

王道模拟试题（五）答案

一、单项选择题

1. D。

【解析】考查栈和队列的区别。栈和队列的逻辑结构都是线性的，都有顺序存储和链式存储，有可能包含的运算不一样，但不是主要区别，任何数据结构在针对具体问题时所包含的运算都可能不同。插入和删除运算的限定不一样才是栈和队列的最主要区别。

2. A。

【解析】考查出入栈序列和栈深的关系。由于栈的最大深度不能超过 3。故第一个出栈元素不能是 5 或 4，第二个出栈的元素不能是 5，由此可以排除 B、C、D。

3. A。

【解析】考查栈的应用。设中间计算结果 $S1=C/D$ ， $S2=(B+C/D)$ ，则扫描过程如下：

扫描字符	运算数栈（扫描后）	运算符栈（扫描后）	说明
A	A		‘A’ 入栈
-	A	-	‘-’ 入栈
(A	-(‘(’ 入栈
B	A B	-(‘B’ 入栈
+	A B	-(+	‘+’ 入栈
C	A B C	-(+	‘C’ 入栈
/	A B C	-(+ /	‘/’ 入栈
D	A B C D	-(+ /	‘D’ 入栈
	A B S1	-(+	计算 S1
)	A B S1	-(+)	‘)’ 入栈
	A S2	-	计算 S2
×	A S2	- ×	‘×’ 入栈
E	A S2 E	- ×	‘E’ 入栈

扫描到 E 时，运算符栈中的内容依次是“-×”，因此选 A。

4. D。

【解析】考查二叉树的遍历。对于 I，显然任何遍历都相同。对于 II，根结点无右孩子，此时前序遍历先遍历根结点，中序遍历最后遍历根结点，所以不相同。对于 III，是一棵左单支树，前序遍历和后序遍历的序列相反。对于 IV，所有结点只有右子树的右单支树，前序遍历和中序遍历的序列相同。选 D。

5. C。

【解析】考查平衡二叉树的性质与查找操作。设 N_h 表示深度为 h 的平衡二叉树中含有的最少结点数，有： $N_0=0$ ， $N_1=1$ ， $N_2=2$ ，...， $N_h=N_{h-1}+N_{h-2}+1$ ， $N_3=4$ ， $N_4=7$ ， $N_5=12$ ， $N_6=20>15$ （考

生应能画出图形)。也就是说,高度为 6 的平衡二叉树最少有 20 个结点,因此 15 个结点的平衡二叉树的高度为 5,而最小叶子结点的层数为 3,所以选项 D 错误。选项 B 的查找过程不能构成二叉排序树,错误。选项 A 根本就不包含 28 这个值,错误。

6. A。

【解析】考查完全二叉树性质。完全二叉树第 5 层共有 $2^4=16$ 个结点。第 6 层最左边有 3 个叶子结点,对应第 5 层最左边 2 个结点,所以第 5 层右边有 $16-2=14$ 个叶子结点,因此共有 17 个叶子结点。

【另解】画出草图的片段部分进行求解,比较形象且不易出错。

7. B。

【解析】考查无向完全图的性质。 n 个结点的无向完全图共有 $n(n-1)/2$ 条边。对于 $n+1$ 个结点和 $n(n-1)/2$ 边构成的非连通图,仅当 n 个顶点构成完全图、第 $n+1$ 个顶点构成一个孤立顶点的图;若再增加一条边,则在任何情况下都是连通的。 n 个顶点构成的无向图中,边数 $\leq n(n-1)/2$,将 $e=36$ 代入,有 $n \geq 9$,现已知无向图是非连通的,则 n 至少为 10。

8. D。

【解析】考查拓扑序列的性质。选项 D 中的情况是不可能出现的,因此若 G 中有一条 V_j 到 V_i 的路径,则在图的拓扑序列中顶点 V_j 应该在顶点 V_i 之前。以分析中的示例说明:若有一条 V_j 到 V_i 的路径,说明 V_j 是 V_i 的前驱,则拓扑排序 V_j 应该在 V_i 的前面,显然矛盾。

