3，在串 11101000 首尾相连构成的环中，由 3 位二进制数字组成的每个可能的子序列都在环中恰好出现一次，它们依次是 111、110、101、010、100、000、001、011，如图所示。

图 8-1　1995 年试题 7

【程序】

```
#define N 1024
#define M 10
int b[N+M-1];
int equal(int k,int j, int m)
{ int i;
for(i=0;i<m;i++)
if(b[k+1] (1) ) return 0;
return 1;
}
int exchange(int k, int m, int v)
{ while(b[k+m-1] == v)
{ b[k+m-1] = !v; (2) ;}
(3) =v; return k;
}
init(int v)
{
int k;
for(k=0;k<N+M-1;k++) b[k] = v;
}
main()
{ int m,v,k,n,j;
printf("Enter m(1<m<10),v(v=0,v=1)\n");
scanf("%d%d",&m,&v);
n=0x01<<m; init(!v); k=0;
while( (4) < n)
for(j=0;j<k;j++)
if(equal(k,j,m))
{ k=exchange(k,m,v);
```

```
j= (5) ;
}
for(k=0;k<n;k++)
printf("%d\n",b[k]);
}
```

[解析]

本程序比较短，但处处都有填空，难度比较大，着重考查考生对试探回溯算法的掌握。

程序的功能是用回溯算法来产生由 0 或 1 组成的 2^m 个二进制字串，所以主程序首先输入 m 和 v，其中 m 是一个二进制字串的位数，v 为 0 或 1。由函数 init 可知，在主程序中先将要求的二进制字串初始化为!v，从而 v 只能为 0 或 1，v 为 0 表示将二进制字串初始化为 1，第 1 个 m 位二进制字串为 m 位 1；v 为 1 表示将要求的二进制字串初始化为 0，第 1 个 m 位二进制字串为 m 位 0。接着程序通过移位运算使 n 为 2m（其实就是一个乘幂运算，但用移位速度要快得多），n 是要填写的位数，即要填写的二进制字串的元素的个数；置 k 为 0，k 为最新填写的 m 位二进制字串在 b[ ]（b[ ]为所求的二进制字串）中的开始下标。在初始化时，自 b[0]开始，已填写了一个 m 位的二进制字串，其值为!v。k+1 也可看作已在 b[ ]中正确填写的二进制数的个数，然后是一个试探求解的循环。在已有 k+1 个二进制数满足要求，且 k+1<n 的情况下循环。一般地，试探循环应包括以下工作：

(1)增 1。

(2)在 b[k+m-1]处预填入试探的值（此处工作已由循环之前的函数完成，将其置为!v），表示一个新的 m 位二进制数字已填入。其高 m-1 位是前一个 m 位二进制数的低 m-1 位。

(3)检查新的 m 位二进制数是否与前面已经填写的 k 个 m 位二进制数相等。如都不相等，本次试探完成，继续下一次试探；如有相等的，就得改变最新填入得二进制数字的值，改变后，需重新检查。若仍不符合要求，则需要回溯，改变上一次试探的二进制数字的值，若仍不符合要求，则继续回溯，直至符合要求或无解。

在新一轮试探循环开始时，已经有 k+1 个正确的 m 位二进制字串，而本次循环的目的是测试第 k+2 个 m 位二进制字串是否符合要求，即与前 k+1 个 m 位二进制字串是否不相等。如初始循环时 k=0，则已有一个正确的 m 位二进制数字（m 位全是!v），进入循环是测试从 b[1]开始的第 2 个 m 位二进制数字（m 位也全是!v）与第 1 个 m 位二进制数字是否相等。所以 while 后的条件表达式为++k<n，即天空（4）的答案为"++k"。

for 循环的功能是检查第 k+1（因为在 while 条件表达式中 k 值已加 1）个 m 位二进制数字串与前 k 个 m 位二进制字串是否相等。它首先调用函数 equal 判断第 k 个 m 位二进制字串是否与第 j 个 m 位二进制字串相等。函数 equal 比较两个 m 位二进制字串是否相等，采用的是逐位比较的方法，依次比较与 b[k+i]与 b[j+i]（0<=i<=m-1），若相等则继续比较下一位，直至最后一位仍相等，则这两个二进制字串相等，返回 1；否则，两个二进制字串肯定不相等，不用再进行下面的比较，直接返回 0 即可。所以填空（1）的答案为"!=b[j+i]"。若 equal 函数返回 1，即两个二进制字串相等，则说明这一步试探失败，此时调用 exchange 函数进行更改，包括可能的回溯，并返回更改（回溯）后的 k 值。更改（回溯）后，又要从第 1 个 m 位二进制字串开始检查第 k+1 个 m 位二进制字串是否与前面 k 个 m 位二进制字串的某个相等，即要让 for 循环重新开始（j=0）。因为 for 循环执行结束时，要先执行表达式 j++，为了使下次循环重新开始时执行，即执行 j++后，j 的值是 0，填空（5）处应置为-1。所以填空（5）的答案为"-1"。

函数 exchange 的功能是第 1 次试探失败后对 b[k+m-1]（b[k+m-1]为第 k+1 个 m 位二进制的最后一位）的值进行更改（包括回溯）。首先判断 b[k+m-1]的值，若为 v 则肯定 b[k+m-1]的值已经更改过了，必须进行回溯，此时应先恢复 b[k+m-1]的值为!v，然后 k 值减 1。继续进

行判断，若前一位的值仍为 v，则继续回溯，直至其值不等于 v 为止才停止回溯，然后将其值改为 v，并返回更改后的 k 值。若为!v 则还没有更改过，将其值变为 v 后返回。所以填空（2）的答案是"k--"，或其他使 k 减 1 的 C 代码；填空（3）的答案是"b[k+m-1]"。
最后将求得得二进制字串打印。
[答案]

（1）!=b[j+1]
（2）k--或--k 或 k=k-1 或 k-=1
（3）b[k+m-1]
（4）++k 或 k=k+1 或 k+=1
（5）-1

试题 12（1994 年试题 7）

阅读以下程序说明和 C 程序，将应填入_(N)_处的字句，写在答卷的对应栏内。
[程序说明]
由二叉树的前序遍历和中序遍历两个遍历序列能唯一确定一棵二叉树。
前序遍历为：访问根结点、访问左子树、访问右子树。
中序遍历为：访问左子树、访问根结点、访问右子树。
如图 8-2 所示的二叉树，其前序和中序遍历序列分别为：

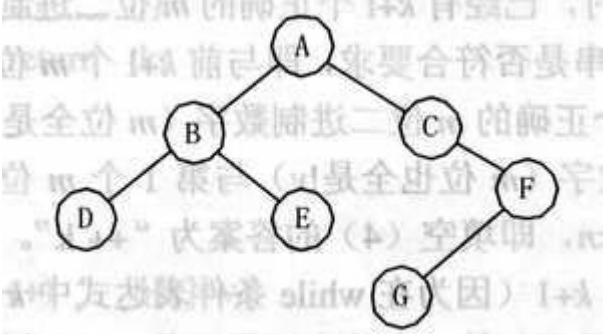


图 8-2　1994 年试题 7

pred[]：A、B、D、E、C、F、G。
inod[]：D、B、E、A、C、G、F。
本程序实现由已知某二叉树的前序遍历和中序遍历序列，生成一棵链接表示的二叉树。
构造二叉树的算法的要点是：由前序遍历序列，该序列的第一个元素是根结点元素（例中为 A）。该元素将中序遍历序列分成左、右两部分，那些位于该元素之前的元素是它的左子树上的元素（例中为 D、B、E），位于该元素之后的元素是它的右子树上的元素（例中为 C、G、F）。对于左、右子树，由它们的前序遍历序列的第一个元素可确定左、右子树的根结点，参照中序遍历序列又可进一步确定子树的左、右子树元素。如此递归地参照两个遍历序列，最终构造出二叉树。
两个遍历序列作为主函数 main()的参数。为简单起见，程序假定两个遍历序列是相容的。主函数调用函数 restore()建立二叉树。函数 restore()以树（子树）的前序遍历和中序遍历两序列及序列长为参数，采用递归方法建立树（子树）。
函数 postorder()实现二叉树的后序遍历序列输出，用来验证函数 restore()建立的二叉树。
[程序]

```
#include <stdio.h>
#include <stdlib.h>
#define MAX 100
typedef struct node{
char info;
struct node *llink,*rlink;
}TNODE;
char pred[MAX],inode[MAX];
TNODE *restore(char*,char *,int);
main(int argc, char **argv)
{
TNODE *root;
if(argc <3) exit(0);
strcopy(pred,argv[1]);
strcpy(inode,argv[2]);
root = restore(pred,inodstrlen(pred));
postorder(root);
prinft("\n\n");
}
TNODE *restore(char *ppos, char *ipos,int n)
{
TNODE *ptr;
char *rpos;
int k;
if(n<=0) return NULL;
ptr = (TNODE*)malloc(sizeof(TNODE));
ptr->info = (1) ;
for( (2) ;rpos<ipos+n;rpos++)
if(*rpos == *ppos) break;
k= (3) ;
ptr->llink = restore(ppos+1, (4) ,k);
ptr->rlink = restore( (5) +k, (6) +1,n-1-k);
return ptr;
}
oistirder(TNODE *ptr)
{
if(ptr == NULL) return;
postorder(ptr->llink);
postorder(ptr->rlink);
printf("%c",ptr->info);
}
```

[解析]

本程序的功能是已知某二叉树的前序遍历和中序遍历序列，生成一棵链接表示的二叉树。其依据是由二叉树的前序遍历和中序遍历序列能唯一确定一棵二叉树。前序遍历序列的第一个

元素即为二叉树的根结点，同时该元素也把对应的中序遍历序列分成两个部分，左边是二叉树的左子树，右边是二叉树的右子树。前序遍历序列的第二个元素是二叉树左子树的根结点，它又可将对应的左子树的中序遍历序列分成两部分。如此递归下去，则能够确定每个结点再二叉树中的位置。

整个程序的结构很简单，由主程序和两个子程序组成。主程序根据输入的前序序列和中序序列，调用子程序 restore 来建立二叉树，然后再调用 postorder 子程序后序遍历二叉树并输出，以验证 restore 建立的二叉树，postorder 功能比较简单，下面着重讲一下 restore。

子程序 restore 以二叉树的前序序列*ppos、中序序列*ipos 及二叉树的结点个数 n 为参数，返回指向二叉树的指针。当 n 小于等于 0 时，反回空。否则先申请一个二叉树结点空间，并将对应的元素的值赋给该结点，与该结点对应的是前序序列的递一个元素，所以填空（1）的答案是"*ppos"或"ppos[0]"。查找该结点元素在中序序列的位置，查找是从前往后，以循环方式进行的，一旦找到就退出循环，所以填空（2）的答案为"rpos=ipos"。接下来应该是递归分析出该结点的左子树及右子树。这一步的关键是把正确的参数传给递归子程序。

观察程序可知，填空（4）应为该层二叉树的右子树的中序序列，（5）应为该层二叉树的右子树的前序序列，（6）应为该层二叉树的右子树的中序序列，（3）应为该层二叉树的左子树的结点个数。左子树中结点的个数，就是在中序序列中根结点之前的元素的个数。所以填空（3）的答案是"rpos-ipos"。在前序序列中元素的顺序是根结点、左子树的前序序列、右子树的前序序列，所以递归调用时左子树的前序序列的指针为 ppos+1，而右子树的指针为 ppos+1+左子树的结点个数。所以填空（5）的答案是"ppos+1"。在中序序列中元素的顺序是左子树的中序序列、根结点、右子树的中序序列，所以递归调用时左子树的中序序列为，而右子树的中序序列为 ipos+1+左子树的结点个数（或为 rpos+1，rpos 在循环结束时指根结点）。所以填空（4）的答案是"ipos"，填空（6）的答案是"rpos"或"ipos+k"。

[答案]

（1）*ppos 或 ppos[0]

（2）rpos=ipos

（3）rpos-ipos

（4）ipos 或 rpos-k

（5）ppos+1

（6）rpos 或 ipos+k

试题 13（1993 年试题 7）

阅读以下程序说明和 C 程序，将应填入_(N)_处的字句，写在答卷的对应栏内。

[程序说明]

对于正整数 n，输出其和等于 n 且满足以下限制条件的所有正整数的和式，即组成和式的数字自左至右构成一个非递增的序列。如 n=4，程序输出为：

4 = 4

4 = 3 +1

4 = 2+2

4=1+1+1+1

程序中给出了分别采用递归和非递归解法的两个函数 rd()和 nd()。

函数 rd()采用递归解法，它有两个参数 n 和 k。其意义分别是被分解和式的数 n，及当前第 k 深度分解。算法思想是对 n 的所有合理的和式分解，将分解出的数（称为和数）存于数组 a[ ]中。当其中一个分解已不再需要进一步分解时，即找到下一个解，将存于数组 a[ ]中的一个完整和式的和数输出。当还需要进一步分解时，以要进一步分解的数及分解深度为参数，

递归调用分解和式函数。

函数 nd()以要分解的数为参数，另开设一个数组 r[ ]，用于存贮当前还未分解的答数。在求一个解的第 k 步时，a[k]为第 k 个和数，r[k]为相应的余数。当找到一个分解后（此步 r[k]等于 0），输出解，并作回溯处理，从当前 k 退回到第一个不为 1 的和数，将其减 1，并将其余数加 1，准备去找另一个解；否则，生成下一步的分解和数与余数。

[程序]

```
#define MAXN 100
int a[MAXN],r[MAXN]
rd(int n,int k)
{
int i,j;
for(j= (1) ;j>=1;j--)
{
a[k]=j;
if( (2) )
{
printf("%d=%d",a[0],a[1]);
for(i=2;i<=k;i++)
printf(+%d",a[i]);
printf("\n");
}
else (3) ;
}
}
nd(int n)
{
int i,k;
k=0;r[0]=n;
do
{
if( (4) )
{
printf("%d=%d",a[0],a[1]);
for(i=2;i<=k;i++)
printf(+%d",a[i]);
printf("\n");
}
while(k>0 && (5) ) k--;
if(k>0){
a[k]--;r[k]++;}
else{
a[k+1] = (6) ;
r[k+1]=r[k]-a[k+1];
k++;
```

```
}
}while(k>0);
}
int test_data[]={3,4,5};
main()
{
int i;
for(i=0;i<sizeof(test_data)/sizeof(int);i++)
{
a[0]=test_data[i];
rd(test_data[i],1);
printf("\n_____\n\n");
nd(test_data[i]);
printf("\n_____\n\n");
}
}
```

[解析]

本程序的功能是递归和非递归解两种解法，将给定的正整数 n 分解成为一个数字自左向右构成一个非递增序列的和式。主程序比较简单，只是用 3 组数据测试一下递归和非递归解两种解法的结果是否一致。本程序的关键在于递归和非递归解两种求解算法。

首先看一下采用递归解法的函数 rd( )，它有两个参数：被分解的正整数 n 和当前的分解深度 k。根据题意，要分解出来的和数序列构成一个非递增序列，因此需限制分解出来的数不能大于 a[k-1]，即对 n 分解时，最大数是 n 和 a[k-1]中较小的那个，所以填空(1)的答案是"n<a[k-1]?n:a[k-1]"。当从 n 分解出和数 j，将 j 存入 a[k]后，就可处理对余数 n-j 的分解。当 n-j 等于 0 时，即 j = = n 时，正整数 n 的分解宣告结束，a[1]至 a[k]就是 n 的一个完整的和式分解，将它们输出即可；否则，递归调用函数 rd( )对 n-j 作和式分解，此时被分解的数为 n-j，当前分解深度为 k+1。所以填空(2)的答案为"j = = n"或"a[k]= = n"；填空（3）的答案为"rd(n-j,k+1)"。

对于采用非递归解法的函数 nd( )，它只有一个参数，即要分解的正整数 n。根据试题的解法叙述，程序预设第 0 步分解的初值，k 为 0，r[0]为 n。求解过程是一个循环，直到回溯到第 0 步找出全部解。由程序说明可知，当前分解的余数 r[k]为 0 时，就找到了一个解，程序将存于数组 a[ ]中的和数按试题的格式要求输出，所以填空（4）的答案为"r[k] = = 0"。输出后，进行回溯调整处理，准备找下一个解，应当回溯至第一个不为 1 的和数为止，所以填空（5）的答案是"a[k] = = 1"。回溯后进行调整，将该步的和数减 1，余数增 1，程序已给出调整的全文。如当前分解的余数，则要确定下一步分解出的和数和余数。已知当前分解的和数为 a[k]，余数为 r[k]，下一步是对 r[k]进行分解。因为所求的和式是一个非递增的序列，所以分解出的最大和数应取 a[k]与 r[k]中的小者，所以填空（6）的答案是"a[k]<r[k]?a[k]: r[k]"。当回溯至 k = = 0 时，所有的分解都已求出，此时循环结束。

[答案]

（1）n<a[k-1]?n:a[k-1]

（2）j==n 或 a[k]==n

（3）rd(n-j,k+1)

（4）r[k]==0 或 r[k]<=0

（5）a[k]==1 或 a[k]<=1

（6）a[k]<r[k]?a[k]:r[k]

试题 14（1992 年试题 7）

阅读以下程序说明和 C 程序，将应填入_(N)_处的字句，写在答卷的对应栏内。

[程序说明]

（1）本程序利用辗转相除法求两个均不超过 100 次的多项式 A、B 的最大公因式。

例：

$A(x)=x^3-x^2+x-1=(x^2+1)(x-1)$

$B(x)=x^5-7x^4+7x^3-3x^2+6x+4=(x^2+1)(x^3-7x^2+6x+4)$

最大公因式为：x^2+1

（2）辗转相除法的算法如下：

用其中的一个多项式去除另一个多项式；然后，将所得余式变成除式，原除式变成被除式。如此反复相除，当余式为 0 时当前除式即为最大公因式。

[程序]

```
#include <stdio.h>
#include <math.n>
#define DECISE E.0005
#definde MAX_POWER 100
main()
{
int i,a,b;
float Ca[MAX_POWER+1],Cb[MAX_POWER+1];
void Remainder();
scanf("%d",&a);
for(i=0;i<=a;i++)
scanf("%d",&Ca[i]);
scanf("%d",&b);
for(i=0;i<=b;i++)
scanf("%d",&Cb[i]);
Remainder(Ca,Cb,a,b);
}
void Remainder(PointerA,PointerB,a,b)
float *PointerA,*PointerB;
int a,b;
{
float x,y,*Temp;
int i,j,Flag=1;
while(Flag)
{
i=0;
while(PointerB[i]==0)
{
i++;b--;
(1) ;
```

```
}
x=PointerB[i];
while(i<=b)
PointerB[i++] /=x;
for(i=0;i<=a-b;i++)
{
(2) ;
for(j=0;j<b;j++)
{
y = PointerA[i+1+j]-x*PointerB[j+1];
PointweA[ (3) ]=(y<DECISE && -y<DECISE)?0.0:y
}
}
Termp = PointerA;
PointerA = PointerB;
PointerB = (4) ;
a=b--;
for(Flag=0,i=0;i<b &&Flag==0;i++)
if(PointerB[i]==0.0) Flag=1;
}
printf("The Greatest Common Factor is:\n);
for(i=0;i<a;i++)
if(PointerA[i]=0.0)
printf("%5.3f*^%d%s",PointerA[i].a-i,(PointerA[i+1]<0.0)?"":"+");
printf("%5.3f\n",PointerA[a]);
}
```

[解析]

本题中函数 Remainder 用来实现求两个多项式的最大公因式。函数体内的"while(FLAG)"循环用来控制辗转相除过程。该循环中的"while(PointerB[i] = = 0"循环是求多项式 Cb[ ]中系数为非零项的最高幂次 b。根据后续程序的要求，要使指针指向幂次为 b 的那一项，并使保持为 0。因此填空（1）为"PointerB+i-- "。