9. D。

【解析】考查散列表的构造过程。任何散列函数都不可能绝对的避免冲突,因此采用合理的冲突处理方法,为冲突的关键字寻找下一个“空”位置。将前面各元素分别放入散列表中,其中 8、9、10 的位置分别存放 25、26、8。元素 59 经过哈希函数计算应该存入位置 $59 \bmod 17 = 8$,发生冲突,采用线性探测再散列,依次比较 9、10、11,发现 11 为空,所以将其放入地址 11 中。各关键字对应的散列地址见下表。

关键字	26	25	72	38	8	18	59
散列地址	9	8	4	4	8	1	8

10. A。

【解析】考查各种排序算法的特点。冒泡排序和选择排序经过两趟排序之后,应该有两个最大(或最小)元素放在其最终位置;插入排序经过两趟排序之后,前 3 个元素应该是局部有序的;只有可能是快速排序。

注意:在排序过程中,每一趟都能确定一个元素在其最终位置的有:冒泡排序、简单选择排序、堆排序、快速排序,其中前三者能形成全局有序的连续子序列,后者能确定枢轴元素的最终位置。直接插入排序每一趟排序形成的有序子序列只是局部有序的。

11. A。

【解析】考查快速排序的性质。当待排序列有序或接近有序时,快速排序的效率最低。当每次的枢轴都把表等分为长度相近的二个子表时,速度是最快的。选项 A 第一趟枢轴 21 将表划分为二个子表 {9,17,5} 和 {25,23,30},而后对两个子表划分时,枢轴再次地将它们等分,所以该序列是快速排序的最优情况,速度最快。

12. A。

【解析】本题考查计算机的性能指标。CPI是一种衡量CPU性能的指标，即执行一条指令所需的时钟周期数，系统结构、指令集、计算机组织都会影响到CPI。时钟频率不会影响到CPI，但可以加快指令的执行速度。如一条指令的执行需要10个时钟周期，则执行这条指令时钟频率为1GHz的CPU比100MHz的CPU要快。

13. B。

【解析】考查不同进制数之间的转换与算术移位运算。对于本类题型，应先将-1088 转换为 16 位的补码表示，执行算术右移后，再转换为十六进制数。R1 的内容首先为 $[-1088]_{补}=1111\ 1011\ 1100\ 0000B=FBC0H$ 。算术右移 4 位的结果为 $1111\ 1111\ 1011\ 1100B=FFBCH$ ，则此时 R1 中的内容为 FFBCH。

注意：算术移位时保持最高的符号位不变，对于正数（符号位为 0），原码、补码、反码的算术左移/右移都是添 0；对于负数（符号位为 1），添补规则见下表。

原码	0
补码	左移添 0，右移添 1
反码	1

14. B。

【解析】本题考查机器零。只有当数据发生“上溢”，才会终止运算操作，转去进行溢出处理，A 错误。规格后化可以判断运算结果是否上溢出（超过表示范围），但和机器零没有关联，C 错误。定点数中所表示的 0，是实实在在的 0（坐标轴上的），而不是趋近 0 的机器零，D 错误。在各种数码的表示法中，移码相当于真值在坐标轴上整体右移至正区间内，当移码表示的阶码全 0 时，为阶码表示的最小负数，此时直接认为浮点数是机器零，B 正确。

注意：当浮点运算结果在 0 到最小正数之间（正下溢）或最大负数到 0 之间（负下溢）时，浮点数值趋于 0，计算机仅将其当做机器零处理。

15. C。

【解析】本题考查 Cache 的性能因素。Cache 的命中率指 CPU 要访问的信息已在 Cache 中的比率。显然与 Cache 的存取速度无关，而选项 ABD 与 Cache 的命中率都有一定的关系。