双重 for 循环的功能是实现多项式相除，把余式保留在 Ca 中。内循环中每次使一个 PinterA[i]为，要把其系数赋值给 x，因此填空（2）应为"x=PointerA[i]"。考虑到两数相减，其差的绝对值可能很小，此时可视为 0。每次求得一个以后，都要做这样的判断，因此填空（3）应为"i+1+j"。

辗转相除后，余式比原 Ca[ ]降低了 a-b+1 次，所以余式系数的开始位置为"PointerA+a-b+1"，但若 a<b，前面的辗转相除并未行、执行，故填空（4）应为"(a>=b)?Temp+a-b+1:Temp"。

[答案]

（1）PointerB += i--;

（2）x = PointerA[i];

（3）i+1+j

（4）(a>=b)?Temp+a-b-1:Temp

试题 15 （1991 试题 7）

阅读以下程序说明和 C 程序，将应填入_(N)_处的字句，写在答卷的对应栏内。

[程序说明]

本程序用来对英文文献中出现的单词按下列计算公式统计其使用频度：

某单词的使用频度＝ 该单词的使用次数/文献中单词的总数

程序中用一个根结点为的分类二叉树存放文献中的每个英文单词及其在文献中的使用次数。对于二树中的每个结点中的单词，按字典序比较大小，小的存放在它的左子树中，大的存放在其右子树中。当统计完毕后，程序将按中序遍历该分类二叉树，得到下一个按英文单词字典分类的单词表（），并且按单词的使用次数为关键字降序输出统计结果。

程序调用了如下一些函数：

1、函数将某单词按二分查找法插入分类二叉树中，并统计该单词在文献中的使用次数。

2、函数按中序遍历分类二叉树，并形成相应的单词表中序遍历，即先访问其左子树，然后访问根结点，最后访问其右子树。

3、函数从存放英文文献的数据文件""中取出一个单词，放入字符数组中。如果遇到文件结束标志，则返回 0；否则，返回 1。为简单起见，假定每个英文单词长度不超过 20 个字母。

4、函数将单词表以单词的使用次数为关键字，按降序排序。

5、函数输出单词表中各单词和它的使用频度。

6、函数将字符串拷贝至字符串。

7、函数按字典顺序对两字符串和进行比较。倘若大于，则返回一个正整数；若等于，则返回 0；若小于，则返回一个负整数。

程序清单中省去了上述编号为 3、4、5、7 的函数的内容。

[程序]

```
#include <stdio.h>
#include <alloc.h>
#define EMPLY 0
#define MAXLEN 21
typedef struct tree
{ char word[MAXLEN];
int count;
struct tree *leftson;
struct tree *rightson;
}BINARY;
typedef struct
{
char word[MAXLEN];
int count;
}LIST;
BINARY *treeroot=EMPTY;
main()
{
FILE *fp,*fopen();
char keyword[MAXLEN];
int total_word=0;
fp=fopen("TEXT.H","r");
while(getword(keyword,fp))
```

```
{
binary_tree(keyword);
total_word++;
}
fclose(fp);
reault=(LIST*)malloc(total_word * sizeof(LIST));
travel_tree(treeroot,result);
sort_list(result,total_word);
display_list(result,total_word);
}
binary_tree(keyword)
char * keyword;
{
BINARY *ptr,*temp;
int strcmp_reault;
temp=EMPTY;
ptr=treeroot;
while(ptr=EMPTY)
{
strcmp_result=strcmp(keyword,ptr->word);
if(strcmp_reault==0)
{
ptr->count++;
return;
}
else
temp=ptr;
if(strcmp_reault>0)
ptr-> (1) ;
else
ptr = (2) ;
}
}
ptr=(BINARY *)malloc(sizeof(BINARY));
ptr->rightson=ptr->leftson=EMPTY;
ptr->count=1;
strcpy(keyword,ptr->word);
if(temp==EMPTY)
treeroot=ptr;
else
if(strcmp_reault>0)
temp-> (3) =ptr;
else
temp-> (4) =ptr;
```

```
}
travel_tree(root,result)
BINARY *root;
LIST result[];
{
static x=0;
if(root==EMPTY)
(5) ;
else
{
travel_tree( (6) ,result);
strcpy(root->word,result[x].word);
result[x++].count=root->count;
}
travel_tree( (7) ,result);
}
strcpy(s1,s2)/*copy string s1 to s2*/
char *s1,*s2;
{
while( (8) ='\0');
}
```

[解析]

本程序的功能是用来统计英文文献中单词的使用频度的。本题的程序比较长，乃至的函数也比较多，似乎很麻烦，其实不然。程序开始定义了一个二叉树的结点类型和一个线性表的结点类型。在主程序中，先从文件中逐个读入单词，并调用函数 binary_tree 将其插入到分类二叉树中，同时统计总单词量。然后，调用函数 bravel_tree 中序遍历生成的分类二叉树，并将结果放入线性表 result 中。接着调用函数 sort_list，按单词的使用次数对线性表 result 排序。最后，将排好序的二叉树打印出来。主程序比较简单，除了一个 while 循环外，其他语句都是顺序执行。下面我们看重分析函数 binary_tree 和 travel-tree。

首先看一下函数 binary_tree。binary_tree 的功能是将一个单词插入至分类二叉树中。分类二叉树，即排序二叉树中、，其插入算法是：

1、若二叉树为空，则申请一个结点，将关键字 keyword 赋给它；

2、若关键字大于根结点的关键字，则在右子树上插入；

3、若关键字小于根结点的关键字，则在左子树上插入。

对照程序，首先进行判断，若该结点不空，则比较 keyword 与根结点的关键字。若相等，即该单词已在二叉树中存在，则将该结点对应的 count 字段加 1，并返回；否则，保存指向该结点的指针（即 temp 指针指向下一次循环查找的结点）。若 keyword 大于该结点的关键字结点的指针（即 temp 指针指向下一次循环查找的结点）。若 keyword 大于该结点的关键字(strcmp_result>0)，则在其右子树上查找插入，融在其左子树上查找插入，下一步就以该结点的左子树（或右子树）为新的开始点进行查找插入，一直到找到譔单词的结点或结点为空（该单词在二叉树中不存在）为止。所以填空（1）的答案是："ptr->rightson"，填空（3）的答案是"ptr->result>0"。若未在 while 循环中退出函数，则表明单词 keyword 不在二叉树中，需要将其插入。先申请一个结点空间，指针 ptr 指向刚申请到的空间地址，令 ptr 的左右子树为空，将其 count 字段.置为 1，将 keyword 赋给其 word 字段。若指针 temp 为空，则表明

二叉树为空，将 ptr 赋给二叉树的根结点指针；若 temp 不为空，则由于 temp 指针指向下一次循环查找的结点，而在最后一次循环中结点 ptr 为空，所以在退出 while 循环后，temp 指针的左子树或右子树为空（由 strcmp_result 决定），即我们要插入的结点应该是 temp 的左儿子或右儿子。当 strcmp_result>0 时，要插入的结点应该是 temp 的右儿子，否则，要插入的结点应该是 temp 的左儿子。所以填空（3）的答案是"rightson"，填空（4）的答案是"lsftson"。

函数 travel_tree 按中序遍历该分类二叉树，并形成相应的单词表，将其存放于线性表 result 中。所谓中序遍历，即先访问其左子树，然后访问根结点，最后访问其右子树。二叉树的遍历一般采用递归算法。本函数就是采用递归来实现中序遍历分类二叉树。若当前结点为空则返回，否则先递归遍历其左子树，然后访问该结点，再遍历右子树。对照程序，显然，填空（5）的答案是"return"，填空（6）的答案是"root->leftson"，填空(7)的答案是"root->rightson"。

静态变量 x 的作用是指示数组 result 的下标，用以确定在递归调用时结点存入数组 result 的位置。

函数 strcopy(s1,s2)的功能是将字符串 s1 拷贝至字符串 s2。正常的作法是用一个循环一个字符一个字符地拷贝，当遇到字符串结束符'\0'时结束。而此处只用一条语句就要实现这个功能，难度比较大。其实只要我们掌握了 C 语句里的指针和"＋＋"运算符，做这道题訑不是太难。填空（8）的答案为"(*s2++ = = *s1++)"。*s1++的值指 s1++的是之前的所指的字符，待取出后才改变 s1 的值，同理，先将字符存入 s2 所指的位置，再改变 s2 的值。当遇到字符串结束符'\0'时结束（此时'\0'也拷贝给了 s2），否则将把*s1++!='\0'的比较结果赋给*s2++。

[答案]

（1）ptr->rightson
（2）ptr->leftson
（3）rightson
（4）leftson
（5）return
（6）root->leftson
（7）root->rightson
（8）(*s2++=*s1++)

试题 16 （1990 年试题 7）

阅读以下程序说明和 C 程序，将应填入_(N)_处的字句，写在答卷的对应栏内。

[程序说明]

设对于一个 n*n 的上三角矩阵 a，为节约存储，只将它的上三角矩阵元素按行主序连续存放在数组 b 中。下面的函数 trans 在不引入工作数组的情况下，实现将 a 改为按列主序连续存放在数组 b 中。

设 n=5，

$$a=\begin{vmatrix}1&2&3&4&5\\0&6&7&8&9\\0&0&10&11&12\\0&0&0&13&14\\0&0&0&0&15\end{vmatrix}$$

b=(1，2，3，4，5，6，7，8，9，10，11，12，13，14，15)

经调用 trans 函数后，b 变为

b=(1，2，6,3，7,10,4，8,11,13,5,9，12,14,15)

函数 trans 对数组元素的存储位置用调整。调整过程中存在若干个循环传送链：
b(i1)→b(i2)…b(ij) →b(i1) 1≤j<n
例如，考查调整后的数组元素 b(2)（值为 6），与该元素相减的位置调整将形成下面的循环传送链：b(2) →b(3) →b(6) →…b(12) →b(9) →b(5) →b(2)。
关键是确定循环传送链的下标 i1 , i2 , …ij ，以及在考查调整后的元素 b(k)(k=3,4,…)时能判定 b(k)是已传送过的某传送链上的元素。
函数 ctr(k,n)计算调整后的数组 b 的第 k 个元素 b(k)在原数组 b 中的位置，该位置作为函数 ctr(k,n)的返回值。函数 ctr 根据 k 确定它在矩阵中的行号 i 和列号 j（注意行号和列号均从 0 算起），然后按矩阵存放原则计算出它在 b 中的位置。
[程序]

```
trans(b,n)
int n,b[];
{
int m,k,r,cc,rr;
int w;
m=(n+1)*n/2-4;
k=2;
while(m>0)
{
r=ctr(k,n);
if(r==k)
m--;
else
{
cc=k;rr=r;
while( (1) )
{cc=rr;rr=ctr(cc,n);}
if ( (2) )
{
cc=k;rr=r;w=b[k];
while( (3) )
{
b[cc]=b[rr];m--;
cc=rr;rr=ctr(cc,n);
}
b[cc]=w; (4) ;
}
}
k++;
}
}
ctr(k,n)
int k,n;
{
```

```
int i,j;
i=k;j=0;
while( (5) )
i-=++j;
return (i*n+j-i*(i+1)/2);
}
```

[解析]

由程序说明可知，从初始按行存放的数组转换为按列存放的数组——不妨设为 bb[ ]的过程中，不能使用内存数组，而是使用循环传送链。一旦一条传送链求出，这条传送链上的元素便可依次调整成功，以后再遇上这条链上其他元素时便无需再调整。

由例 a[5][5]可知，该矩阵只有一条传送链 b(2) →b(3) →b(6) →b(4) →b(10) →b(8) →b(11) →b(9) →b(5) →b(2)，另外 b(7)=8 未动。

程序中的值是可能需要调整的元素个数（数组 b[ ]中的前两个和末两个不需要调整）。

1．子函数 ctr(k,n)的功能：先求出 bb[k]（按列存放）在 a[ ][ ]中的行号 i、列号 j，然后，按照 i、j 求出 a[i][j]在 b[ ]（按行存放）中的位置 r，并将 r 作为返回值，r=i*n+j-i*(i+1)/2。"while((5))"是求 i、j 的值。在此循环中，从 i=k,j=0 开始，每次 j+1→j ，然后 i-j→i，故若 i≤j，则不再循环，（5）处应填"i>j"。

2．将 b[r]放入 bb[k]。

若 r=k 则不必再处理；否则需求出以 k 为首的传送链。初始 k=2 时，以 bb[2]为首的链中的其他元素 bb[p](p>2)均有 p>2。由此可知，设当前位置为 k，则若在以 k 为首的链中出现小于 k 的位置的元素时，说明链上的元素已调整，从而无需再调了。

"while((1))"循环便是求以 k 为首的传送链。故（1）处应填入"rr!=k && cc>=k"。rr!=k 即此链还未求完（首尾未相接）；cc>=k 即当前位置大于 k。"if((2)) "之后则可依次传送，故（2）处应为"cc>=k"，表明当前链上的元素未调整过。"while((3))"循环便是传送了，故（3）处应为"rr!=k"，直至链尾遇到链首（链首位置为 k）。填空（4）处之前的"b[cc]=w(w=b[k]"是将链首位置的值放入链尾。既然此处又调整了一个元素，则（4）外显然应为"m--"。

[答案]

（1）rr!=k&&cc>=k

（2）cc>=k

（3）rr!=k

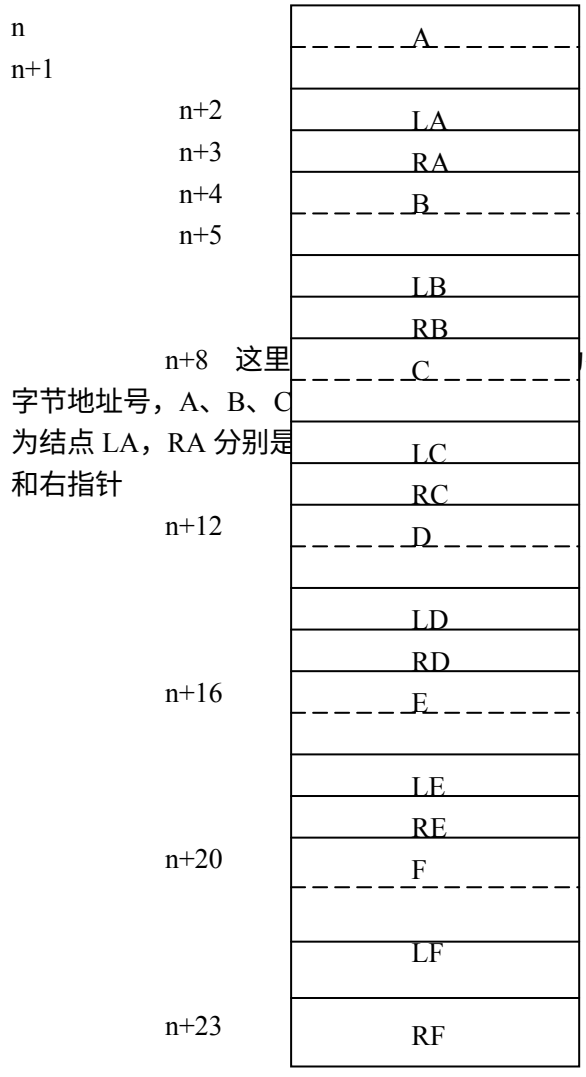
（4）m--

（5）i>j

八 2001 年度上午试题分析与解答

试题（1）~ （5）分析

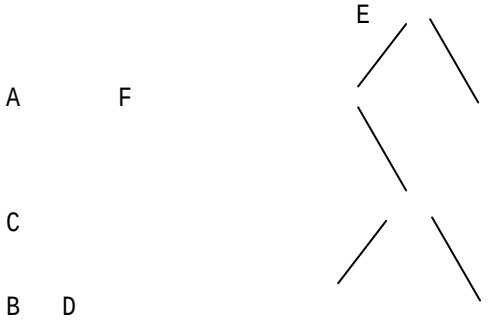
查找二叉树是一种既方便结点的增删，又具有二分法那样的快速查找功能。查找二叉树存储时，把结点依次存放，每个结点有 3 个区：数据元素值区、左指针区、右指针区。它们分别存放数据值。该结点左子女结点的存储地址，该结点右子女结点的存储地址。
结点在查找二叉树中存放必须满足：根结点的关键码值大于左子树中所有结点的关键码值，小于右子树中所结点的关键码值。
查找二叉树在查找时，将待查的关键码值与根结点的关键码值比较。若相等则查到；否则根据比较的大小结果，进入左子树或右子树，再重复上述过程，直至查到为止。因此，它具有二分法的查找速度。



这里 存储的字节地址号，A、B、C 为结点 LA，RA 分别是 和右指针

本题中有结点 6 个，其关键码值分别为 A、B、C、D、E、F，此时值的大小是按字典顺序确定的。按题中意思，其存放情况如上页图所示。
题中指定 E 是根结点，由于 A、B、C、D 的值都小于 E，仅有 F 大于 E，因此，该查找二叉树仅有一个结点属于右子树。而 A、B、C、D 属于左子树，因前序遍历是 NLR 顺序，因此其前序序列必为 E×××F 形式。在本题供选择答案中（1）所给出的前序序列只有最

后一个是上述形式，从而确定其前序序列为 EACBDF，且知道二叉树的左子树是 ACBD，再考虑前序遍历的性质和 A 是左子树的根结点，从而 CBD 均是 A 结点下的右子树，再由前序遍历的特性知 C 是 A 结点下右子树的根。最后由二叉查找树的特性知 B 是 C 的左子树，D 是 C 的右子树，从而得到该二叉查找树的形状如下：



立即可得层次遍历序列为 EAFCBD。

这样，就知道 Lc 中应存放 C 的左子树 B 的地址。由于 B 存放在 n+4 开始的单元中，所以 Lc 的内容应是 n+4。Ra 应存放 A 的右子树的根结点 C 的地址，即 Ra 的内容是 n+8。

解答

（1）D　（2）A　（3）B　（4）A　（5）B

试题（6）~ （10）分析

软件开发工具是指用于辅助软件开发、运行、维护、管理、支持等过程中的活动的软件，通常也称为 CASE（computer aided software engineering　计算机辅助软件工程）工具。

一个软件工具通常只为某项活动提供辅助，或者只支持某种方法。并且，不同的工具往往采用不同的用户界面风格，工具之间难以通信，一个工具所产生的结果不易被另一个工具使用。集成型软件开发环境是一种把支持多种软件开发方法和开发模型、支持软件开发全过程的软件工具集成在一起的软件开发环境。这种环境通常应具有开放性和可剪裁性。开放性为环境外的工具集成到环境中来提供方便；可剪裁性可根据不同的应用或不同的用户需求进行剪裁，以形成特定的开发环境。

集成型开发环境通常可由工具集和环境集成机制两部分组成。环境集成机制主要有数据集成机制、控制集成机制和界面集成机制。数据集成机制为相互协作的工具提供统一的数据接口规范；控制集成机制支持各工具间的通信、切换、调度和协同工作；界面集成机制为统一的工具界面风格和统一的操作方式提供支持，使得环境中的工具具有相同的视觉效果和操作规则。

解答

（6）D　（7）A　（8）C　（9）B　（10）C

试题（11）~ （12）分析

在应用计算机解决问题时，算法的设计是一项不可缺少的又是困难的任务。特别是在大型的科学计算与复杂问题求解时，具体的算法针对具体问题，可以有无数种。但一般常用类型有：迭代法、递推法、回溯法、递归法、贪婪法、穷举法、规划法等。

递归算法是常用算法之一，它的基本思想是把一个大的问题（不妨设规模为 N）分解为一些规模较小的问题，从这些较小问题的解，构造出大问题的解，而这些规模较小的问题，用同

样的方法分解成更小的问题，并从更小问题的解，构造出较小的问题。不断地分解问题成更小的问题，一层层下去，一般来说，分解到最后的小问题（例如 N＝1 时）应该是可直接求解的。