16. B。

【解析】本题考查 FIFO 算法。FIFO 算法指淘汰先进入的，易知替换顺序为：

走向	0	1	2	4	2	3	0	2	1	3	2	3	0	1	4
c			2	2	2	2	0	0	0	3	3	3	3	3	3
b		1	1	1	1	3	3	3	1	1	1	1	1	1	4
a	0	0	0	4	4	4	4	2	2	2	2	2	0	0	0
命中否					√						√	√		√	

表中除了标注为命中的，其余均未命中，所以命中率为 $4/15=26.7\%$ 。

17. A。

【解析】本题考查基址寻址和变址寻址的区别。两者的有效地址都加上了对应寄存器的

内容，都扩大了指令的寻址范围，Ⅰ 正确。变址寻址适合处理数组、编制循环程序，Ⅱ 正确。基址寻址有利于多道程序设计，Ⅲ 正确。基址寄存器的内容由操作系统或管理程序确定，在执行过程中其内容不变，而变址寄存器的内容由用户确定，在执行过程中其内容可变，故Ⅳ和Ⅴ 错误。

注意：基址寻址和变址寻址的真实地址 EA 都是形式地址 A 加上一个寄存器中的内容。

18. C。

【解析】本题考查取指周期完成的操作。CPU 首先需要取指令，取指令阶段的第一个操作就是将指令地址（程序计数器 PC 中的内容）送往存储器地址寄存器。题干中虽然给出了一条具体的指令“MOV R0, #100”，实际上 CPU 首先要完成的操作是取指令，与具体指令是没有关系的。

注意：取指周期完成的微操作序列是公共的操作，与具体指令无关。

19. C。

【解析】考查流水线的时空图。流水线在开始时需要一段建立时间，结束时需要一段排空时间，设 m 段流水线的各段经过时间均为 Δt ，则需要 $T_0 = m\Delta t$ 的时间建立流水线，之后每隔 Δt 就可以流出一条指令，完成 n 个任务共需时间 $T = m\Delta t + (n - 1)\Delta t$ 。具有三个功能段的流水线连续执行 10 条指令共需时间 = 3+9 =12。

20. B。

【解析】本题考查总线的定时方式。由题意可知，医生是主模块，护士是从模块。医生伸出手后（即主模块发出请求信号），等待护士将手术刀递上（主模块等待回答信号），护士也必须等待医生握紧后才松开收（从模块等待主模块的回答信号），以上整个流程就是异步通信的全互锁方式。

21. C。

【解析】本题考查外中断方式和 DMA 方式的区别。和中断方式相比，DMA 连接的是高速设备，其优先级高于中断请求，以防止数据丢失，Ⅰ 正确。DMA 请求的响应时间可以发生在每个机器周期结束时，只要 CPU 不占用总线，而中断请求的响应时间只能发生在每条指令执行完毕，Ⅱ 错误。通常情况下，DMA 的优先级要高于外中断，所以 DMA 优先级一般要比非屏蔽中断请求要高，Ⅲ 错误。如果不开中断，非屏蔽中断（以及内中断）仍可响应，Ⅳ 错误。在 DMA 方式的预处理和后处理中，需要 CPU 的干预，只是在传送的过程中不需要 CPU 的干预，Ⅴ 错误。

注意：中断方式具有对异常时间的处理能力，而 DMA 方式仅局限于完成传送数据块的能力。

22. B。

【解析】本题考查通道的基本工作过程。通道的基本工作过程：用户程序使用访管指令进入操作系统管理程序；CPU 通过管理程序组织一个通道程序，并用 I/O 指令启动通道；通道执行通道指令，完成 I/O 操作；通道程序结束后向 CPU 发中断请求。通道程序放与主存之中，由 CPU 执行 I/O 指令启动通道，通道执行通道程序。在整个传输过程中，数据传输结束时，需要中断来处理。

23. D。

【解析】本题考查操作系统功能的实现。中断处理流程的前 3 个步骤是由硬件直接实现（隐指令）的；时钟管理需要硬件计数器保持时钟的运行；地址映射中需要基地址（或页表）寄存器和地址加法器的支持。页面调度是由相关调度算法完成，不需要硬件支持。

对于此类题型，需要掌握各个选项的基本原理才能解答。

24. B。

【解析】本题考查短进程优先调度算法和平均周转时间。J1 第一个执行，J1 在 10:00 执行完毕；之后调度 J4；接着调度 J3；最后调度 J2。故平均周转时间 $= (2+3.3+1.9+1.2)/4=2.1$ 。

25. A。

【解析】本题考查互斥信号量的设置。互斥信号量的初值应为可用资源数，在本题中为可同时进入临界区的资源数。每当一个进程进入临界区，S 减 1，减到 $-(n-m)$ 为止，此时共有 $|S|$ 个进程在等待进入。

26. B。

【解析】本题考查死锁的性质。死锁是由于多个进程因竞争资源而造成的一种僵局（互相等待），若无外力作用，这些进程都将无法向前推进。A、C、D 都是对有限资源的竞争造成的僵局，属于死锁。而 B 是由于资源不足造成的饥饿。

27. B。

【解析】本题考查页面置换算法与抖动、Belady 现象。I 正确，举例如下：页面走向为 1,2,3,4,1,2,5,1,2,3,4,5 时，当分配 3 帧时产生 9 次缺页中断，分配 4 帧时产生 10 次缺页中断。最近最少使用法不会产生 Belady 现象，II 错误。若页面在内存中，不会产生缺页中断，也不会出现页面的调入/调出，而不是虚拟存储器（包括作为虚拟内存那部分硬盘），故 III 错误、IV 正确。

28. B。

【解析】本题考查抖动现象的分析。从测试数据看，CPU 不忙，交换空间也不满，就是硬盘 I/O 非常忙，所以不是交换空间不够，系统也没有死锁，主要瓶颈在内外存交换上，因此最可能的情况就是抖动，即由于内存紧缺，并发进程数多，采用按需调页而引起的频繁的换入换出作业。对于抖动问题的解决，最好的办法是增加内存或减少并发进程数（提高各个进程的物理块数），单纯地增大交换分区或增加 CPU 都没有解决根本问题。

注意：内存出现的异常，如抖动和 Belady 现象，都要从产生原因的角度认真分析。在做这道题的同时，也可以总结一下死锁、饥饿这些进程管理中会出现的异常，互相对比，举一反三。首先判断系统异常属于哪种异常。

29. A。

【解析】考查逻辑地址和物理地址的转换。块大小为 $128KB/32=4KB$ ，因为块与页面大小相等，所以每页为 4KB。第 3 页被装入到主存第 6 块中，故逻辑地址 $[3, 70]$ 对应的物理地址为 $4KB*6+70=24576+70=24646$ 。

30. D。

【解析】本题考查文件的物理结构。物理文件的组织是文件管理的内容，而文件管理是操作系统的主要功能之一；此外存储介质的特性也决定了文件的物理结构，如磁带机只能采用顺序存放方式。

31. A。

【解析】本题考查磁盘读取的速度。由题中条件知，旋转速度为 3000 转/分=50 转/秒，即 20ms/转。

读一个扇区需要时间为 $20/8=2.5\text{ms}$ 。

读一个扇区并将扇区数据送入内存需要时间为 $2.5 \times 3=7.5\text{ms}$ 。

读出一个磁道上的所有扇区需要时间为 $20/2+8 \times 7.5=70\text{ms}=0.07\text{s}$ 。

每磁道数据量为 $8 \times 512=4\text{KB}$ 。

数据传输速度为 $4\text{KB}/0.07\text{s}=57.1\text{KB/s}$ 。

故依次读出一个磁道上的所有扇区需要 0.07s，其数据传输速度为 57.1KB/s。

32. C。

【解析】本题考查虚拟设备的概念。虚拟设备是指采用虚拟技术将一台独占设备转换为若干台逻辑设备的情况。这种设备并不是物理地变成共享设备，而是用户在使用它们时“感觉”是共享设备，是逻辑的概念。引入虚拟设备的目的是为了克服独占设备速度慢、利用率低的特点。