依据上述思想，递归算法的过程一般分为递推和回归两个阶段。在递推阶段把复杂问题逐级推到简单一点的问题的求解。在递推到小问题能直接求解时，递推终止。在回归阶段，则从所获得的最简单的直接解，逐级返回，得到大问题的解。

解答

（11）B　　（12）B

试题（13）~ （14）分析

递推算法是一种常用算法，它的基本思想是：对本身具有递推关系的问题，其初始解已知或者很容易得到，然后从 i=0 开始，逐级从 i=0，1，2，…进行递推，每次是从上一次递推的结果开始，利用递推关系，求出下一次的递推的结果，直到符合要求为止。

递归算法相对递推算法要复杂得多。递归算法中是递推分解问题，然后再将最简单情况的解回归成大问题的解。由于递归会引起一系列函数调用，有不少重复计算，其执行的效率也较低。

因此，若某问题即能用递归算法求解，又能用递推算法求解时，常常是使用递推方法求解要容易，效率高得多。

解答

（13）D　　（14）A

试题（15）分析

对一些大的复杂问题，要求取精确的最优解，常常是很困难的。这里还包括数学模型的确切与精确性。此时人们就退而求其次，能得到一个满足需要的解就可以了，即满意解或次最优解。

贪婪法所追求的目标就是不追求最优解，希望得到满意解。

贪婪法的基本思想是：不追求整体问题的最优解，而是从当前、局部情况为基础求取最优解，再逐步成为整体解。由于局部最优解未必是整体最优，因此所得的解是次最优解，即满意解。贪婪法大大节省了为求整体最优解而耗费的大量精力，而得到的是一个可以满意的解。

解答

（15）A

试题（16）~ （20）分析

一个文法 G 可用一个 4 元组 G＝（V_T，Vn，S，P）来表示。其中 V_T 是非空的终结符集，Vn 是非空的非终结符集，S∈Vn 是文法的开始符号，P 是形如 α→β 的产生式集。当 α∈Vn，β∈（V_T ∪Vn）*时，该文法是上下文无关文法。本题给出的就是一个上下文无关文法。

若 α→β∈P，γ 和 δ∈（V_T ∪Vn）*，则称 γ α δ $\Rightarrow$ γ β δ 为 γ α δ 到 γ β δ 的一个直接推导；若 γ α δ $\overset{*}{\Rightarrow}$ γ β δ，则称 γ β δ 是 γ α δ 的一个推导。若 S $\overset{*}{\Rightarrow}$ α，α∈（V_T ∪Vn）*，则称 α 是该文法的一个句型；若 S $\overset{*}{\Rightarrow}$ α 且 α∈V_T*，则称 α 是该文法的一个句子，

即仅包含终结符的句型称为一个句子。
如果在推导的任何一步 α ⇒ β（其中 α 和 β 都是句型），都是对 α 中的最右（最左）非终结符进行替换，则称这种推导为最右（最左）推导，其中最右推导也称为规范推导。

若 $S\overset{*}{\Rightarrow}\alpha A\delta$，$S\overset{*}{\Rightarrow}\alpha\beta\delta$，$A\overset{+}{\Rightarrow}\beta$，则称 β 是句型 α β δ 相对于非终结符 A 的短语；特别当 $A\Rightarrow\beta$ 时，称 β 是句型 α β δ 相对于产生式 A→β 的直接短语。一个句型的最左直接短语称为该句型的句柄。
素短语是一个短语，它至少含有一个终结符，而且除它自身以外不再含有更小的素短语。
根据上述定义，可对本试题进行分析。
1．句型 F*F＋T 的推导过程如下：

$E\Rightarrow E+T\Rightarrow T+T\Rightarrow T*F+T\Rightarrow F*F+T$，由于 $E\overset{*}{\Rightarrow}T*F+T$，$E\overset{*}{\Rightarrow}F*F+T$，$T\Rightarrow F$，所以 F 是句型 F*F＋T 相对于产生式 T→F 的直接短语，由于它是该句型的最左直接短语，所以 F 是该句型的句柄。
同理，可分析出句型 F*F＋T 的短语有 F，F*F，F*F＋T。由于素短语中至少应含有一个非终结符，所以 F 不是素短语；由于 F*F＋T 中包含了短语 F*F，所以它也不是素短语。因此该句型的素短语是 F*F。
2．因为句型 $F*F+T\Rightarrow F*F+T*F$，所以 F*F＋T*F 是该句型的直接推导。而 F*F＋i，F*F＋F*F,i*i+T 都不能由句型 F*F＋T 直接推导出来。
3.由于最左推导是对句型右部的最左非终结符进行推导，所以在（19）的供选择答案中只有（E）*F＋T 满足此条件，所以（E）*F＋T 是句型 F*F＋T 的最左推导。
4．由于句子是仅含终结符的句型，所以（20）的供选择答案中只有 i 有可能是句子。因为 $E\Rightarrow T\Rightarrow F\Rightarrow i$,所以 i 是该文法的一个句子。

解答

（16）A　　（17）B　　（18）B　　（19）D　　（20）C

试题（21）～（25）分析

软件开发模型是指软件开发全部过程、活动和任务的结构框架。常用的软件开发模型有瀑布模型、演化模型、螺旋模型、喷泉模型等。
瀑布模型给出了软件生存周期各阶段的固定顺序，上一阶段完成后才能进入下一阶段。演化模型是在快速开发一个原型的基础上，根据用户在试用原型的过程中提出的反馈意见和建议，对原型进行改进，获得原型的新版本。重复这一过程，直到演化成最终的软件产品。螺旋模型将瀑布模型和演化模型相结合，它综合了两者的优点，并增加了风险分析。它以原型为基础，沿着螺线自内向外旋转，每旋转一圈都要经过制订计划、风险分析、实施工程、客户评价等活动，并开发原型的一个新版本。经过若干次螺旋上升的过程，得到最终的软件。
喷泉模型主要用来描述面向对象的开发过程。它体现了面向对象开发过程的迭代和无间隙特征。迭代意味着模型中的开发活动常常需要多次重复；无间隙是指开发活动（如分析、设计）之间不存在明显的边界，各项开发活动往往交叉迭代地进行。

解答

（21）C　　（22）A　　（23）D　　（24）C　　（25）B

试题（26）～（27）分析

MIDI 是英文 musical instrument digital interface 的缩写。英文的直译就是乐器数字接口。它是由世界上主要电子乐器厂商建立起来的一个通信标准，与多媒体结合后，日趋完善。
MIDI 音频是多媒体计算机产生声音（特别是音乐）的主要方式之一。
MIDI 文件记录的不是声音本身，不是声波的采样值。它是把每个音符记录为数字，记录着定时、音长、音量、力度、通道信息等。其文件是一系列的指令。由此 MIDI 文件既有强大的功能，又节省大量的存储空间。例如半小时的立体声音乐，如用波形文件记录，大约要 300MB，而 MIDI 文件只要 200KB 就够了，这是 MIDI 文件的重要特色。

解答

（26）B　　（27）A

试题（28）~ （29）分析

MPEG 是英文 moving pictures experts group 的缩写。其目标是为视频、音频制定数据压缩标准。MPEG 的各种版本已陆续面世。
MPEG 相比于其他数据压缩方案有下列优点：兼容性强；压缩比高；在高压缩比下的数据失真相对很小。
MPEG 的视频压缩技术是针对动态图像的。为了提高压缩比，帧内图像的数据压缩与帧间图像的数据压缩同时进行，它采用 DCT 变换技术，预测压缩算法和插补法等。使数据的冗余大为减少。失真很小的情况下，压缩比提高到 200。

解答

（28）B　　（29）B

试题（30）分析

MPEG 于 1988 年开始工作，1992 年制订出 MPEG－I，主要是考虑到工业级标准，用于数字电话网络上的视频传输，也可在 Internet 上传输音频用。对动态图像的压缩编码速率可达 1.5Mbps 到 5Mbps。
1994 年制订出 MPEG－II，有更高的图像质量，也有更高的速率（达到 10Mbps），可播放 VCD，提供 CD 级的音质，还可提供较广的范围改变压缩比，以适应不同画面质量，存储容量及带宽要求。
MPEG－III 在设计过程中，尚未推出就被抛弃了。
MPEG－IV 的设计是为了适应交互计算机和交互性电视技术的发展。为了具有更高的数据压缩比，更高的图像质量，更快的传输速率，突破了 MPEG－I 和 MPEG－II 中的文法，采用基于内容的压缩编码方法，使 MPEG－IV 具有更强有交互能力的特点。

解答

（30）C

试题（31）~ (35)分析

关系数据库设计理论主要包括个方面的内容：数据依赖、范式和关系模式规范化。其中起核心作用的是数据依赖，范式和关系模式规范化都是在数据依赖的基础上定义和发展而来的。数据库的完整性是指数据的正确性和相容性，这是数据库理论中的重要概念。完整性控制的主要目的是防止语义上不正确的数据进入数据库。关系模型中的完整性约束条件包括实体完整性、引用完整性和用户定义完整性。

实体完整性要求关系中的元组在组成主键的属性上不能有空值。
引用完整性要求“不允许引用不存在的元组”。
联接有两种：θ联接和 F 联接。θ联接操作是从关系 R 和 S 的笛卡儿积中选取属性之值满足某一θ运算的无组，记为$R\underset{i\theta j}{\bowtie}S$，这里 i 和 j 分别是关系 R 和 S 中的第 i 个、第 j 个属性的序号，θ是比较运算符。形式定义如下：

$R\underset{i\theta j}{\bowtie}S \equiv \sigma_{i\theta(r+j)}(R\times S)$，这里 r 是关系 R 的元数。$R\underset{i\theta j}{\bowtie}S$是在 R 和 S 的笛卡儿积中挑选第 i 个分量和第（r+j）个分量满足θ运算的元组。如果θ是等号“＝”，那么这个联接操作称作等值联接。
按照θ联接的定义，第（34）题的答案 C、D 显然错误，这里 r＝2，i＝1，j＝2
θ联接条件 1＜2（iθj），在笛卡儿积中应改为 1＜4（iθ(r+j)），故正确答案应为 B。
SQL 中的 SELECT－FROM－WHERE 句型与关系代数表达式的关系是：

```
SELECT  A1，A2，…，An
FROM R1,R2, …,Rm
WHERE F
```

等价于$\pi_{A1,A2,\cdots,An}(\sigma_F(R_1\times R_2\times\cdots\times R_m))$。

这里的笛卡儿积，有时为自然联接，由具体情况而定。
第（35）题是要查出不选修课程号为“c102”的学生的学号和姓名。为保证 S 和 SC 两表中对应的是同一个学生，故必须作自然联接。答案（B）的减号前部分是查出全部学生的学号和姓名，减号后部分是查出选修课程号为“c102”的全部学生的学号和姓名，进行集合差操作得到不选修课程号为“c102”的全部学生的学号和姓名。

解答

（31）C　　（32）C　　（33）A　　（34）B　　（35）B

试题（36）～（40）分析

当多个任务竞争同样的两个或多个临界资源时，就会出现死锁。产生死锁的必要条件是互斥条件、不可抢占条件、保持与等待条件、循环等待条件。
采用生产者一消费者方式解决同步与互斥时，通常需要 3 个信号量 empty、full 和 mutex，empty 是表示空缓冲区数目的信号量，full 是表示满缓冲区数目的信号量，mutex 是对临界缓冲区进行操作的互斥信号灯。
虚拟存储器是内存一定程度上的扩展，使得程序可以在透明的情况下访问比内存大得多的地址空间，使运行内存需要大于内存实际容量的程序成为可能，操作系统从硬盘中分配一部分空间作为虚拟存储器，在需要时同内存中的页面进行相互替换。
虚存的页面调度算法有多种：
先进先出算法（FIFO，first input first output），又称轮转法（RR）：即先进入主存的页，先退出主存。这种算法实现起来较简单，只要系统保留一张次序表即可，对于具有按线性顺序访问地址空间的程序而言是比较合适的，而对其他情况则效率不高。
循环检测法；
最近最少使用页面先淘汰（LRU，least recently used）：当需要转换一页时，选择最长时间未被使用的那一页淘汰掉。该算法的出发点是如果某页访问则该页可能再被访问。该算法对于循环多的程序有利；

最不经常使用的页面先淘汰（LFU，least frequent used）；
最近没有使用页面先淘汰（NUR）；
最优淘汰算法（OPT，optimal replacement algorithm）；
随机数淘汰页面算法（random replacement algorithm）。
实存的可变式动态分区分配在作业执行前并不建立分区，而是在处理作业过程中按需要建分区。有以下几种分配算法：
首次适应法：把内存中的可用分区单独组成可用分区表或可用分区自由链，按起始地址递增的次序排列。每次按递增次序向后找。一旦找到大于或等于所要求内存长度的分区，则结束探索，从找到的分区中找出所要求内存长度分配给用户，并把剩余的部分进行合并；
循环适应法：上述首次适应法经常利用的是低地址空间，后面经常可能是较大的空白区，为使内存所有线性地址空间尽可能轮流使用到，每重新分配一次时，都在当前之后寻找；
最佳适应法：最佳适应算法是将输入作业放入主存中与它所需大小最接近的空白区中，将剩下的未用空间最小，该法要求空白区大小，从小到大次序组成空白区可用表或自由链。在进行分配时总是从最小的一个开始查寻，因而找到的一个能满足要求的空白区便是最佳的一个；
最差适应法：分配时把一个作业程序放入主存中最不适合它的空白区，即最大的空白区（空闲区）内。
在文件存储设备管理中，有 3 类常用的空闲块管理方法，即位图向量法、空闲块链表法和索引法。位图向量法中位图的每个字的每一位都对应一个物理块。空闲块链表法是把所有的空白块链在一起，当创建文件需要一块或几块时，就从链头上依次取下，反之，回收空间时，把这些空白块依次链到链头上。索引法是文件存取器上每一个连续的空闲区建立一个索引，这种分配技术适于建立连续文件。

解答

（36）D　　（37）C　　（38）A　　（39）B　　（40）D

试题（41）~(43)分析

可以利用二进制加法器对二-十进制编码的十进制数求和，但由于十进制数是逢十进一，而二进制运算在第 4 位上是逢十六进一，并且在二-十进制编码中大于 1001 时（即为 1010、1011、1100、1101、1110 和 1111）都是非法编码，故做加法后有时需要对和进行修正。
当和的本位十进制数二-十进制编码小于等于 1001 且向高位无进位时显然不需要进行修正。
例如，十进制数 17＋21 采用二-十进制编码及二进制加法器进行，结果如下

```
  00010111
＋）00100001
  ---------
  00111000
```

其结果就是和 38 的二-十进制编码。换句话说，（41）的正确答案是 A。
当和小于等于 1001 且向高位有进位时说明本位十进制数之和大于 16，则需要进行加 6 修正。
例如，十进制数 19＋28 采用二-十进制数编码及二进制加法器进行，结果如下：

```
  00011001
＋）00101000
  ---------
  01000001          （有进位）
```

若不进行修正，变成 41 就不对了；进行加 6 修正后，得

```
01000001
＋）    0110
```

01000110
是 47 的二-十进制编码，是正确的。故，（42）的正确答案是 B。
当和大于 1001 时，对于十进制数位来说产生的是非法编码，而应该向高位进位，但二进制加法器却未进位，这时也需要进行加 6 修正。例如十进制数 18＋23 采用二-十进制编码及二进制加法器进行，结果如下：

```
  00011000
＋）00100011
  00111011
```

1011 是二-十进制的非法编码，进行加 6 修正后，得

```
00111011
  ＋）  0110
01000001
```

是 41 的二-十进制编码，是正确的。故，（43）的正确答案也是 B。
解答
（41）A　　（42）B　　（43）B

试题（44）~(45)分析

现代微机大都采用了多层总线的结构，如 CPU 模块内部有内部总线，输入/输出系统内部有输入/输出总线，而存储器模块内部也有存储器总线；CPU 模块、输入/输出系统和存储器模块之又通过系统总线互连在一起。
常用的微机系统总线有工业标准体系结构 ISA（industry standard architecture）、扩展的工业标准体系结构 EISA（extended ISA）、VESA(video electronics standards association)总线以及外围部件互连 PCI（peripheral componet interconnection）总线等。ISA 是早期的工业标准，由于其数据总线宽度只有 16 位，数据传输率不高，目前已不大使用；EISA 是对 ISA 的扩展，数据总线及地址总线宽度皆为 32 位，但向下仍与 ISA 兼容。所以，（44）的正确答案是 C；VESA 是由美国视频电子标准协会提出的一种总线结构，但它未提供数据缓冲器，当处理器速度高于 33MHz 时会引起处理延迟；PCI 是由 Intel 公司开发并首先应用于奔腾（Pentium）机的总线，其数据线宽度为 64 位，并有数据缓冲能力，因而其处理突发数据流的传输能力优于 VESA。所以，（45）的正确答案是 D。
在本题供选择的答案中还有 SCSI（small computer system Interface），它是一种连接高速输入/输出设备的接口，也可以说它是输入/输出总线，但不能用作为系统总线。
解答
（44）C　　（45）D

试题（46）~ （50）分析

逻辑左移移位指令对无符号数进行移位。最高位“ 进入 ”位，其余所有位接收其相邻低位值，最低位移入 0。计算机常用 2 的补码系统方式存取数据，即最高位是符号位，0 表示正数，1 表示负数，对正数用它的原码表示，对负数用它的绝对值的 2 的补码表示。被操作数的最高为保持不变，即数的符号表示不变。其余所有位接收相邻高位值，最低位移到“ 进位 ”中的操作是算术右移，即除 2 运算。
计算机中按程序计数器中的内容读出相应地址中的指令，执行此指令，并根据该指令的长度（占用的地址数）修改程序计数器的值，从而可取得下一条指令，再执行。只有转移指令，

它把即将执行的下一条指令的存放地址直接送入程序计数器，改变了程序计数器顺序读出指令的过程，使指令执行转向另外一段程序。
计算机中使用了多种寻址方式来获得运算需要的操作数。在使用相对寻址方式下，所需的实际地址等于程序计数器的内容加上指令中形式地址值，即实际数的位置是相对指令所在的地址给出，其位移量由指令给出。
特权指令是指具有特殊权限的指令。由于这些指令的权限最大，若使用不当，会破坏系统和其它用户信息，因此这类指令只用于操作系统和其他系统软件，不直接提供给用户使用。在多用户、多任务的计算机系统中，要分配和管理系统资源，例如要改变系统工作方式，检查用户的访问权限，修改虚拟存储器管理的段表、页表，完成任务的创建和切换等都需要使用特权指令。

解答

（46）A　　（47）D　　（48）D　　（49）A　　（50）D

试题（51）~ （55）分析

并行性是指在同一时刻或在同一时间间隔内完成两种或两种以上的工作。其要点是在完成时间上相互重叠。严格说起来，并行性分为同时性和并发性两种。前者指多个事件在同一时刻发生，后者指多个事件在同一时间间隔内发生。供选择的答案中的多道程序，多用户和非单指令流单数据流方式工作等都有并行性工作，但不能作为定义。计算机系统可采取多种措施来提高并行性，这此措施可分为三类，即资源重复、时间重叠和资源共享。例如采用多个处理器一起进行处理属于资源重复，流水线结构使多个处理过程在时间上互相错开，轮流重叠使用硬件设备的各个部分属于时间重叠，而多道程序，分时系统便是资源共享。同样，题中其他答案有些不属于并行性工作，有些仅是具体的并行性工作方式，不能作为并行性措施的分类。
阵列处理机是采用多个处理机各自对共给的数据作同样的操作，因此它是以 SIMD，单指令流多数据流方式工作。在矩阵运算或图像处理的应用中，运算量很大，但它们是对大量单元作相同的运算，因此最适宜采用阵列处理机来进行并行处理。
多处理机属于多指令流多数据流（MIMD）计算机。它与阵列处理机不同，它有较强的通用性，例如对不同的数组进行不同的处理。但一个具体任务是否能并行处理是与任务本身的性质有关。有因果关系的任务，只能先获得原因才能计算结果。设计的多处理机当然不能只处理完全没有并把约束关系的程序。因此，多处理机需要专门的指令来表示程序中并发关系和控制并发执行，它才能获得正确的结果，同时能充分利用多处理机的并行处理功能。