33. A。

【解析】本题考查 OSI 参考模型和 TCP/IP 模型的比较。在 OSI 参考模型中，网络层支持无连接和面向连接的两种方式，传输层仅有面向连接的方式。而 TCP/IP 模型认为可靠性是端到端的问题，因此它在网络层仅支持无连接的方式，但在传输层支持无连接和面向连接的两种方式。

34. B。

【解析】本题考查物理层设备。电磁信号在网络传输媒体中进行传递时会衰减而使信号变得越来越弱，还会由于电磁噪声和干扰使信号发生畸变，因此需要在一定的传输媒体距离中使用中继器，来对传输的数据信号整形放大后再传递。放大器常用于远距离模拟信号的传输，它同时也会使噪声放大，引起失真。网桥用来连接两个网段以扩展物理网络的覆盖范围。路由器是网络层的互连设备，可以实现不同网络的互连。中继器的工作原理是信号再生（不是简单的放大），从而延长网络的长度。

35. C。

【解析】本题考查最小帧长与信道利用率。在确认帧长度和处理时间均可忽略不计的情况下，信道的利用率 $\approx t_{\text{发送时间}} / (t_{\text{发送时间}} + 2 \times t_{\text{传播时间}})$ 。根据信道利用率的计算公式，当发送一帧的时间等于信道的传播时延的 2 倍时，信道利用率是 50%，或者说当发送一帧的时间等于来回路程的传播时延时，效率将是 50%，即 $20\text{ms} \times 2 = 40\text{ms}$ 。现在发送速率是 4000 bps，即发送一位需要 0.25ms，则帧长 $40/0.25=160\text{bit}$ 。

36. C。

【解析】考查各种网络设备。路由器用于分割广播域，路由器和交换机用于分割冲突域，

而集线器既不能隔离冲突域又不能隔离广播域。所以上图中一共存在两个广播域，6（左边 1 个，右边 5 个）个冲突域。

37. A。

【解析】本题考查特殊的 IP 地址。几类重要的特殊地址如下：

特殊地址	Net-id	Host-id	源地址或目的地址
网络地址	特定的	全 0	都不是
直接广播地址	特定的	全 1	目的地址
受限广播地址	全 1	全 1	目的地址
这个网络上的主机	全 0	全 0	源地址
这个网络上的特定主机	全 0	特定的	源地址
环回地址	127	任意	源地址或目的地址

网络的广播地址就是将主机位全部置为 1；/26 表示 32 位 IP 地址中前 26 都是网络号，最后 6 位是主机号。131 的二进制形式为 10000011。根据广播地址的定义，主机段全 1 即为广播地址，即 10111111，转换为十进制为 191，故广播地址为 172.16.7.191。

38. C。

【解析】本题考查 IP 报头字段的功能和 ICMP 报文。ICMP 报文作为 IP 分组的数据字段，用它来使得主机或路由器可以报告差错和异常情况。路由器对 TTL 值为零的数据分组进行丢弃处理，并向源主机返回时间超时的 ICMP 报文。

39. A。

【解析】本题考查默认路由的配置。所有的网络都必须使用子网掩码，同时在路由器的路由表中也必须有子网掩码这一栏。一个网络如果不划分子网，就使用默认子网掩码。默认子网掩码中 1 的位置和 IP 地址中的网络号字段 net-id 正好相对应。主机地址是一个标准的 A 类地址，其网络地址为 11.0.0.0。选项 I 的网络地址为 11.0.0.0，选项 II 的网络地址为 11.0.0.0，选项 III 的网络地址为 12.0.0.0，选项 IV 的网络地址为 13.0.0.0，因此，和主机在同一网络的是选项 I 和选项 II。