解答

（51）D　　（52）D　　（53）B　　（54）C　　（55）A

试题（56）~ （60）分析

利用元件冗余提高系统可靠性的根据是冗余的元件在需要时可替代有故障的原来工作的元件。例如可采用双机的方法来达到容错的目的。其中一个完成所需的工作。另一个作备份。若使备份机也一直保持与承担正式工作的机器同步运行，那么此备份机是专做备份用，且一旦工作机出故障，只需将它切入到系统，它便能立即代替原来那个机器。这种备份方式称为“热备份”。这种系统称为双重系统。当备份的那个机器平时停机或者做其他工作，仅在工作的那个机器出故障时，再让它去代替，使整个系统仍为正常，这样的系统称为双工系统。按题中图 A 给出的并联模型，仅当 3 个子系统都失效时整个系统才失效，否则系统能正常

工作。3 个子系统同时失效的概率为（1－0.8）^{3}，所以这种并联后的系统可靠性为 1－（1－0.8）^{3}＝0.992。按题中图 B 的表决模型，任何 2 个子系统正常或 3 个子系统都正常时系统能正常工作。因此系统正常工作的概率为 3 个子系统都正常的概率加上所有 2 个系统正常同时有一个子系统不正常的概率。它可用 $\sum_{i=2}^{3} C_3^i R^i (1-R)^{3-i}$ 来计算，其中 $C_N^i = \frac{3!}{(3-i)!} \bullet i!$，R 是子系统正常的概率。由 R＝0.8，可得系统正常概率为 0.896。当 R＝0.5 时，可算得整个系统的可靠性为 0.5。可以预计在 R＜0.5 时，即单个子系统出错的概率大于正常的概率时，这种采用多数表决的模型其出错概率反而增大。

解答

（56）C　　（57）A　　（58）C　　（59）B　　（60）A

试题（61）～（65）分析

访问万维网（WWW，world wide web）页面是目前因特网上最为广泛的应用之一。用户端只要装有浏览器，如微软公司的 IE（Internet explorer）或网景公司的导航器（navigator），并连接到因特网上就可以访问成千上万的 WWW 页面。所以能在成千上万的 WWW 页面中找到端用户所要访问的那一个特定的页面，依赖于标准化的统一资源定位器 URL（uniform resource locator）地址，如http://www.yahoo.com/directory/file。其中，www.yahoo.com就是雅虎公司主页所在 Web 服务器的域名。从域名找到相应的 IP 地址是由称为域名服务器 DNS（domain name server）的系统来完成的。知道了要访问的 Web 服务器的 IP 地址后，就可以在用户端和该 Web 服务器的间建立一条 TCP 连接。TCP 连接是在对等的服务端口间建立的，不同的服务使用不同的端口。WWW 页面访问时通常采用超文本传送协议 HTTP（hypertext transfer protocol），其服务端口就是 HTTP 服务端口。然后浏览器就可以在该 TCP 连接上依照相关协议向 Web 服务器发送 GET 命令，取得所需的信息。

依上面所述，WWW 页面访问的大致过程为：用户在浏览器中输入要访问的 WWW 页面的 URL 地址；浏览器通过 DNS 查询上述输入信息所指的 Web 服务器的 IP 地址；浏览器通过网络与该 IP 地址处的 Web 服务器的 HTTP 服务端口间建立一条 TCP 连接 ;浏览器依照相关协议发送 GET 命令；Web 服务器根据命令取出文档，发送回来；浏览器释放连接，显示该文档。

解答

（61）A　　（62）D　　（63）B　　（64）C　　（65）B

试题（66）～（70）分析

该文的中文译文如下：

MIMD 系统可以分类为面向吞吐量的系统，高可用性系统和面向响应的系统。面向吞吐量的多处理的目标是以最小计算成本得到高吞吐量。多处理器操作系统为达到此目标而采用的技术利用了工作负载上处理对于输入/输出的固有的平衡的产生系统资源的平衡和均匀的负载。

解答

（66）D　　（67）A　　（68）B　　（69）A　　（70）B

试题（71）～（75）分析

该文的中文译文如下：
不久，通过因特网接收的信息将会更多地以数字包裹形式打包到达。
包裹由软件代码组成。其目的对封装在其中的数据做些特定的事情，例如帮助为搜索引擎定义查询。它们也防止局外人获取对该代码的存取。

解答

（71）B　　（72）A　　（73）D　　（74）C　　（75）D

九 2001 年度下午试题分析与解答

试题 1　分析

本题给出的流程图是描述了某计算机厂发货、收款和催款的处理过程。
在数据处理中，检查输入数据的正确性是必要的，也是很重要的。对输入数据一般可对以下几方面进行检查：
检查数据的类型是否符合要求。
检查数据的数值范围是否合法。
检查数据间的依赖关系是否正确，例如金额等于单价乘数量。
若同一数据通过两个不同的渠道输入，则要检查两次输入的数据是否一致。
在本题给出的处理流程中，处理 1 和处理 4 分别用来输入发货单和收款单，它们均可作上述 1~3 的检查。处理 7 在发货文件中对当天已收款的记录上加上已收款标记，此时收款单和发货单来自不同的渠道，因此处理 7 中可作上述 4 的检查。
在流程图中，处理 7 的主要任务是从发货文件中找到那些已收款的发货单，并对它标上已收款标记。为了提高查找速度，处理 7 中所使用的与发货和收款有关的文件都应该按相同的关键字所排序。若将日收款分类文件到处理 7 的连线改成从日收款文件到处理 7 连线，则由于日收款文件存放的是未经排序的收款单，所以就会降低处理 7 中的查找速度。通常发货文件是存放在外存中的，所以处理时就会增加访问外存的时间。
因为处理 9 每月末执行一次，除利用收款文件产生并输出本月的收款报告外，还得删除收款文件中的所有记录。

解答

问题 1　处理 1 和处理 4 中要检查发货单和收款单的非法销售点代号和非法商品代号的错误；数量和金额非正数的错误；单价乘数量不等于金额的错误；输入中的非法字符。处理 7 要检查收款单与发货单中不一致的错误。

问题 2　因日收款文件未分类，处理时要增加访问外存的时间或需大量内存从而降低了处理速度。

问题 3　删除收款文件中的所有记录。

试题 2　分析

本题给出了一个不完整的分层数据流图，用来描述考务处理系统，要求考生在此基础上，添加相关的 DFD 成分，以实现说明中所指定的考务处理系统应具有的功能。
在分层的数据流图中，父图与子图应该是平衡的，即子图的所有输入数据流必须是父图中相应加工的输入数据流；子图的所有输出数据流必须是父图中相应加工的输出数据流。如果父图中相应加工的一个输入（输出）数据流对应于子图中若于个输入（输出）数据流，且子图中这些数据流的成分之和正好等于父图中的这个数据流，那么仍认为它是平衡的。
在 0 层数据流图中的加工 1———报名处理，被分解成加工 1 子图，在父中有一个输入数据流和两个输出数据流，而在加工 1 子图中只有一个输出数据流，遗漏了报名单输入数据流和准考证输出数据流。根据父图和子图的数据流应该平衡的原则，应将报名单数据流作为加工 1.1———检查报名单加工的输入数据流；应将准考证数据流作为加工 1.2———编制准考证加工的输出数据流。

0 层数据流图中的加工 2———成绩处理，被分解成加工 2 子图，在父图中的所有输入输出数据流中，除成绩分布表输出数据流外，其余数据流在加工 2 子图中都有反映且方向也是一致的，而加工 2 子图中的四个子加工都不可能流出成绩分布表数据，因此应增加一个子加工 2.5，加工名称为“制作成绩分布表”，它的输入数据流来自成绩册文件，输出数据流是成绩分布表。另外，在加工 2 子图中有考生名册和成绩册两个文件，它们与加工 2.1~2.4 都无联系，这是不正确的，遗漏了数据流。从 4 个子加工的功能可知，考生名册文件应是加工 2.1、加工 2.2 和加工 2.3 的流入数据；成绩册文件应是加工 2.2、加工 2.3 和加工 2.4 的流入数据，成绩册文件的生成是由加工 2.1 完成的，所以加工 2.1 的流出数据是到成绩册文件。

此外，在一层数据流图中，如果一个文件仅作用于一个加工，那么该文件可作为局部文件出现在该加工的子图中，而不必出现在它的父图中。

在 0 层数据流图中，“成绩册”文件仅在加工 2 成绩处理中使用，因此它是一个局部文件，可以从 0 层图中删去。

解答

问题 1　成绩册文件可删。

问题 2　报名单数据作为加工 1.1 输入数据流；准考证数据作为加工 1.2 的输出数据流。

问题 3　增加加工 2.5，加工名称为“制作成绩分布表”，它的输入流是成绩册文件，输出流是成绩分布表；增加从考生名册文件到加工 2.1 至 2.3 的数据流；增加从成绩册文件到加工 2.2 至 2.4 的数据流；增加加工 2.1 到成绩册文件的数据流。

试题 3　分析

阅读程序说明可知子程序 DEHZ 是对 HZ 编码的字串（有 ASCII 字符和汉字字符）进行解码处理。HZ 编码是将原高位为 1 的汉字双字节串转换成高位为 0 的 ASCII 字符串，并在汉字字串前后两端分别添加~{和~}，而字串中的原 ASCII 字符~符改变成~ ~，其余 ASCII 字符保持不变。

解码处理是将~{和~}间的 ASCII 字符串复原成高位为 1 的汉字双字节串，将~ ~恢复成~。按照解码要求去分析子程序的结构，理解和推敲子程序中已给出的指令在解码过程中的作用，就能逐步求得解答。

子程序开始处的 3 条指令是置初值，GR1 和 GR2 分别被置成 HZ 编码字串存放首地址和解码后结果字符序列存放首地址；GR3 被置成 0 表示初始化为处理 ASCII 字符。接着的指令是判断 GR0 是否为字符~，若不是则转 GOON 在执行（4）处指令后，将 GR0 送结果区、将 GR1 和 GR2 分别增 1，再判断 GR0 是否为零———HZ 编码字串的结束标志。若是零结束子程序，否则就应该转到取下一个字符作解码处理，所以（5）的解答是 JNZ　LOOP 转去取下一个字符。同时可得（1）的解答是 LD　GR0，0，GR1 取下一个字符。

经过上述分析和求解可推得子程序的 6~ 25 行构成一个循环，完成 HZ 编码字串的解码处理。因标号 GOON 起的指令仅是将 GR0 中的字符送结果区和修改 GR1、GR2 及其判断是否遇结束标志等指令，所以标号 GOON 之前的指令一定是区分 HZ 编码中的各种情况并对它们实现解码工作。

子程序第 7~ 11 行指令是判断 HZ 编码序列中是否连续出现~ ~，若是应将~ 符送结果区，所以（2）处的指令是 JZE　GOON；若出现一个~ 符，就执行指令

CPA　GR0，MARK1，GR3

它是判断是否出现~ {（GR3 为零时）或是判断是否出现~}（GR3 为 1 时）。若是前者，表示进入汉字双字节区域，要把 GR3 变为 1 且要将以后各字节的高位置成 1；若是后者，表示汉字双字节区域结束，要把 GR3 变为零，所以（3）处的指令是改变 GR3 的值，即将 0 变

成 1 或将 1 变成 0，可用指令 EOR　GR3，V1 实现此要求。
在出现~｛时应将其后直至~｝之前的各字符（在 GR0 中）的高位置 1 后才能存入结果区，其他范围内的字符保持不变。因此标号 GOON 处的空格（4）要根据 GR3 的状态，把 GR0 的第八位置 1（GR3 为 1 时）或 GR0 不变（GR3 为 0 进）的处理后，再存入结果区。注意到常数区连续出现的 0 和＃0080 两个常数，可用指令 OR　GR0，V0，GR3 实现上述要求。

解答

LD　GR0,0，GR1
JZE　GOON
EOR　GR3，V1
OR　GR0，V0，GR3
JNZ　LOOP

试题 4　分析

函数 NODE*makeTree（）的功能是参照已知列表生成一棵 M 叉树，其中列表字符串由全局指针变量 str 所指。从函数的结构中看到，除对结点有一个循环控制子树的构造外，没有对结点的子树的构造有特别的构造过程的代码，所以函数是一个构造子树的递归函数。通过递归调用构造子树根结点的子树。按列表的构成规则，首先是子树根结点的值，函数 makeTree（）也得先为子树的根结点向系统申请存储空间，所以框（1）的解答应为（NODE*）malloc(sizeof(NODE))。根结点有了空间后，列表字符串的当前字符就是根结点的值。接着函数为根结点的子结点预置空值。如结点值之后接有字符'('，说明该结点有子树。由于有多棵子树，函数用循环控制对子树的构造。在第一个子树的列表之前有字符左括号，其他子树的列表之前有逗号，所以函数在构造每棵子树前，列表字符指针先前移一个字符位置，接着采用递归调用自己构造子树。所以框（2）的解答是 makeTree()。若子树之后遇字符右括号，则表示结点不再有更多的子树，函数跳出构造子树循环。根据约定，子树之间用逗号分割，若子树的后继字符是逗号，表示还有后继子树，需继续循环。所以框（3）解答是*str==','。
函数 void walkTree(NODE *t)做与函数 makeTree()相反的工作，由已知 M 叉树的根结点指针，输出该树的列表字符列。由函数 walkTree()的结构知，它与函数 makeTree()一样，也是采用递归算法，实现一棵子树的列表输出。对于一棵非空的子树，函数首先输出子树根结点的值，所以框（4）的解答应是 putchar(t->val)。函数接着判定子树是否有子树，若子树的第一个子结点指针为空，说明该结点没有子结点，函数转换结束而返回。否则，先输出左括号，并逐一对该结点的各子树做转换输出工作。对各子树的转换用递归实现，提供的实参是子树根结点的指针，所以框（5）的解答是 walkTree(t->subTree[i])。每棵子树转换输出后，若还有更多的子树，函数输出逗号；若不再有更多的子树，函数输出右括号，结束全部转换输出工作。

解答

（1）（NODE*）malloc(sizeof(NODE))
（2）makeTree()
（3）*str==','
（4）putchar(t->val)
（5）walkTree(t->subTree[i])

试题 5　分析

函数 int back(int *ip,int color[])实现回溯法找解过程中的回溯功能。其中形参 ip 指向记录当前考察区域号的变量。用指针形参是因为回溯时，函数要修正当前区域号。回溯过程是一个循环，若当前区域*ip 已着 4 号颜色，则要回溯。回溯时，若发现当前区域*ip 已是 0 号区域，不可再回溯，函数返回 0 值，表示已穷尽了所有可能方案。否则，当前区域*ip 减 1，取出*ip 区域的着色方案，并置*ip 区域为未着色。*ip 区域的着色方案存于 color[*ip]中，所以框（1）的解答是 color[*ip]。若*ip 区域的着色方案不是最后一种颜色，则回溯结束，函数返回该区域原来的着色方案，让*ip 区域选用编号更大的颜色。

函数 int colorOk(int i, int c , int adj[ ][N] ,int color[ ])的功能是检查区域 i 对 c 号颜色的可用性。区域 i 是否可用 c 号颜色，得逐一考察区域号比 i 小，而与区域 i 相邻的所有区域 j，看区域 j 是否已用 c 号颜色着色。区域 i 与区域 j 相邻可用表达式 adj[i][j]!＝0 表示，区域 j 的着色方案是 color[j]。若某区域 j 与区域 i 相邻，并且区域 j 用 c 号颜色着色，则函数返回 0 值，表示区域 i 不能用 c 号颜色着色。所以框（2）的解答是 adj[i][j]!=0 && color[j]= =c 。若区域号比 i 小，而与区域 i 相邻的所有区域 j 都不用 c 号颜色着色，则函数返回 1，表示区域 i 可用 c 号颜色着色。

函数 int select(int i ,int c ,int adj[ ][N],int color[ ])的功能是为区域 i 选一种可着的颜色。为了避免早先判定为不可着的颜色被重新反复检查，函数设置起始颜色参数 c，表示找到的颜色号至少是 c 号颜色。函数只要对 c 至 4 的 k 号颜色逐一调用函数 colorOk()，若某次函数调用 colorOk()返回非 0 值，说明 k 号颜色是编号最小，区域 i 可着的颜色号。调用 colorOk()函数需提供区域 i、颜色号 k、邻接矩阵 adj、当前各区域的着色方案 color，所以框（3）的解答是“ i,k,adj,color ”。

函数 int coloring(int adj[ ][N])的功能是利用给定的图的邻接矩阵，寻找各种着色方案。函数首先预置各区域都未着色、区域 i 为 0 号、可选颜色号 c 为最初颜色号减 1（＝0）、已找到方案数 cnt 为 0 个。寻找全部着色方案是一个循环，直至不再有可能的方案结束。每次循环首先是为 i 号区域寻找一种着色方案，可选的初始颜色是 c+1 号颜色，邻接矩阵是 adj，选定的颜色存于数组 color。所以框（4）的解答是 select(i,c+1,adj,color)。若函数调用 select(i,c+1,adj,color)返回 0 值，表示不能为 i 号区域选定一种着色方案，函数就调用回溯函数 black（）。回溯函数回溯至区域号更小的某个 i 号区域，并返回该区域的着色颜色号。若回溯至没有可着的颜色，说明已穷尽了所有可能方案，函数返回找到的方案个数。若为 i 号区域寻找到一种着色 c，则应将 c 号颜色存入到 i 号区域的着色方案 color[i]中，所以框(5)的解答是 color[i]＝c。接着函数准备为下一个区域寻找着色方案。若全部区域都设定了着色方案，函数输出找到的解，并累加找到方案的计数器。若还有区域未着色，则下一个区域再次从编号最小的颜色号开始，寻找着色方案。

解答

color[*ip]

adj[i][j]!=0 && color[j]= = c

i ,k , adj ,color

select(i,c+1,adj,color)

color[i]=c

十 2002 年上午试题分析与解答

试题（1）~ （2）分析

美国数据加密标准 DES 是一种对称加密的算法，“对称”是指采用的保密密鈅既可用于加密，也可用于解密，DES 的算法是公开的，密鈅由用户自己保护。密鈅长度为 64 位，其中有 8 位奇偶校验，有效密使长度为 56 位，即采用一个 56 位有效密鈅对 64 位的数据块进行加密。

解答

（1）A　　（2）B

试题（3）~ (5)分析

在面向对象技术中，对象在收到信息后要予以响应，不同的对象收到同一消息可产生完全不同的结果，这一现象称为多态。在使用多态技术时，用户可以发送一个通用的消息，而实现的细节则由接受对象自行决定，这样同一消息就可以调用不同的方法。多态有多种不同的形式，其中参数多态和包含多态称为通用多态，过载多态和强制多态成为特定多态。

参数多态是应用比较广泛，被称为最纯的多态。这是因为同一对象、函授或过程能以一致的形式用于不同的类型。包含多态最常见的例子就是子类型化，即一个类型是另一类型的子类型。过载多态是同一变量被用来表示不同的功能，通过上下文以决定一个名所代表的功能。即通过语法对不同语义的对象使用相同的名，编译能够消除这一模糊。强制多态是通过语义操作把一个变元的类型加以变换，以符合一个函授的要求，如果不做这一强制性变换将出现类型错误。类型的变换可在编译时完成，通常是隐式地进行，当然也可以在动态运行时来做。