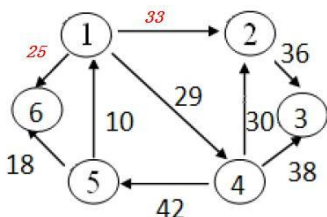
40. A。

【解析】本题考查发送窗口与拥塞窗口和接收窗口的关系。题中出现了拥塞窗口和接收端窗口，为了保证 B 的接收缓存不发生溢出，发送窗口应该取两者的最小值。先看拥塞窗口，由于慢开始门限值为 2KB，第一个 RTT 中 A 拥塞窗口为 4KB，按照拥塞避免算法，收到 B 的确认报文后，拥塞窗口增长为 5KB。再看接收端窗口，B 通过确认报文中窗口字段向 A 通知接收端窗口，那么接收端窗口为 2KB。因此在下一次发送数据时，A 的发送窗口应该为 2KB，即一个 RTT 内最多发送 2KB。

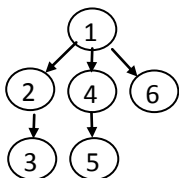
二、综合应用题

41. 解析：

（1）该邻接表存储对应的带权有向图如下：



(2) 以顶点 v_1 为起点的广度优先搜索的顶点序列依次为 $v_1, v_2, v_4, v_6, v_3, v_5$ ，对应的生成树如下：



(3) 生成树：顶点集合 $V(G)=\{v_1, v_2, v_3, v_4, v_5, v_6\}$ ，边的集合 $E(G)=\{(v_1, v_2), (v_2, v_3), (v_1, v_4), (v_4, v_5), (v_5, v_6)\}$ 。（图略）

(4) v_1 到 v_3 最短路径为 67: $(v_1-v_4-v_3)$ 。

(5) 从 v_1 点开始，第一趟寻找 v_1 和点集 $\{v_2, v_3, v_4, v_5, v_6\}$ 之间的最小权值的边。(v_5, v_1)。

第二趟寻找点集 $\{v_1, v_5\}$ 和点集 $\{v_2, v_3, v_4, v_6\}$ 之间的最小权值的边。(v_5, v_6)。

第三趟寻找点集 $\{v_1, v_5, v_6\}$ 和点集 $\{v_2, v_3, v_4\}$ 之间的最小权值的边。(v_1, v_4)。

第四趟寻找点集 $\{v_1, v_4, v_5, v_6\}$ 和点集 $\{v_2, v_3\}$ 之间的最小权值的边。(v_4, v_2)。

第五趟寻找点集 $\{v_1, v_2, v_4, v_5, v_6\}$ 和点集 $\{v_3\}$ 之间的最小权值的边。(v_2, v_3)。

所以最小生成树的边集合为 $\{(v_5, v_1), (v_5, v_6), (v_1, v_4), (v_4, v_2), (v_2, v_3)\}$ （图形略）。

42. 解析：

(1) 基本的设计思想：

一个数字出现的次数超过了长度的一半，那么我们可以这样认为这个数字出现的个数一定大于其他全部数字出现的个数之和。算法的步骤如下：

① 设数组为 $data[]$ ，数组长度为 n ， $i=1$ 。置 $currentAxis=data[0]$ ， $currentNum=1$ 。

② 当 $data[i]==currentAxis$ 时， $currentNum++$ ，转向④；否则转向③。

③ $currentNum--$ ，当 $currentNum==0$ 时， $currentAxis=data[i]$ 。

④ 当 $i==data.length$ 时，转向⑤；否则， $i++$ ，转向②；

⑤ 返回 $currentAxis$ 。

(2) 算法的实现如下：

```
int funtion(int data[],int length){
    int currentAxis;           //假设要求的元素的位置
    int currentNum = 0;        //要求元素的个数变量
    for(int i =0;i<length;i++){
        if(currentNum ==0){
            //如果要求的元素还没出现，设置要求的元素为现在要比较的元素
            currentAxis=data[i];
        }
        if(data[i]==currentAxis){
            currentNum++;
        }else{
            currentNum--;
        }
        if(currentNum==0){
            currentAxis=data[i];
        }
    }
    return currentAxis;
}
```



```

        currentNum=1;
    }
    else
    {
        if(currentAxis==data[i])
            //假设的结果与比较元素不同，假设结果个数增 1，否则减 1
            currentNum++;
        else
            currentNum--;
    }
}
return currentAxis;           //返回最终结果的位置。
}