解答

（3）A　　（4）D　　（5）C

试题(6) ~ (9) 分析

若已知非确定的有穷自动机 N(NFAN)={S，$\sum$ ，move，s0，F}，则可以构造出相应的确定有穷自动机 M(DFA M)= {S＇，$\sum$ ，move＇，q0，F＇}。其中 S(或 S＇)是一个状态集合；$\sum$ 是一个有穷字母表；move(或 move＇)是状态转移函数，例如“ move(s1,a)=s2”表示当系统处于状态 s1 且输入字母为“ a”时，系统转换到状态 s2；s0(或 q0)是起始状态；F(或 F＇)为接受状态集合。在介绍构造方法之前，我们先给出一个定义。若 I 是 NFAN 的状态集的一个子集，我们定义ε_CLOSURE(I)如下：

状态集 I 的ε_CLOSURE(I)是一个状态集；

状态集 I 的所有状态属于ε_CLOSURE(I)；

若 s∈I，那么从 s 出发经过任意条ε弧到达的状态 S＇都属于ε_CLOSURE(I)。

状态集ε_CLOSURE(I)称为 I 的ε_闭包。

由以上的定义可知，I 的 ε _闭包就是从状态集 I 的状态出发，经 ε 所能到达的状态的全体。

假定 I 是 N 的状态集的一个子集，a 是 $\sum$ 中的一个字符，我们定义：

Ia= ε _CLOSURE(J)

其中，J 是所有那些可以从 I 中的某一状态结点出发经过一条 a 弧而到达的状态结点的全体。

在定义了 ε _CLOSURE(J)后，我们就可以用子集法将一个 NFAN 确定化为一个 DFA，也就是求解出 S＇、move＇、q0 和 F＇。用子集法将非确定的有穷自动机 N(NFA　N)确定化的算法如下：

1．求出 M 的初态 q0,即令 q0= ε _CLOSURE({s0})，此时 S＇仅含初态 q0，并且没有标记。

2．对于 S＇中尚未标记的状态 qi={si1,si2,…,sim},sij∈S(j=1, …,m)：

标记 qi；

对于每个 a∈ $\sum$,令

T=move({si1,si2,…,sim },a), qj= ε _CLOSURE(T)

若 qj 尚不在 S＇中，则将 qj 作为一个未加标记的新状态添加到 S＇，并把状态转移 move＇(qi,a)=qj 添加到 M 中；

3．重复进行步骤 2，直到 S＇中不再有未标记的状态时为止。

4．令 F＇={q|q∈S＇且 q∩F≠Φ}。

从题目给出的 NFA 的状态转换图可以看出，状态结点 S 就是自动机的初始状态，令 S＇的初值为空集合。根据 ε _闭包的定义，可以求出 ε _CLOSURE({S0})={S，1，2，3}，将{S，1，2，3}记为 q0，并将 q0 作为未加标记的状态加入 S＇，因此 S＇={q0}；其后求解 DFAM 的各个状态的过程如下：

　根据题中所给出的状态转换图以及算法中的第 2 步，进行如下计算：

将 q0 作上标记，以“＊”作为标志的记号，即 S＇={q0*}。

求出 q0 对于字母表是 0 和 1 的状态转移。

因为 move(1,0)=1,move(3,0)=4，状态 S 及状态 2 中不存在关于字母“0”的转移函数，所以 move(q0,0)={1,4}，那么根据 ε _闭包的定义可求得 ε _CLOSURE({1,4})={1,3,4,5，Z}，我们将{1,3,4,5，Z}记为 q1；

同时由于 move(2,1)=2,状态 S、状态 1 及状态 3 中不存在关于字母“1”的转移函数，所以 move(q0,1)={2}，因此可求得 ε _CLOSURE({2})={2,3}，我们将{2,3}记为 q2；

将计算所得到的 q1 和 q2 作为未加标记的新状态添加到 S＇，即 S＇={q0*,q1,q2}；依次类推，重复进行以上算法得到的中间结果如下：

ε _CLOSURE(move(q1,0))= ε _CLOSURE({1,4,6})={1,3,4,5,6,Z},将{1,3,4,5,6,Z}标记为 q3,

ε _CLOSURE(move(q1,1))= ε _CLOSURE({})={},{}无须标记，

S＇={q0*,q1*,q2，q3}；

ε _CLOSURE(move(q2,0))= ε _CLOSURE({4})={4,5,Z},将{4,5,Z}记为 q4,

ε _CLOSURE(move(q2,1))= ε _CLOSURE({2})={2,3}=q2,

S＇={q0*,q1*,q2*，q3,q4}；

ε _CLOSURE(move(q3,0))= ε _CLOSURE({1,4,6})={1,3,4,5,6,Z}=q2,

ε _CLOSURE(move(q3,1))= ε _CLOSURE({5})={5,Z}，将{5，Z}记为 q5,

S＇={q0*,q1*,q2*，q3*,q4,q5}；

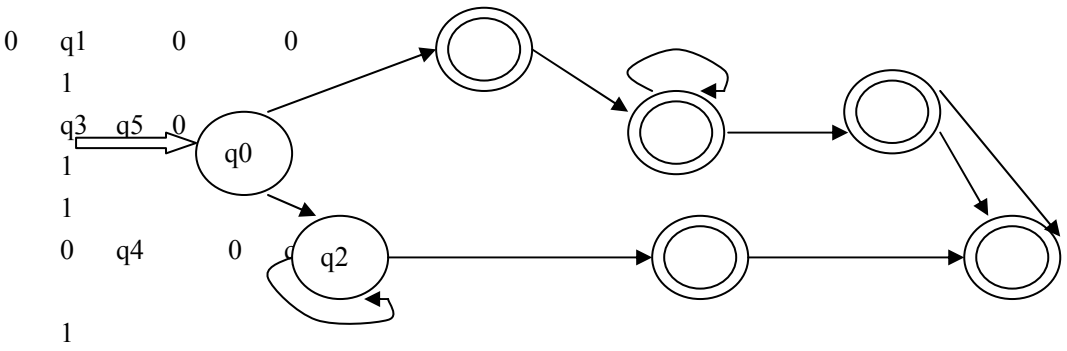
ε _CLOSURE(move(q4,0))= ε _CLOSURE({6})={6}，将{6}记为 q6,
ε _CLOSURE(move(q4,1))= ε _CLOSURE({})={},
S＇={q0*,q1*,q2*，q3*,q4*,q5,q6}；

ε _CLOSURE(move(q5,0))= ε _CLOSURE({6})={6}=q6,
ε _CLOSURE(move(q5,1))= ε _CLOSURE({})={},
S＇={q0*,q1*,q2*，q3*,q4*,q5*,q6}；

ε _CLOSURE(move(q6,0))= ε _CLOSURE({})={},
ε _CLOSURE(move(q6,1))= ε _CLOSURE({5})={5,Z},
S＇={q0*,q1*,q2*，q3*,q4*,q5*,q6*}；
当 S＇中不再有未加标记的状态时，算法即可终止，将计算结果以表格形式表示如下：

I	I0	I1
q0({S,1,2,3})	q1={1,3,4,5,Z}	q2={2,3}
q1({1,3,4,5,Z})	q3={1,3,4,5,6,Z}	{}
q2({2,3})	q4={4,5,Z}	q2={2,3}
q4({4,5,Z})	q6={6}	{}
q3({1,3,4,5,6,Z})	q3={1,3,4,5,6,Z}	q5={5,Z}
q6({6})	{}	q5={5,Z}
q5({5,Z})	q6={6}	{}

对照题目中的表格，可知 T1 即为 q3(={1,3,4,5,6,Z})，T2 即为 q4(={4,5,Z})，T3 即代表空集{}。下面给出以状态转换图表示的 DFA。



解答
(6) A　　(7)D　　（8）D　　（9）D

试题（10）　分析

正规式（a|b）*表示字符 a 和 b 组成的任何长度的字符串，例如空串，a,b,aa,ab,ba, bb,aaa, aba,abb,baa,bab,bba,bbb 等字符串。
正规式 a*|b*表示的是由若干个 a 组成的字符串或者是若干个 b 组成的任何长度的字符串，例如空串，a,b,aa,bb,aaa,bbb 等。
正规式 a*b*表示的是由若干个 a 跟若干个 b 所组成的任何长度的字符串，例如空串，a, b, aa, ab, bb, aaa, aab, abb, bbb 等。
正规式（ab）*表示的是每个 a 后面必跟一个 b,以及除了最后一个 b 外，每个 b 后面必跟一

个 a 所组成的任何长度的字符串，例如空串，ab，abab，ababab 等。
正规式（a*b*）*表示由字符 a 和 b 组成的任何长度的字符串，例如空串，a，b，aa，ab，ba，bb，aaa，aab，aba，abb，baa，bab，bba，bbb 等字符串。
因此，正规式（a|b）*与（a*b*）*是等价的。
解答
（10）C

试题（11）分析

算法是对特定问题求解步骤的一种描述，它是指令的有限序列，其中每一条指令表示一个或多个操作。一个算法具有下列五个重要特性：
有穷性。一个算法必须总是（对任何合法的输入值）在执行有穷步骤之后结束，且每一步都在有穷时间内完成（在这里，有穷时间应是在实际求解过程中合理和可接受的）。
确定性。算法中每一条指令必须有确切的含义，读者理解时不会产生二义性。并且，在任何条件下，算法只有唯一的一条执行路径，即对于相同的输入只能得出相同的输出。
可行性。一个算法是可行的，既说明算法中描述的操作都是可以通过已实现的基本运算执行有限次来实现的。
输入。一个算法有零个或多个的输入，这些输入取自于某个特定的对象的集合。
输出。一算法有一个或多个的输出，这些输出是同输入有着某些特定关系的量。
综上所述，算法中描述的操作都可以通过已实现的基本操作在限定时间内执行有限次来实现的，这句话说明了算法的可行性特点。
解答
（11）C

试题（12）　分析

任何一个可以用计算机求解的问题所需的计算时间都与其规模有关，问题的规模越小，解题所需要的计算时间往往也越少，从而也较容易处理。例如，对于 n 个元素的排序问题，当 n=1 时，不需任何计算。n=2 时，只要作一次比较即可排好序。n=3 时只要作 3 次比较即可，而当 n 较大时，问题就不那么容易处理了。要想直接解决一个规模较大的问题，有时是相当困难的。快速排序的基本思想描述为：在无序集合中任选一个记录作为基准元素，以此元素为基准将当前无序区划分为前后两个较小的子区间，并使前面子区间所有记录的关键字均小于等于基准记录的关键字，后面的子区间中元素反之，而基准记录则位于其在最终有序序列的正确位置上，它无须参与后续的排序。然后再递归地对前后两部分子区间实施快速排序。

分治法的基本思想是：将原问题分解为若干个更小规模但结构与原问题相似的子问题，然后递归地将这些子问题的解组合原问题的解。因此在用递归描述的分治算法的每一层递归上，都有如下 3 个步骤：
分解：将原问题分解为若干个子问题，这一不步骤亦为划分。
求解：递归地解各子问题，如果子问题的规模足够小，则直接求解。
组合：将各子问题的解组合成原问题的解。

动态规划法与分治法类似，也是将待求解的问题分解成若干个子问题，先求解子问题，然后从子问题的解得到原问题的解。与分治法不同的是，适合于用动态规划法求解的问题，

经分解得到的子问题往往不是独立的。若用分治法求解，则由于分解得到的子问题太多，而且需要重复求解相同子问题，使得求解问题需要耗费指数时间。

动态规划法通常用于求解具有最优性质的问题，即问题的最优解包含了其子问题的最优解。通常这类问题有多个解，每个解都对应一个值，我们的目的是找到具有最优值（最大和最小值）的那个解。为了避免重复求解相同的子问题，引入一个记忆表，用来记录所有已经解决的子问题的解，当再次遇到该子问题是查表即可。用动态规划法求解问题的步骤为：
分析最优解的性质，并刻画其结构特征。
递归地定义最优值。
以自底向上的方式或自顶向下的记忆方法（备忘录法）计算出最优值。
根据计算最优值时得到的信息，构造一个最优解。

回溯法采用既带有系统性又带有跳跃性的搜索方法，在包含问题的所有解的解空间树中，按照深度优先的策略，从根结点出发进行搜索。算法搜索至解空间树的任一结点时，总是先判断该结点是否肯定不包含问题的解决。如果肯定不包含，则跳过对以该结点为根的子树的系统的搜索，逐层向其祖先结点回溯。否则，进入该子树，继续按照深度优先的策略进行搜索。回溯法在用来求问题的所有解时，要回溯到根，且根结点的所有子树都已被搜索后才结束。而回溯法在求问题的任一解时，只要搜索到问题的一个解时就可以结束。这种以深度优先的方式系统地搜索问题的解的算法称为回溯法。

分支定界法类似于回溯法，也是一种在问题的解空间树上搜索解的算法。但在一般情况下，分支定界法与回溯法的求解目标不同。回溯法的求解目标是找出解决空间树中满足约束条件的所有解，而分支定界法的求解目标则是找出满足约束条件的一个解，或者是在满足约束条件的解中找出使某一目标函数值到极大或极小的解，即在某种意义下的最优解。分支定界法是以广度优先或以最小耗费优先的方式进行搜索，其基本策略是，在扩展结点处，首先生成该结点所有的儿子节点（分支），然后再从当前的活结点表中选择下一个扩展节点。为了有效地选择下一扩展节点，以加速搜索的进程，在每一个结点处，计算一个函数值（定界），并根据这些已计算出的函数值，从当前的活结点表中选择一个最有利的节点作为扩展结点，使搜索朝着解空间树上有最优解的分支推进，以便尽快找出一个最优解。

在冒泡排序中，一次扫描只能确保最大值的元素移到了正确位置，而剩余待排序序列的长度可能只减少 1。快速排序是对冒泡排序的一种本质的改进。它的基本思想是通过一趟扫描后，使待排序序列的长度能较大幅度的减少。快速排序通过一趟扫描使某个元素移到中间的正确位置，并使它左边序列的元素值都比它小，而它右边序列的元素值都不比它小。称这样的一次扫描为“划分”。每次划分使一个长序列变成两个新的较小的只序列，对这两个小子序列分别作同样的划分，直至新的子序列的长度为 1 时才不再划分。当所有的子序列长度都为 1 时，序列已是排好序的了。这是一种分而治之的排序策略。
可见，快速排序算法采用的设计方法是分治法。

解答

（12）B

试题（13）~（14）分析

先给出前缀码的概念：给定一个序列的集合，若不存在一个序列是另一个序列的前缀，则该序列集合称为前缀码。

在进行信息传输时，首先需要对由字符构成的信息进行编码，因此在确定了信息原文中所有可能的字符后，设计出每一个字符的编码，就可对整个信息原文进行编码，然后进行发送。若希望所传送的信息的编码总长尽可能地地短，就需要根据字符的出现频率分别设计出

长度不等的编码，也就是说为字符设计编码时，令出现频率高的字符编码短一些，而出现频率低的字符编码长一些。当被编码的信息到达接收信息的一端时，需要进行翻译才能得到信息源文。而能够正确进行翻译的前提是字符的编码是前缀码。

哈夫曼（Huffman）算法可用来设计前缀编码，下面我们首先了解用哈夫曼算法构造一棵有 n 个叶子（每个叶子具有一个权值）的二叉树的过程。

1.根据 n 个权值（w1,w2,...,wn）构成 n 棵二叉树的集合 F={T1,T2,...,Tn}，其中每棵二叉树 Ti 中只有一个带权为 wi 的根结点，其左右子树均空。

2.在 F 中选取两棵根结点的权值最小的树作为左右子树构造一颗新的二叉树，且置新的二叉树的根结点的权值为其左、右子树结点的根结点的权值之和。

3.在 F 中删除这两棵树，同时将新得到的二叉树加入 F 中。

4.重复 2 和 3，直到 F 中只含一个棵树时为止。称这棵树为最优二叉树（或哈夫曼树）。

如果约定将每个结点的左分支表示字符‘ 0 ’，右分支表示字符‘ 1 ’，则可以把从根结点到某叶子结点的路径上分支字符组成的字符串作为该叶子结点的编码。

对于所有可能传输的字符，令每个字符对应一个叶结点，权值为其出现的频率，那么根据哈夫曼算法构造出二叉树后，就得到了每个字符的二进制编码。

根据构造过程可知，这种编码方案得到的字符的编码长度的数学期望值为最小，因此这种编码方案是一个最优前缀码。在构造过程中，每次都是选取两棵最小权值的二叉树进行合并，因此作出的是贪心选择。

解答

（13）B　　（14）A

试题（15）　　分析

设由 n 个元素构成一个有序表存放在数组元素 A[1]~ A[n]中，查找表中是否存在和 k 相同的元素。如果存在，算法返回相应数组元素的下标；否则，返回值 0。

在由数组元素构成的有序表 A[low]~ A[high]中，二分查找元素 k 的基本方法是：设 mid=[(low+high)/2],那么 A[mid]就是中间位置上的元素。若 k 等于 A[mid],那么就完成查找；若 k 小于 A[mid],那么 k 只可能在 A[low]~A[mid-1]这个区间（前半区），因此只在前半区间查找即可；若 k 大于 A[mid],那么 k 只可能在 A[mid+1]~A[high]这个区间（后半区），因此只在后半区间查找即可。查找过程结束时，或者在某个区间找到元素 k（成功查找）；或者没有找到元素 k（查找失败），即所确定的区间中没有元素（low>high）。

下面的算法中 k 的类型为 Elem Type,数组类型为 Stable。第一次调用该算法时 low 的值为 1，high 的值为 n。

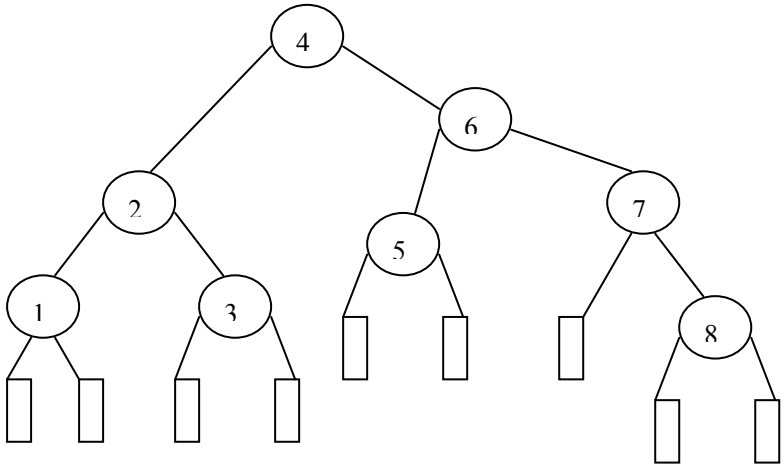
递归实现的二分查找算法如下：

```
int Binary_search(Stable A, ElemType k, int low , int high ){
// 在 A[low] ~ A[high]所构成的子表中查找元素 k
if   (low <= high)
{
mid=[(low+high)/2];
if (A[mid]= =k)   return mid ;
else if (low<high)
if (A[mid]>k)   return Binary_search(A,k,low,mid-1);//继续在前半区进行查找
else   return   Binary_search (A,k,mid+1,high);//继续在后半区进行查找
```

```
else    return 0;
}}
```

由于需要栈结构实现递归调用，因此栈的容量要保证能存放失败查找时所有未完成运行的算法的活动记录。第一次调用该算法时，栈中加入了一条活动记录，表示待查有序表中元素的个数为 n；第二次调用该算法时，无论是在前半区或后半区进行查找，栈中又加入了一条活动记录，所确定的查找区间中的元素最多为$\lfloor n/2 \rfloor$；第三次调用该算法时，栈中又加入了一条活动记录，所确定的查找区间中的元素最多为$\lfloor (n/2)/2 \rfloor$；依次类推，当所确定的查找区间中的元素为 0 时，递归调用该算法的次数为$\lceil \log_2(n+1) \rceil$次，结束查找。

二分查找的过程也可以用一颗二叉判定树描述，下图说明了有 8 个元素的有序表二分查找过程，并设 mid=$\lfloor (low+high)/2 \rfloor$。树中的矩形框表示所有可能的失败查找，也就是所确定的查找区间中没有元素的情况，树的高度即为查找失败时进行递归调用次数最多的情况（结点中的数字为数组元素的下标）。

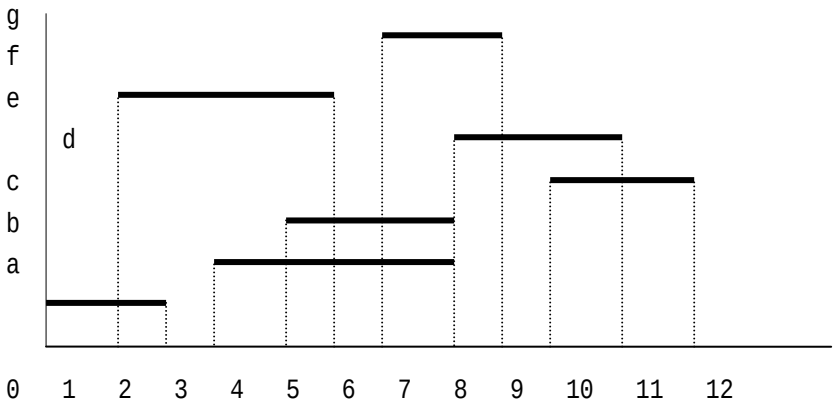


解答

（15）　D

试题（16）　分析

将各个任务执行时的时间关系在下图中表示出来，可容易地观察到 4~5 和 6~7 这两个时间段内，同时有 3 个任务处在执行过程中，其他时间段内同时运行的任务数或是 1 个，或是 2 个。因此，需要并行运行的机器数目最多为 3 个。



解答

（16）　B

试题（17）　分析

渐进时间是考虑当问题规模 n 趋于无穷时函数随时间变化的趋势。所以在比较两个函数的渐进时间应考虑最高数量级，若相同则必须进一步考虑渐进表达式中的常数因子，依次类推。

解答

（17）　A

试题（18）　~　(22）分析

希尔排序的基本思想：先将整个待排记录序列分割成若干个子序列，然后分别进行直接插入排序，待整个序列中的记录基本有序时，再对全体记录进行一次直接插入排序。

快速排序的基本思想：通过一趟排序将待排的记录分割成独立的两部分，其中一部分记录的关键字均比另一部分记录的关键字小，则可分别对这两部分继续进行排序，以达到整个序列有序。

基数排序的基本思想：设有 r 个队列，队列的编号分别为 0,1,2,...,r-1。首先按最低有效位的值把 n 个关键字分配到这 r 个队列中，然后从小到大将各队列中关键字再依次收集起来；接着再按次低有效位的值把刚刚收集起来的关键字分配到 r 个队列中。重复上述收集过程，直至最高有效位，这样便得到一个从小到大的有序序列。为减少记录移动的次数，队列可以采用链式存储分配，称为链式基数排序。每个队列设有两个指针，分别指向队头和队尾。

二路归并排序的基本思想：将两个有序文件合并成一个新的有序文件。它是把一个有 n 个记录的无序文件看成是由 n 个长度为 1 的有序子文件组成的文件，然后进行两两归并，得到$\lceil n/2 \rceil$个长度为 2 或 1 的有序文件，再两两归并，如此重复，直至最后形成包含 n 个记录的有序文件为止。

堆排序的基本思想：对一组待排序记录的关键字，首先把它们按堆的定义排成一个序列（即建立初始小（或大）顶堆），输出堆顶最小（或大）元素，然后将剩余的关键字再调整成新堆，便得到次小（或大）的关键字。如此重复进行，直至全部关键字排成有序序列为止。

解答

（18）C　　（19）B　　（20）D　　（21）B　　（22）C

试题（23）~ (24)分析

在数据流图中，如果有两个以上数据流指向一个加工，或者从一个加工引出两个以上的数据流，这些数据流之间往往存在一定的关系。其中若有符号“ ＊ ”表示相邻的一对数据流同时出现，“ ⊕ ”则表示相邻两个数据流只取其一。

解答

（23）A　　（24）C

试题（25）分析

概要设计是软件开发过程中关键的一步，因为软件系统的质量及一些整体特性基本上是在这一步决定的。概要设计的主要工作是完成模块分解，确定系统的模块层次结构。因此基本任务是将系统划分成模块，决定每个模块的功能、决定模块的调用关系及决定模块界面。

解答

（25）D

试题（26）~（27）分析

若一个软件是给许多客户使用的，那么让每一位用户都进行正式的接受测试是不切实际的。大多数厂商使用一个被称作 alpha 的测试和 beta 测试的过程来发现那些似乎只有最终用户才能发现的错误。alpha 测试是由一个用户在开发者的场所进行的，软件在开发者对用户“指导”下进行测试，开发者负责记录错误和使用中出现的问题，alpha 测试是在一个受控的环境中进行的。beta 测试由软件的最终用户在一个或多个用户场所来进行的，开发者通常不在现场，因此 beta 测试的测试环境是不受控的。

解答

（26）B　　（27）A

试题（28）　分析

随着计算机应用需求的提高，系统软件和应用软件都有很大的发展。但由于软件生成的复杂性和高成本，大型软件的生产出现了很大的困难，即所谓的软件危机。软件危机主要表现在：软件需求的增长得不到满足，软件生产成本高、价格昂贵，软件生产进度无法控制，软件需求定义不够准确，软件质量不易保证，软件可维护性差。归纳起来，软件危机的出现使得人们去寻找产生软件危机的内在原因，发现其原因归结为两个重要方面；一方面是由于软件生产本身存在着复杂性；另一方面是软件开发所使用的方法和技术有关。

解答

（28）D

试题（29）~（31）分析

原型化方法实际上是一种快速确定需求的策略，对用户的需求进行提取、求精，快速建立最终系统工作是模型的方法。原型化方法与结构化方法不同，它不是追求也不可能要求对需求的严格定义、较长的开发时间和熟练的工作人员，但是该方法要求完整的生命周期。为了加快模型的建立，它需要加强用户的参与和决策，以求尽快地将需求确定下来，采用这样一个（与最终系统相比）相对简化的模型就可以简化项目的管理。

原型化是一种动态设计过程，衡量原型化人员能力的重要标准是他能快速获得需求的能力，至于是否有熟练的程序编制调试能力、很强的协调组织能力以及灵活使用工具软件的能力都不是最重要的。

解答

（29）A　　（30）B　　（31）D

试题（32）~（34）分析

事实表明，在无规则和混乱的管理条件下，先进的技术和工具并不能发挥应有的作用。人们认识到，改进软件过程的管理是解决上述难题的突破口，不能忽视软件过程的影响。但是各个软件机构的过程成熟度有着较大的差别，为了作出客观、公正的比较，需要建立一种衡量的标准。使用这个标准一方面可以评价软件承包机构的质量保证能力，在软件项目评标活动中，选择中标机构；另一方面该标准也必然成为软件机构改进软件质量，加强质量管理，以及提高软件产品质量的依据。1987 年美国卡内基——梅隆大学软件工程研究所受国防部资助，提出了软件机构的能力成熟度模型。该模型将软件由低到高分为 5 个级别：初始级、可重复级，已定义级、已管理级和优化级。

解答

（32）A　　（33）B　　（34）C

试题（35）~(37)　分析

本题考查的是关系代数运算和元组演算等价性方面的有关知识。

关系代数表达式 $\pi_{Sno,Sname,Grade}(\sigma_{Cname='数据库'}(S \triangleright\triangleleft SC \triangleright\triangleleft C))$ 的含义为选取同时满足 S.Sno=SC.Sno 且 SCCno=Ccno 且 Cname=”数据库“ 条件的 Sno、Sname 和 Grade。因为 S

1．试题（35）的正确结果为 C。因为试题的关系代数表达式涉及了 3 个关系 S、SC、C，为了转换成等价的元组演算表达式，需要设置 3 个元组变量 u、v、w，而且这 3 个元组变量只要用存在量词“ ∃ ”限定即可。（∃u）S(u)表示在 S 关系中存在一个元组，（∃v）SC(v)表示在 SC 关系中存中一个元组，（∃w）C(w)表示在 C 关系中存在一个元组，这 3 个元组变量应满足 S.Sno＝SC.Sno 且 SC.Cno=C.Cno 且 Cname＝“ 数据库 ”的条件。

2．试题（36）的正确结果为 C。因为 u[1]对应的是 S.Sno，v[1]对应的是 SC.Sno，v[2]对应的是 SC.Cno，w[1]对应的是 C.Cno，w[2]对应的是 C.Cname，所以 S.Sno＝SC.Sno 且 SC.Cno=C.Cno 且 Cname＝“ 数据库 ”等价于 u[1]=v[1]∧v[2]=w[1] ∧w[2]=’数据库’。

3．试题（37）的正确结果 A。因为本题的结果集为 Sno、Sname 和 Grade,而 u[1]对应的是 S.Sno，u[2]对应的是 S.Sname,v[3]对应的是 SC. Grade，所以对属性列 Sno、Sname 和 Grade 的投影等价于 t[1]=u[1] ∧t[2]=u[2] ∧t[3]=v[3]。

解答

（35）C　　（36）C　　（37）A

试题（38）~(39) 分析

本题考查的是关系数据库理论、关系代数运算和关系数据标准语言 SQL 方面的知识。

1．试题（38）的正确结果为 B。因为 R 上的一个函数依赖集为 F＝ {H→J，J→K，I→J，KL→J，JL→H}，对于分解 ρ＝ {HIL，IKL，IJL} 可以构造一个二维表如下：

属性 / 模式	H	I	J	K	L
HIL	a1	a2	b13	b14	a5
IKL	b21	a2	b23	a4	a5
IJL	b31	a2	a3	b34	a5

对于函数依赖集中的 H→J、J→K 对上表进行处理，由于属性列 H 和属性列 J 上无相同的元素，所以无法修改。但对于 I→J 在属性列 I 上对应的 1,2，3 行上全为 a2 元素，所以，将属性列 J 的第一行 b13 和第二行 b23 改为 a3。修改后的表如下：

属性	H	I	J	K	L

模式					
HIL	a1	a2	a3	b14	a5
IKL	b21	a2	a3	a4	a5
IJL	b31	a2	a3	b34	a5

对于函数依赖集中的 JL→H 在属性列 J 和 L 上对应的 1,2，3 行上为 a3 ，a5 元素，所以，将属性列 H 的第二行 b21 和第三行 b31 改为 a1。修改后的表如下：

属性 模式	H	I	J	K	L
HIL	a1	a2	a3	b14	a5
IKL	a1	a2	a3	a4	a5
IJL	a1	a2	a3	b34	a5

从上表可以看到第二行全为 a1a2a3a4a5，所以 ρ 是无损的。

2．试题（39）的正确结果为 A。因为关系代数表达式 $\sigma_{5<'2'}(R)$ 的含义为选取关系 R 中第 5 个属性 E 的值小于 2 的元组，所以等价的 SQL 语句为 SELECT　*　FROM　R WHERE　E＜＇2＇。

解答

（38）B　　（39）A

试题（40）~ （41）分析

本题考查的是关系代数运算方面的知识、

1．试题（40）的正确结果为 A。因为关系代数的除法运算是同时从关系水平方向和垂直方向进行运算的。若给定关系 R（X，Y）和 S（Y，Z），X、Y、Z 为属性组，R/S 应当满足元组在 X 上的分量值 x 的象集 Yx 包含 S 在 Y 上投影的集合。记作：

$$R \div S = \{t_r \mid t_r \in R \wedge t_s[Y] \subseteq Y_x\}$$

其中：Yx 为 x 在 R 中的象集，$x = t_r[X]$。且 R/S 的结果集的属性组为 X。

根据除法定义，此题的 X 有属性为 A3，Y 有属性组（A1，A2），$R \div S$ 应当满足元组在 X 上的分量值 x 的象集 Yx 包含 S 在 Y 上投影的集合，所以结果集的属性为 A3。属性 A3 可以取 3 个值 {c,d,g} ,其中 c 的象集为 {(a,b)} ,d 的象集为 {(b,a) ,(c,d)} ,g 的象集为 {(d,f)}。

实际上此题关系 S 为 $\pi_{A1,A2}(\sigma_{1<3}(S))$，在属性组 Y（A1，A2）上的投影为 {(b,a) ,(c,d)}

如下表所示：

$$\pi_{A1,A2}(\sigma_{1<3}(S))$$

A1	A2
b	a
c	d

从上分析可以看出，只有关系 R 的属性 d 的象集包含了关系 S 在属性组 Y（即 A1、A2）上的投影，所以，$R \div S$ = {d}。

2．试题(41)正确结果为 C。两个关系 R 和 S 进行自然连接时，选择两个关系 R 和 S 公共属

性上相等的元组，去掉重复的属性列构成新关系。这样，关系 R 中的某些元组有可能在关系 S 中不存在公共属性值上相等的元组，造成关系 R 中这些元组的值在运算时舍弃了；同样关系 S 中的某些元组也可能舍弃。为此，扩充了关系运算左外连接、右外连接和完全外连接。

左外连接是指 R 和 S 进行自然连接时，只把 R 中舍弃的元组放到新关系中。

右外连接是指 R 和 S 进行自然连接时，只把 S 中舍弃的元组放到新关系中。

完全外连接是指 R 和 S 进行自然连接时，只把 R 和 S 中舍弃的元组都放到新关系中。

试题（41）R 与 S 的左外连接、右外连接和完全外连接的结果如下表所示：

R 与 S 的左外连接

A1	A2	A3	A4
a	b	c	null
b	a	d	h
c	d	d	d
d	f	g	null

R 与 S 的右外连接

A1	A2	A3	A4
a	z	null	a
b	a	d	h
c	d	d	d
d	s	null	c

R 与 S 的完全外连接

A1	A2	A3	A4
a	b	c	null
b	a	d	h
c	d	d	d
d	f	g	null
a	z	null	a
d	s	null	c

从运算的结果可以看出 R 与 S 的左外连接、右外连接和完全外连接的元组个数分别为 4，4，6。

解答

（40）A　　（41）C

试题（42）~(46) 分析

本试题考查的是操作系统装入过程、SPOOLING 系统、处理机管理方面的基本知识。

1.操作系统装入过程

操作系统是计算机系统的管理者和控制者，用来管理计算机系统中的各种软硬件资源。存储器是计算机系统的硬件资源，是微机的重要组成部分。按其用途可分为主存储器（main memory，简称主存或内存）和辅助存储器（auxiliary memory，简称辅存或外存），但 CPU（中央处理机）只能直接从内存中取指令和数据。要想管理和控制计算机系统的正常工作，操作系统应当放在内存中。

内存储器按存储信息的功能可分为随机存储器 RAM（random access memory）和只读存储器

ROM(read only memory)。ROM 中的信息只能被读出，而不能被操作者修改或删除，故一般用来存放固定的程序，如微机的引导程序、管理程序、监控程序、汇编程序，以及存放各种字库表格等。RAM 就是我们常说的内存或叫做随机存储器，它主要用来存放各种程序、现场的输入、输出数据、中间计算结果，以及与外存交换信息和作堆栈作用。它的存储单元的内容按需要既可以读出，也可以写入或改写。由于 RAM 由电子器件组成，所以只能暂时存放正在运行的数据和程序，一旦关闭电源或掉电，其中的数据就会消失。

外存通常是磁性介质或光盘，能长期保存信息，不依赖于电来保存信息，所以在不通电的情况下仍能保持信息不丢失。不过，任何保存在辅存中的信息只有先装入内存才能为计算机的 CPU 所识别和使用。ＰＣ机上的操作系统是放在辅存上的（因为辅存可以长期保存数据），它是计算机系统的管理者和控制者，因此需要将操作系统装入内存。

BIOS 引导程序是固化在 ROM 芯片上的，每当开机时自动执行 BIOS 引导程序。它主要执行如下任务：

标识和配置所有的即插即用设备：如果系统有即插即用设备，系统将搜索和测试所有安装的即插即用设备，并为他们分配 DMA 通道、IRQ 及它们需要的其他设备。

完成加电自检：加电自检到主要检测和测试内存、端口、键盘、视频适配器、磁盘驱动器等基本设备。有一些新版本的系统还支持 CD－ROM 驱动器。

对引导驱动器可引导分区定位：在 CMOS 中，用户可以设置系统中的引导顺序，以便对引导驱动器的可引导分区重新定位。大多数系统的引导顺序是软驱，然后是硬驱，其次是 CD－ROM 驱动器。

加载主引导记录及引导驱动器的分区表，执行主引导记录 MBR。

主引导记录在硬盘上找到可引导分区后，将其分区引导记录装入内存，并将控制权交给分区引导记录。由分区引导记录定位根目录，然后装入操作系统。

通过以上分析，试题（42）的正确答案是 C，试题（43）的正确答案是 A。

2．Soopling 系统

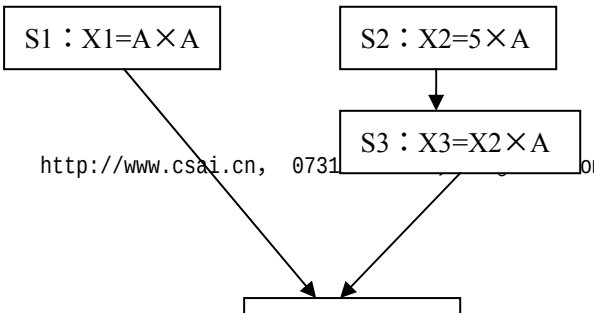
脱机的输入、输出技术是为了缓和 CPU 高速性与 I/O 设备的低速性间的矛盾而引入的。该技术利用了专门的外围控制机将低速 I/O 设备上的数据传送到高速设备上；或者相反。但是当引入多道程序后，完全可以利用其中的一道程序来模拟脱机输入时的外围控制机的功能，把低速的 I/O 设备上的数据传送到高速磁盘上；再利用另一道程序来模拟脱机输出时的外围控制机的功能，把高速磁盘上的数据传送到低速的 I/O 设备上。这样便可以在主机的控制下实现脱机输入、输出的功能。此时的外围操作与 CPU 对数据的处理同时进行，我们将这种在联机情况下实现的同时外围操作称为 Spooling（simultaneous peripheral operation on line ）,或称为假脱机操作。对于试题（44）正确答案是 B。

3．处理机管理

前驱图是一个有向无循环图，图中的每一个节点都可以表示一条语句、一个程序段或一个进程，结点的有向边表示两个节点间存在的偏序或前驱关系“→”。其中：

→＝｛（Pi,Pj）|Pi 必须在 Pj 开始执行之前完成｝如果（Pi,Pj）∈→,可以记为 Pi→Pj,表示直接前驱关系，而 Pi→Pj→…→Pk 表示之间接前驱关系。

对于求值公式 $A^2/(5A+B)$，且 A、B 已赋值，该公式求值过程可用前驱图可表示。下面我们采用自底向上分析。



分析 1：S4 必须等待 S1 和 S3 都执行完成才能执行，因此需要两个信号　量且初值＝0，一个信号量是当 S1 完成时执行 V 操作，发一个消息通知 S4 另一个信号量是当 S3 完成时执行 V 操作，发一个消息通知 S4。
分析 2：S3 必须等待 S2 执行完才能执行，因此需要一个信号量且初值＝0，该信号量是当 S2 完成时执行 V 操作，发一个消息通知 S3 可以执行。
综上所述，本题共需要 3 个信号量，且信号量的初值都为 0。

解答

（42）C　　（43）A　　(44)B　　(45)A　　（46）D

试题（47）~（48）分析

多媒体数据的压缩方法有很多。根据数据所包含信息的有无损失可分为有损编码和无损编码。

有损压缩是指使用压缩的数据进行重构，重构后的数据与原来的数据有所不同，但不影响正确反映对原始资料表达的信息。常用的有损压缩编码技术有：子带编码、模型编码和矢量量化编码等。