```

(3) 时间复杂度为 $O(n)$ 。空间复杂度 $O(1)$ 。

【另解】本题最直接的方法就是对每个数字进行排序。然后输出出现次数最大的数字即可，但排序的时间复杂度最快也要 $O(n\log_2 n)$ 。

43. 解析：

(1) x 是无符号整数，所有的二进制位均为数值位， $C000\ 0004H$ 的真值为 $2^{31}+2^{30}+2^2$ 。 $x/2$ 是由逻辑右移一位得到的，即 $(2^{31}+2^{30}+2^2)/2$ ，其真值为 $2^{30}+2^{29}+2$ ，存放在 $R1$ 中的机器码是 $6000\ 0002H$ 。 $2x$ 是由 x 逻辑左移一位得到的，真值发生溢出，存放在 $R1$ 中的机器码是 $8000\ 0008H$ 。

(2) 机器码 $C000\ 0004H$ 表示这是一个负数，数值位取反末位加 1，得到的二进制真值为 $-0111\ 1111\ 1111\ 1111\ 1111\ 1111\ 1111\ 1100$ ，对应的十进制真值为 $-(2^{30}-2^2)$ 。 $x/2$ 是由 x 算术右移一位得到的，其真值为 $-(2^{29}-2)$ ，存放在 $R1$ 中的机器码是 $E000\ 0002H$ 。 $2x$ 是由 x 算术左移一位得到的，其真值为 $-(2^{31}-2^3)$ ，存放在 $R1$ 中的机器码是 $8000\ 0008H$ 。

(3) 在 IEEE754 单精度浮点数中，最高位为数符位；其后是 8 位阶码，以 2 为底，用移码表示，阶码的偏置值为 127；其后 23 位是尾数数值位，隐藏数值的最高位“1”。转换为二进制 1 100 0000 0 000 0000 0000 0000 0000 0100，可知， x 为负数，阶码为 1，尾数为 $1+2^{-21}$ ，故真值为 $-(1+2^{-21})\times 2$ 。 $x/2$ 的真值是 $-(1+2^{-21})$ ，存放在 $R1$ 中的机器码为 1 011 1111 1 000 0000 0000 0000 0000 0100，即 $BF80\ 0004H$ 。 $2x$ 的真值是 $-(1+2^{-21})\times 2^2$ ，存放在 $R1$ 中的机器码为 1 100 0000 1 000 0000 0000 0000 0000 0100，即 $C080\ 0004H$ 。

44. 解析：

(1) 取指周期：

节拍T0: PC->MAR, 1->R (注: M->MAR)

节拍T1: M(MAR)->MDR,(PC)+1->PC

节拍T2: MDR->IR,OP(IR)->ID

执行周期：

节拍T0: K(IR)->MAR

节拍T1: PC->MDR, 1->W (注: M+1->MDR)

节拍T2: MDR->M(MAR),K+1->PC

(2) 采用微程序控制, 还需要增加的微操作有:

M->CMAR

//将取指周期微程序首地址放入

CM(CMAR)->CMDR

//将对应控存M地址单元中的第一条微指令独到控存数据寄存器中

AD(CMDR)->CMAR

//让微指令的顺序控制字段指出下一条微指令的地址为M+1, 送入CMAR

(3) $2^6=64$ 个微程序, 一条机器指令对应一段微程序。

(4) 微程序控制器采用了“存储程序”的原理, 每条机器指令对应一个微程序, 因此修改和扩充容易, 灵活性好, 但每条指令的执行都要访问控制存储器, 所以速度慢。硬布线控制器采用专门的逻辑电路实现, 其速度主要取决于逻辑电路的延迟, 因此速度快, 但修改和扩