无损压缩是指使用压缩的数据进行重构，重构后的数据与原来的数据完全相同。Huffman 编码可以完全恢复原始数据。

试题（47）、（48）考查的是多媒体有关图像数据压缩方面的基础知识。

解答

（47）A　　(48)C

试题（49）~（50）分析

颜色深度是表达位图图像中单个像素颜色或灰度所占的位数，8 位的颜色深度，表示每个像素有 8 位颜色位，可表示 256 种不同的颜色。存储位图图像的数据量与图像大小有关。而位图图像的大小与分辨率、颜色深度有关。本题图像的垂直方向分辨率为 640 像素，水平方向分辨率为 480，颜色深度为 8 位，则该图像所需存储空间为（640×480×8）/8（字节）＝307200。

试题（49）、（50）考查的是多媒体有关图像数据量估算方面的基础知识。

解答

（49）C　　（50）D

试题（51）~（52）分析

声音的三要素为音调、音强和音色。音调与声音的频率有关，频率高则声音高，频率低则给人的感觉声音低沉。音强是对声音强度的衡量，取决于声音的幅度（振幅大小）。音色是由

混入基音的泛音决定，每个基音又都有其固有的频率和不同音强的泛音，从而使得每个声音具有特殊的音色效果。

声音进入计算机的第一步就是数字化，实现声音数字化涉及采样和量化。采样是指按一定时间间隔采集声音样本，每秒钟采集多少个声音样本，通常用采样频率表示。量化是指将声音样本的幅度划分为有限个幅度值，反映度量声音样本的大小，通常用二进制数字表示，称为量化位数或采样深度。声音能按波形声音采样、存储和再现，如果不经过压缩，声音数字化后每秒所需的存储量可由下式估算：

（采样频率×采样深度×声道数）/8(字节/秒)

10 分钟声音、双声道、8 位采样深度、2205kHZ 采样频率，存储量为 441000 字节。

试题（51）、（52）考查的是多媒体有关声音、声音数据量估算方面的基础知识。

解答

（51）B　　（52）A

试题（53）~（55） 分析

cache 的出现是基于两种因素：首先，是由于 CPU 的速度和性能提高很快而主存速度较低且价格高，第二就是程序执行的局部性特点。因此，才将速度比较快而容量有限的 SRAM 构成 cache，目的在于尽可能发挥 CPU 的高速度。很显然，要尽可能发挥 CPU 的高速度就必须用硬件实现其全部功能。

cache 与主存之间可采取多种地址映射方式，直接映射方式是其中的一种。在这种映射方式下，主存中的每一页只能复制到某一固定的 cache 页中。由于 cache 块（页）的大小为 16B，而 cache 容量为 16KB。因此，此 cache 可分为 1024 页。可以看到，cache 的页内地址只需 4 位即可表示；而 cache 的页号需用 10 位二进制数来表示；在映射时，是将主存地址直接复制，现主存地址为 1234E8F8（十六进制），则最低 4 位为 cache 的页内地址，即 1000，中间 10 位为 cache 的页号，即 1010001111。Cache 的容量为 16KB 决定用这 14 位编码即可表示。题中所需求的 cache 的地址为 10100011111000。

Cache 中的内容随命中率的降低需要经常替换新的内容。替换算法有多种，例如，先入后出（FILO）算法、随机替换（RAND）算法、先入先出（FIFO）算法、近期最少使用（LRU）算法等。这些替换算法各有优缺点，就以命中率而言，近期最少使用（LRU）算法的命中率提高。

解答

（53）B　　（54）C　　（55）D

试题（56） 分析

该问题可以直接计算，有 98%的取指令操作只需 10ns，只有 20%的取指令操作需要 100ns。而取指令操作数时 95%只需 10ns，只有 5%的存/取操作数需要 100ns，并且只有 20%的指令需要存/取一个操作数。为此，列出设置 cache 后，每条指令的平均访存时间的计算公式如下：

100×0.02＋10×0.98＋0.2×0.05×100＋0.2×0.95×10＝14.7ns

解答

（56）B

试题（57）分析

在相联处理机中，处理机是以相联存储器为核心，配上必要的中央处理部件、指令存储

器、控制器以及输入输出部件构成。在这种相联处理机中存储器除了存储信息外，还要求它具备一定的信息处理能力。而相联处理机不是按地址顺序对存储器中的信息进行访问，而是根据相联存储器中所存信息的全部内容或部分内容并行地对存储器进行多字访问。显然，这是按内容访问。

解答

（57）C

试题（58）~（60） 分析

磁盘容量有两个指标，即非格式化容量和格式化容量。它们可以分别计算如下：

非格式化容量=最大位密度×最内圈周长×总磁道数

在这里位密度为 250 位/mm；内圈周长为 3.1416×100＝314.16mm；

每记录面总磁道数=(150-50) ×8=800 道，则每记录面的非格式化容量为：

314.16×250×800＝60M

该磁盘由 4 个记录面，则其非格式化容量为：

60Mb×4＝240Mb＝30MB

格式化容量计算公式如下：

格式化容量＝每道扇区数×扇区容量×总磁道数

＝16×512×800×4÷1024÷1024＝25（MB）

硬盘的平均数据传输率可由下式计算：

平均数据传输率＝每道扇区数×扇区容量×盘片转速

根据题中所给出的已知条件和上述计算式，算出：

平均数据传输率＝16×512×7200÷60＝960KB/s

该传输率最接近可选的 1178KB/s。

在题目所给出的条件中可以看到，该硬盘是由多个盘片构成的盘组。该盘组有 4 个记录面构成。同号磁道构成一系列的柱面。因此，在向磁盘记录一个文件时，应将文件尽可能记录在同一柱面上，当一个柱面记录不下时，再记录到相邻柱面上。因此，当一个文件超出一个磁道容量时，剩下的部分应存于其他盘面的同一编号的磁道上，即一柱面的其他磁道上。

解答

（58）B　　（59）D　　（60）B

试题（61）~(65) 分析

本试题考查的是网络安全方面的技术知识。

1．试题(61)、(62)的分析:

数字签名是一种网络安全技术，可以有多种实现方法。与人们手写签名的作用一样，数字签名系统向通讯双方提供服务，使得 A 向 B 发送签名的消息，以便

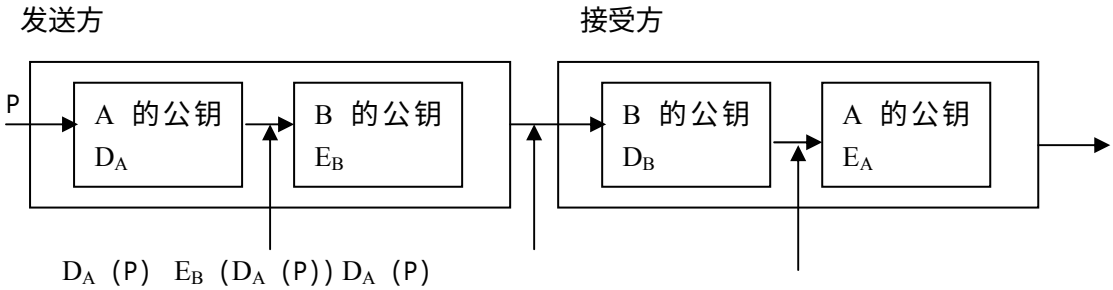
B 可以验证消息 P 确实来源于 A。

A 不能否认发送过 P。

B 不能篡改消息 P。

下图是利用公钥加密钥算法实现的数字签名系统。A 发送消息 P 给 B，首先用 A 的私钥加密，然后用 B 的公钥加密,在网上传送的是只有用 B 的私钥才能解密的报文 $E_B(D_A(P))$。B 用自己的私钥解密后得到 $D_A(P)$，这就是证据。然后 B 用 A 的公钥解密得到明文 P，说明消息

确实是 A 发来的。以后如果 A 方否认了，B 可以拿出 $D_A(P)$，并用 A 的公钥 E_A 解密得到 P，从而证明 P 是 A 发送的。如果 B 把消息 P 更改了，当 A 要求 B 出示证据时，B 拿不出 $D_A(P)$。



可见,数字签名是一种网络安全技术，利用这种技术，接收者可以确定发送者身份是否真实，同时发送者不能否认发送的消息,接收者也不能篡改接收的消息。

2．试题(63)、(64)的分析

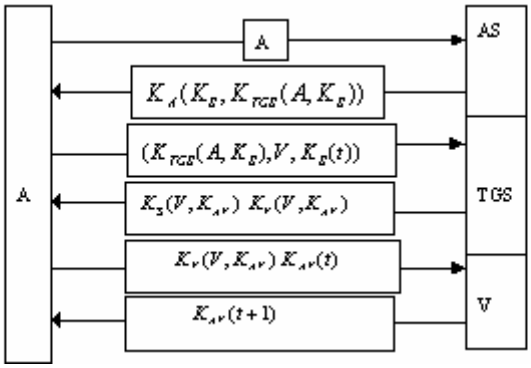
Kerberos 是分布式环境下的身份认证系统，是美国麻省理工学院 Athena 工程设计的安全机制。身份认证有多种方式:静态认证、加强型认证和连续型认证。常用的静态认证方式是通过口令和密码来实现,这种方式的缺点是在网络中传输口令或密码时可能被窃听者捕获并被重新使用 加强型认证利用密码自动生成令牌产生一次性的密钥,可以防止认证信息的重用，但是如果黑客可以看到两个会话之间交换的数据，就可能对这些数据进行修改从而劫持会话，加强型认证不提供这种保护；连续型认证是对会话的所有内容进行加密，或者同时提供消息认证和数字签名。Kerberos 的安全机制如图所示，其中:

AS：认证服务器(Authentication Server)，用于在登录其间验证用户身份。

TGS：发放身份证明的服务器(Ticket-Granting Server)，它发出一次性的 Ticket。

V：用户 A 需要访问的文件服务器(Server)。

A 对服务器 V 的访问要经过图中的 6 个步骤。首先 A 在工作站上发出自己的名字，这个名字以明文形式向认证服务器 AS 传送；然后 AS 向 A 返回消息 $K_A(K_S, K_{TGS}(A, K_S))$，该消息来到时 A 键入口令，并生成一个 Hash 值 K_A，所以 A 可以对其解密，得到的 K_S 是会话密钥，$K_{TGS}(A, K_S)$ 是证明 A 的身份的一次性票证，至此 A 完成了登录过程。第 3 步是 A 向 TGS 发也



消息 $(K_{TGS}(A, K_S), V, K_S(t))$，TGS 在响应报文中指定了会话密钥 K_{AV}，A 可以用 K_S 对

$K_S(V,K_{AV})$ 解密，并把 $K_V(V,K_{AV})$ 发送给服务器 V，现在 A 就可以与服务器用密钥 K_{AV} 会话了。以后如果 A 要与另一个服务器 V＇通信，则在消息 3 中说明 V＇的标识就可以了，即把消息 3 变成 $K_{TGS}(A,K_S)$，V＇，$K_S(t)$，它就会得到另一个密钥 $K_{AV'}$。

可以看出，这种安全机制有下列特点:

K_A 用户 A 的工作站根据用户键入的口令字导出的 Hash 值，最容易受到攻击，但是 K_A 的使用是很少的。

系统的安全是基于对 AS 和 TGS 的绝对信任，实现这两个系统的软件是不允许修改的。

时间戳可以防止重放(Replay)攻击。

2 ~ 6 步使用加密手段，实施了连续认证机制。

AS 存储所有用户的 K_A，以及 TGS 的标识和 KTGS，TGS 要存储 K_{TGS}，服务器要存储 K_V。

K_{AV} 或 $K_{AV'}$ 也是一次性票证。

3．试题(65)分析

PGP 是 pretty good privacy 的缩写，这是一种公钥加密系统，可以提供数字证书或数字标识(digital certificate, digital ID)，用来证明通信双方的真实身份，与日常生活中的身份证相似。在网上进行电子商务活动时，交易双方需要使用数字证书来表明自己的身份，并使用数字证书来进行有关交易操作。通俗地讲，数字证书就是个人或单位在 Internet 上的身份证。

在使用数字证书的过程中，通过运用对称和非对称密码技术建立起一套严密的身份认证系统，能够保证:

信息不会被除发送方和接收方外的其他人窃取。

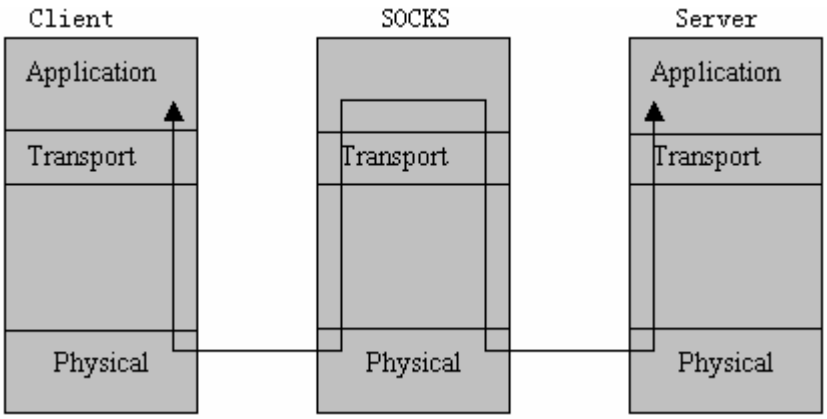
信息在传输过程中不被篡改。

发送方能够通过数字证书来确认接收方的身份。

发送方对于自己发出的信息不能抵赖。

开发的传输层安全协议，用于再造传送机密文件。首先是要建一条安全的年间，然后使用公司加密方法传输数据。常用的浏览器都支持，许多站点利用博取用户的心理信息，例如信用卡好的。使用链接的一套，。

SSL(security socket layer)是 Netscape Communication 开发的传输层安全协议，用于在 Internet 上传送机密文件。SSL 首先要建立一条安全的连接，然后使用公钥加密方法传输数据。常用的浏览器(Netscape Navigator ,Internet Explorer)都支持 SSL，许多 Web 站点利用 SSL 获取用户的机密信息，例如信用卡号等。使用 SSL 连接的 URL，以https://开头。另外一种在互联网上传送机密数据的协议是安全的超文本协议 S-HTTP(security hypertext transfer protocol)。SSK 是在客户机和服务器之间建立一条安全的连接，而 S-HTTP 是安全地传送单个报文，属于应用层协议，因而这两个协议并非竞争的技术，而是互相补充的。SOCKS 是 IETF 的一个正式的标准，用于代理基于 TCP/IP 的网络应用。SOCKS 系统包含两个元素---SOCKS 服务器和 SOCKS 客户机。SOCKS 服务器实现于应用层，而 SOCKS 客户机实现与应用层和传输层之间。这个协议的主要作用是在两个没有直接 IP 联系的主机之间实现通信。当客户机需要访问应用服务器时，客户机首先连接到 SOCKS 代理服务器上，代理服务器再连接到应用服务器上。代理服务器在客户机和应用服务器之间传送数据，如右图所示。对于应用服务器，代理服务器是客户机。



现有两个版本的 SOCKS 协议 SOCKSv4 协议完成在功能：
发出链接请求。
建立代理链路。
传递应用数据。

解答

（61）D　　（62）C　　（63）B　　（64）D　　（65）A

试题（66）~(70) 参考译文

典型的强制型语言都包含一个应用型子语言，它类似于一种数学抽象，把不耗费时间的函数应用于无需空间的值。在表达式求值期间的实际操作序列和使用的存储空间都隐藏在计算现场之后。在这种假定之下，被传递的值只是少量的数据结构，通常只有几个计算机字长，甚至更少。这意味着不耗费空间的设想可以这样来实现：表达式求值过程中产生的中间结果可以根据语言的实现存储在任意位置，结果参数的传递和赋值操作都可以通过值的拷贝来实现。

解答

（66）A　　（67）D　　（68）B　　（69）C　　（70）A

试题（71）~　（75）参考译文

大部分计算机系统都容易受到两种不同类型的攻击：内部攻击和外部攻击。如果一个系统对外部攻击是安全的，它虽然可以阻止外部的访问，但是对于内部的攻击仍然可能是脆弱的。内部攻击来源于授权用户对其访问权限的滥用。检测内部的权限滥用和来自外部的攻击，不仅提供了能评估损害的信息，而且也有助于防止未来的攻击。通常采用工具来检测这类攻击，这种工具成为入侵检测系统。

解答

（71）A　　（72）B　　（73）B　　（74）C　　（75）D

十一 2002 年度下午试题分析与解答

试题 1　分析

本题给出的流程图是长话计费管理的处理流程，用来生成长话交费通知单。

系统的数据源是记录在电信局程控交换机的磁带上的原始计费数据，这些数据在处理之前需要先进行分类，以提高系统的效率。原始计费数据记录的是每次通话的数据，长话交费通知单是针对每个电话用户的，因此在处理 1 中应该按照电话号码进行分类。

F0 是在处理 4（出账）中用来生成长话，账单文件所要用到的“长话业务档案”。由试题的说明可知，月计费文件中含有各种通话类型的话费，所以处理 4（出账）的功能是将长话话费从月计费文件中分离出来，并进行数据的验证。根据以上分析得知 F0 应该是长话业务档案。

F1 是在处理 6 中生成长话交费通知单所要用到的“长话用户档案”。因为用户编码是用户在系统中的唯一标识，所以，应该先将长话账单文件按照用户编码分类，再根据 F1 长话用户档案，得到用户名和用户地址，产生长话交费通知单。因此 F1 应该是长话用户档案。

解答

【问题 1】

（1）F0 是长话业务档案，F1 是长话用户档案。

（2）处理 1：电话号码

处理 5：用户编码

【问题 2】

处理 4 可能发生的错误有：

根据月计费文件中的电话号码，在长话业务档案中找不到相应的用户编码。

在月计费文件中，某电话号码有国内长途通话的话费，但在长话业务档案中，国内长途许可标志却不许可。

在月计费文件中，某电话号码有国际长途通话的话费，但的长话业务档案中，国际长途许可标志却不许可。

【问题 3】

处理 6 的功能是：

对长话账单文件中的每个记录，根据用户编码查询长途电话用户档案，找到相应的用户名和用户地址，形成长话交费通知单。

试题 2 分析

本题描述了一个自动售票系统，并给出了该系统的转换图和状态迁移图。

1．问题 1“转换图中缺少哪 3 条数据流？”分析如下：

转换图是在数据流程图中附加了过程控制的部分，该图描述了自动售票系统的基本行为。根据说明中给出的系统需求描述和转换图，可以看出，该图没有完整地描述系统的基本行为。由于乘客选择的目的地需要经过系统的验证，确定是否是合法的目的地，因此缺少的数据流起点为“接收目的地”，终点为“核查”。

转换图中只给出了将乘客投入的钱全额退还的数据流，并没有给出在其他情况下系统核查和退钱的数据流。因此缺少两条数据流，一条数据流的起点为“接收钱”，终点为“核查”；另一条数据流的起点为“核查”，终点为“退还钱”。

2．问题 2“在状态迁移图中，a，b，c 分别表示什么事件?”分析如下:
为了进一步地描述系统的行为,试题给出了系统的状态迁移图。乘客购票是要经过三步操作:选定目的地，投入钱币，获得一张票。因此 b 应该是“检查正确”,c 应该是“出票结束”。从“正在接收投钱”状态迁移到初始状态“正在等待选择目的地”，应该是乘客取消了购票操作，因此 a 应该是“取消”操作。
3．问题 3“在过程启动表中，d，e 处应填什么?”分析如下:
对于“退钱”这个动作，必须要启动“退还钱”进程，由状态迁移图可知，“退钱”动作结束后，系统会回到状态“正在等待选择目的地”，因此还应该启动“接收目的地”进程，因此 d 应该是 1001。对于动作“接收新目的地”，必须要启动“接收目的地”进程。因此 e 应该是 1000。

解答

【问题 1】
　　转换图中缺少的三条数据流分别是:
数据流名:目的地；起点:“接收目的地”；终点：“核查”。
数据流名：投入的钱；起点：“ 接收钱 ”；终点：“ 核查 ”。
数据流名：剩余的钱；起点：“ 核查 ”；终点：“ 退还钱 ”。
【问题 2】
　　a:“ 取消 ”操作。b:核查正确。c:出票结束。
【问题 3】
　　d:1001。　e:1000。

试题 3 分析

　　工作流是将一组任务组织起来完成某个经营过程。工作流中最基本的元素是活动和活动之间的连接关系,活动对应于经营过程中的任务,主要是反映经营过程中的执行动作或操作；活动之间的连接关系代表了经营过程的规则和业务流程。一个工作流就是用一组连接关系组合起来的一种活动组成的一个反映企业业务过程模型。
组成工作流的基本元素有工作流对象、角色　、路由和规则。
1．问题 1 分析如下:
工作流对象是指在业务流程中流动的文档、表单、事件或消息。因此本试题中的业务流程中的工作流对象是由客户服务代表（CSR）产生的“服务请求文件”。起始点是当 CSR 接到客户问询电话时，产生服务请求文件；终止点是问题解决时，服务请求文件标记为“已解决”。
角色是指在业务流程中产生行为或授受行为的人或机构/部门。本试题的说明中给出的是技术支持部门的业务处理流程，因此在试题求解的过程中报可以不考虑工作部门的角色。所以角色包括 CSR（客户服务代表）、TSR（技术支持代表）、ENGR（助理工程师）。
2．问题 2 分析如下:
CSR－＞TSR。当 CSR 在 2 小时内没有解决客户的问题时，应将服务请求文件转交给技术支持代表（TSR）处理。
TSR－＞ENGR。当 TSR 无法解决客户的问题时，将邀请助理工程师（ENGR）来一起处理这个问题。
CSR，TSR，ENGR－＞归档库。当客户的服务请求已解决且已经到了月底时，放入归档数据库。
3．问题 3 实质上是要说明过程模型与工作流模型的主要区别及应用场合。
一般的过程模型并不要求计算机来执行这个模型，例如描述项目实施计划的甘特图，一般不需要用计算机来执行这个模型；而工作流模型旨在实现业务过程自动化，它描述的是业务逻

辑，必须要用计算机来执行，因此工作流模型不仅能够描述活动及他们之间的相互连接关系，而且需要定义其他许多信息，如角色、数据、路由、资源等，这样才能有计算机进行解释和执行。另外一方面，由于工作流模型需要由计算机来执行，这就对工作流模型的准确性提出了更高的要求，工作流模型的定义也更加严格、准确。

解答

【问题 1】

工作流基本元素	描述
工作流对象	服务请示文件
起始点	CSR 接到客户问询电话，并产生服务请求文件
终止点	问题被解决，服务请求文件标记为“ 已解决 ”
角色	CSR、TSR、ENGR

【问题 2】

路由	规则
CSR－＞TSR	CSR 在 2 小时内没有解决客户的问题
TSR－＞ENGR	TSR 无法解决客户的问题
CSR，TSR，ENGR－＞归档库	客户的服务请求已解决，且已经到了月底

【问题 3】

过程模型和工作流模型的区别在于，过程模型并不要求用计算机来执行，而工作流模型旨在实现业务过程自动化，需要用计算机来执行。

工作流模型目前主要应用在办公自动化领域，用于业务流程的分析、设计以及实现；过程模型广泛地应用于各种应用领域，主要用来进行分析与设计。

试题 4　分析

本题中，源数据放在 SJ 开始的 80 个字中，它们的地址是 SJ+i，其中，i=0,1,2,…,79。目标数据放在 YS 开始的存储区中，地址是 YS＋j，当 i=0,1,2,3 时,j=0；i=4,5,6,7 时，j=1，等等。我们知道到，字符“ 0 ”到“ 9 ”的 ASCII 码分别是十六进制 30H 到 39H，而把它们换算成 BCD 码的方法通常有两种：

减少字符“ 0 ”的 ASCII 码 30H。

与 0FH 进行 AND 操作保留最右 4 位。

仔细阅读试题的程序时发现，从 SJ 取数时（第 5 行）使用 GR2 作为变址寄存器，因此，GR2 的初值应当是 0，并且每处理一个字符 GR2 应当加 1，GR2 的内容应等于 80 时可以作为程序完成的标志。同时我们发现，向 YS 存数时（第 12 行）使用 GR1 作为变址寄存器，因此，GR1 的初值也应当是 0，并且每处理 4 个字符 GR1 应当加 1。注意试题中 GR3 被赋予初值 4（第 4 行），并且每次循环中减 1（第 10 行），而且要根据减 1 结果是否为 O 决定是否跳过一些指令（第 11 行），由此判断 GR3 被用作组计数器，在程序中被用于判断 4 个字符一组的编码过程是否完成。

根据以上分析，我们可以大致得出程序结构如下：

初始化时对 GR2（源数据的变址寄存器）和 GR1（目标数据的变址寄存器）清 0，并给 GR3（组计数器）赋初值 4，然后开始循环。

循环中首先从源地址取出一个字符，转换成 BCD 码后进行拼装，同时 GR3 减 1。当计数 GR3 减 1 后等于 0 时，说明 4 个一组的字符拼装完毕，于是将结果存入目标数据地址，目标数据

变址寄存器加 1 指向下一单元。组计数器 GR3 恢复初始值 4。如果 GR3 减 1 后不等于 0，说明处理的字符还不够一组，因此跳过上述操作。

循环结束时将源数据的变址寄存器 GR2 的内容加 1，指向下一个字符。如果加 1 后 GR2 的内容等于 80，说明全部字符处理完毕，则应当结束循环退出程序。否则跳到循环起始位置开始下一轮的循环。

根据以上分析的程序结构，对照试题我们发现初始化部分缺少给 GR2 赋值的指令。因此，空缺(1)的位置应当是一条给源数据的变址寄存器 GR2 赋初值 0 的指令,也就是“ LEA　GR2,0 ”。同时我们还发现，数据拼装完毕后的处理中（第 12 到 14 行）没有恢复组计数器 GR3 的初值 4，因此第 14 行的空缺（4）必定是给 GR3 赋值 4 的指令，也就是“ LEA　GR3,4 ”。最后，循环结束时对源数据的变址寄存器 GR2 加 1（第 15 行）并与 80 比较（第 16 行）后缺少有条件地跳到循环起始位置的指令，因此第 17 行的空缺（5）应当是一条“ 不等于 0 则跳转到循环起始位置 ”的指令，也就是“ JNZ　S1 ”。注意 S0 是不能当作循环入口的。如果误将 S0 当作了循环入口,那么由于每处理一个字符 GR3 都被赋值 4,结果拼装操作永远不能完成。

现在再来分析字符的转化与拼装过程。我们进一步分析试题，当一个字符源数据存储区取出后放到了通用寄存器 GR0（第 5 行）中，经过第 6 行空缺（2）的指令处理后存入工作单元 WK（第 7 行）。因此，空缺（2）的指令应当将 ASCII 码数据的高 12 位清除，只保留最低 4 位的内容，完成字符的 ASCII 码向 BCD 码的转换。那么，这条指令到底会是减法指令呢，还是 AND 操作用来分离低 4 位所需要的模板 !根据前面的分析，空缺 (2) 中的指令应当是“ AND GR0，CF ”，将 GR0 中的 ASCII 数据与常数 CF（十六进制数 000FH）“ 与 ”操作，使得高 12 位都清 0，而低 4 位不变。最后分析数据的拼装过程。字符由 ASCII 码转换成 BCD 码后存入工作单元 WK 中（第 7 行），最终与通用寄存器 GR4 的内容进行 OR（或操作）（第 9 行）。由此可见,GR4 在这里用作累加寄存器！通过一遍又一遍连续 4 次的或操作，将 4 个字符的 BCD 码拼装成一个字。显然，为了使的先处理的数码放到左边，或操作之前应当将 GR4 的内容左移 4 位。左边移出的数位丢掉，右边补 0，正好放置新的数据！所以第 8 行的空缺（3）应当执行对累加器 GR4 左移 4 位的操作，对应的指令是“ SLL　GR4,4 ”。

根据以上分析得到 5 个空缺处的指令以后，应当完整的再读一遍试题，确定所做的答案是正确的、合理的。

为了便于读者理解，将程序加了相关注释，并且给出了程序的流程图。

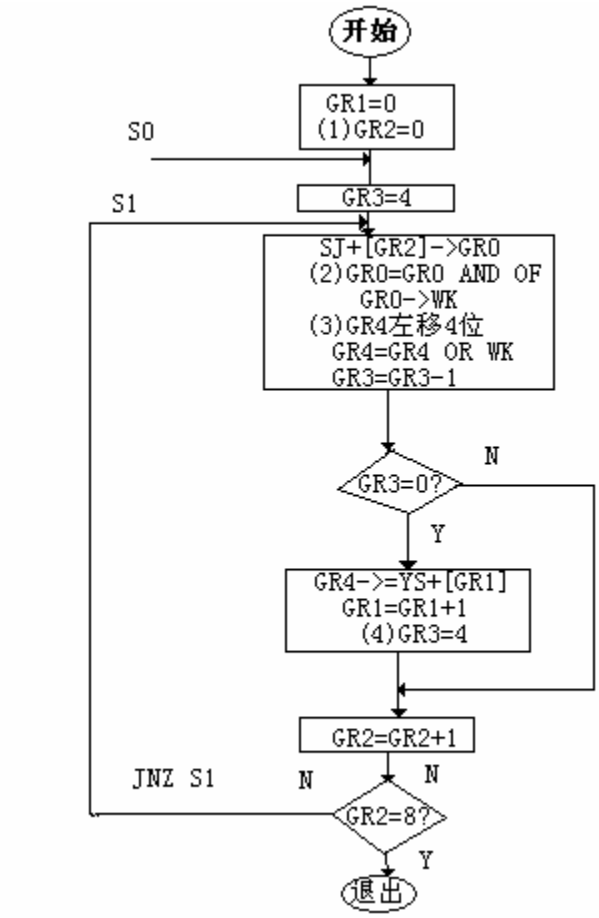
```
1  Y          START
2               LEA  GRI，0              ；0－＞GR1
3               ____（1）____            ；空（1）应填 LEA  GR2,0
4  S0         LEA  GR3,4                ；4－＞GR3。程序中，源数据 4 个单元被处理
；目标数据一个单元中。因此判断 GR3 用于计数。
5  S1         LD  GR0，SJ，GR2           ；SJ＋（GR2）－＞GR0。取被处理的数据到 GR0。
；GR2 中应当包含源数据的偏移。由于此前并未
；有其他指令给 GR2 赋值，故（1）中必定包含
；一条给 GR2 赋值的指令。由于第一次处理应当
；从 SJ 偏移 0 处读出数据，所以（1）包含指令
；应当是 LEA  GR2,0。
6               ____（2）____            ；空（2）应填 AND  GR0，CF。这条指令应当将
；GR0 中的字符数据转换成数字。由于字符 ASCII
；码比数字大 30H，所以屏蔽掉高位是最简单的
；办法。即（2）中包含的指令应当为 AND  GR0，
；CF
```

```
7            ST  GR0，WK              ；GR0 的内容存入工作单元
8            ____（3）____            ；空（3）应填 SLL  GR4,4
9            OR  GR4，WK              ；将 WK 的内容（低 4 位）添加到 GR4 中。由此
；判断，GR4 是累加寄存器，所以（3）的指令
；应当将 GR4 原来的内容左移 4 位，以便腾出空
；间留给新的数据。故（3）包含的指令应当是
；SLL  GR4,4
10           LEA  GR3，－1，GR3       ；(GR3）－1－＞GR3。处理中对源数据组内计数。
11           JNZ  S2                  ；如果一组（4 个）数据处理未完，跳过下列 3
；条指令
12           ST  GR4，YS，GR1          ；如果一组数据处理完毕，结果存入目标地址，
13           LEA  GR1,1，GR1          ；并且 GR1＋1 指向下一单元
14           ____（4）____            ；空（4）应填 LEA  GR3,4。这条指令应当恢复
；GR3 的内容以便下一组数据从头开始，即（4）
；所包含的指令应当是 LEA  GR3,4。
15  S2       LEA  GR2,1，GR2          ；GR2 中的源数据偏移量加 1 指向下一个源数据
16           CPL  GR2，C80            ；判断源数据处理是否已达 80 个。如果处理数
；据已达预定的个数 80 个，应当结束程序。如
；果尚未达到 80 个，应当跳转到 S1 处继续处理
；下一个数据。
17           ____（5）_____           ；通过分析空（5）应当是一个条件转移指令，
；故应当填 JNZ  S1。
18  WL       EXIT                     ；结束程序
19  SJ       DS      80               ；源数据存储区
20  WK       DS      1                ；工作单元
21  CF       DC      ＃000F           ；常数，用于屏蔽除低 4 位外的其他位
22  C80       DC      80              ；常数，待处理字符个数
23  YS       DS      20               ；结果数据的存储区
END
```

解答

```
LEA  GR2,0
AND  GR0，CF
SLL  GR4，4
LEA  GR3,4
JNZ  S1
```

为了验证答案的正确性，可能通过下面的流程图对添加了 5 条指令的程序作进一步分析。



程序分析时要注意的是，程序前部较给出了两个标号 S0 和 S1，因而容易猜想程序可能采用了不同的是循环入口来实现编组。在分析时，由于限制了可填充指令位置和数量，因此并没有多余的选择。答题时应当避免 S0 和 S1 的暗示，而根据流程分析填写正确的答案。

试题 5　分析

程序从 N 件物品中选择若干件，使所选物品恰能放背包中，即所选物品重量之和等于 S。N 件物品的重量保存在数组 w[N+1]中，程序依次选择数组 w 中的物品放入背包，如果该物品能够放入背包（物品的重量小于或等于背包所能盛放的物品的重量），则选择数组中的下一个物品放入背包，否则，将该物品从背包中取出，考察下一个物品。

1．程序 5.1 分析

程序 5.1 用递归的方法实现“背包问题”。每次选择一个物品放入背包，那么剩余的物品和背包剩余的重量，又构成一个“背包问题”。程序中数组中下标最大的物品开始考察，因此第（1）空应填“knap(s-w[n],n-1)”，即将数组中第 N 个物品放入背包，如果它能够放入背包，则该物品是构成解的元素之一；否则，将物品从背包中取出，也就是说，该物品不可能是构成解的元素，在以后的考察中，它可以被排除，因此第（2）个空应填“knap(s,n-1)”。

2．程序 5.2 分析

程序 5.2 是“背包问题”的非递归解法，在程序中使用栈（用数组 stack 表示）来保存到已经考察过的物品。结构 KNAPTP 表示经过考察的物品，s 表示考察过该物品后，背包所能盛放的物品的重量；n 表示该物品在数组 w 中的下标；job 表示物品当前的状态：job 等于 1，表示物品 n 可以放入背包；job 等于 2，表示物品不能放入背包，那么在以后的选取中，将不

再考虑该物品。初始时，job 等于 0，表示背包中没有放入任何物品。K 为有解的标志，k 等于 0 表示没有解，k 等于 1 表示求得了一组解。rep 是一个标志变量，rep 等于 0，表示结束当前的动作；rep 等于 1，表示继续进行当前的动作。当栈顶物品不能装入背包时，将 rep 置为 0，表示下一步不再从数组 w 中取物品。rep 初值为 1。

开始时，将数组中下标最大的物品放入栈中，然后开始考察该物品（出栈）。该物品满足放入背包的条件，第（4）（5）空完成将物品放入背包的操作，因此第（4）个空应填“x.s-w[x.n--]”，修改背包可容纳物品的重量，第（5）个空应填“stack[++top]”，将下一个待考察的物品放入栈中。若该物品不满足放入背包的条件，则将该物品从背包中取出，因此将 rep 置为 0，结束循环 while(!k && rep)。将物品从背包中取出，也就是释放该物品在背包中所占的重量，并标记为不能放入背包（job=2），再将其放入栈中；然后继续考察数组 w 中的下一个物品，因此需要结束循环 wile(top >= 1&& rep)，将 rep 置为 0，所以第(6)个空应填 rep=0。第(3)个空要求给出循环结束条件，即可以继续选取物品的条件。因此第(3)个空应填“top >= 1 && !k”。

解答

knap(s - w[n],n - 1)

knap(s,n - 1)

top => 1 && !k 或 top > 0 && k==0

x.s - w[x.n --]

stack[++ top]

rep=0

试题 6　分析

由程序正文可知，程序主要由类 Item 和 List 组成。类 Item 定义多项式中的项，它定义了 3 个私有数据成员:系数 quot、指数 exp 和指向多项式的下一项的指针 next。该类中定义了一个有两个参数的构造函数,其作用是创建系数为_quot,指数为_exp 的项(即创建类 Item 的一个对象），因此第(1)个空应填“quot=_quot；exp=_exp；next=NULL”。

类 List 定义了多项式的操作，数据成员 list 是多项式链表的链头指针。类的成员函数 createList()的功能是创建按照指数降序顺序链接的多项式,创建的方法是不断地将新的项插入到链表中合适的位置上。若链表中存在一项 P，其指数大于待插入项的指数，则待插入项应该是 p 的前驱。该成员函数在链表上遍历，直至找到满足上述条件的项；若链表中不存在这样的项，那么待插入成为新的链尾。因此第(2)个空应填“p!=NULL && exp < p - > exp”；第(3)个空应填“new Item (quot, exp)”,根据读入的指数和系数创建要插入的项。若链表中存在指数与待插入项指数相等的项，则合并同类项。

成员函数 multiplyList (List L1, List L2)计算多项式 L1 和 L2 的乘积。首先计算 L1 和 L2 的乘积多项式的最高幂次，即多项式 L1 和 L2 的最高次项的指数之和。由于多项式 L1 和 L2 的各项是按照指数降序排列的，两个多项式的第一项分别是最高次幂项，这两项的指数之和就是乘积多项式的最高次幂，因此第(4)个空应填“L1.list -> exp+L2.list -> exp”。

为了实现系数的乘积求和计算，当多项式 L1 从幂次高至幂次低逐一考虑各项的系数时，多项式 L2 应从幂次低至幂次高的顺序考虑各项的系数，以便将两多项式中所有幂次和为 k 的两项系数相乘后累计。由于是单链表，所以成员函数先将其中多项式 L2 的链接顺序颠倒，待计算完成后，再将多项式 L2 的链接顺序颠倒，即恢复多项式 L2 原来的链接顺序。乘积多项式从最高次(L1 与 L2 的幂次和)项系数至 0 次幂系数的顺序逐一计算。为求 k 次幂这一项的系数，对于 L1 的考察顺序是从最高次幂项至最低次项，而 L2 考查顺序是从最低幂次项至最高次项。首先跳过多项式 L1 中高于 k 次幂的项，设低于 k 次幂的项最高次幂是 j 次幂；

对多项式 L2，跳过低于 k-j 次幂的项，因此第(5)个空应填“ pL1 -> exp + pL2 -> exp < k ”。接着是一个顺序考虑多项式 L1 和 L2 等剩余各项的循环，若多项式的当前项幂次和为 k，则累计它们系数的乘积，并分别准备考虑下一项，因此第(6)个应填“ quot += pL1 ->quot * pL2 -> quot ”；若两多项式的当前项幂次和大于 k，则应考虑幂次和更小的项，这样应准备考虑多项式的 L1 的下一项；若两多项式的当前项幂次和小于 k，则应考虑幂次和更大的项，准备考虑多项式 L2 的下一项。若所有幂次和为 k 的项的系数乘积之和不等于 0，则应乘积多项式中有这一项，生成这一项的新结点(让指针 u 指向该结点)，并将它插在乘积多项式的末尾。

解答

quot=_quot；exp=_exp；next=NULL

p！＝NULL && exp < p -> exp

new Item (quot, exp)

L1.list -> exp +L2.list - >exp

pL1 -> exp +pL2 -> exp < k

quot += pL1 -> quot*pL2->quot