45. 解析:

本题考查逻辑地址到物理地址的转换、页面置换等。地址转换过程一般是先将逻辑页号取出, 然后查找页表, 得到页框号, 将页框号与页内偏移量相加, 即可获得物理地址, 若取不到页框号, 那么该页不在内存, 于是产生缺页中断, 开始请求调页。若内存有足够的物理页面, 那么可以再分配一个新的页面, 若没有页面了, 就必须在现有的页面之中找到一个页, 将新的页与之置换, 这个页可以是系统中的任意一页, 也可以是本进程中的一页, 若是系统中的一页, 则这种置换方式称为全局置换, 若是本进程的页面, 则称为局部置换。置换时为尽可能地减少缺页中断次数, 可以有多种算法来应用, 本题使用的是改进的 CLOCK 算法, 这种算法必须使用页表中的引用位和修改位, 由这 2 位组成 4 种级别, 没被引用和没修改的页面最先淘汰, 没引用但修改了的页面其次, 再者淘汰引用了但是没修改的页面, 最后淘汰既引用又修改的页面, 当页面的引用位和修改位相同时, 随机淘汰一页。

(1) 根据题意, 每页1024字节, 地址又是按字节编址, 计算逻辑地址的页号和页内偏移量, 合成物理地址如下表所示。

逻辑地址	逻辑页号	页内偏移量	页框号	物理地址
0793	0	793	4	4889
1197	1	173	3	3245
2099	2	51	--	缺页中断
3320	3	248	1	1272
4188	4	92	--	缺页中断
5332	5	212	5	5332

以逻辑地址0793为例, 逻辑页号为 $0793/1024=0$, 在页表中存在, 页内偏移量为 $0793\%1024=793$, 对应的页框号为4, 故物理地址为 $4\times1024+793=4889$ 。

(2) 第2页不在内存, 产生缺页中断, 根据改进CLOCK算法, 第3页为没被引用和没修改的页面, 故淘汰。新页面进入, 页表修改如下:

逻辑页号	存在位	引用位	修改位	页框号	
------	-----	-----	-----	-----	--

0	1	1	0	4	
1	1	1	1	3	
2	0→1	0->1	0	--→1	调入
3	1→0	0	0	1→--	淘汰
4	0	0	0	--	
5	1	0	1	5	

因为页面2调入是为了使用，所以页面2的引用位必须改为1。

地址转换变为如下表：

逻辑地址	逻辑页号	页内偏移量	页框号	物理地址
0793	0	793	4	4889
1197	1	173	3	3245
2099	2	51	1	1075
3320	3	248	--	缺页中断
4188	4	92	--	缺页中断
5332	5	212	5	5332

46 . 解析：

本题考查缺页中断和页面置换算法。

(1) 缺页中断是一种特殊的中断，它与一般中断的区别是：①在指令执行期间产生和处理中断信号。CPU通常在一条指令执行完后检查是否有中断请求，而缺页中断是在指令执行时间，发现所要访问的指令或数据不在内存时产生和处理的；②一条指令在执行期间可能产生多次缺页中断。如一条读取数据的多字节指令，指令本身跨越两个页面，若指令后一部分所在页面和数据所在页面均不在内存，则该指令的执行至少产生两次缺页中断。

(2) 每个页面大小为100字节，则页面的访问顺序如下：

10	11	104	170	73	309	185	245	246	434	458	364
0	0	1	1	0	3	1	2	2	4	4	3

采用FIFO算法的页面置换情况如下表，共产生缺页中断6次。

走向	0	0	1	1	0	3	1	2	2	4	4	3
块号1	0	0	1	1	1	3	3	2	2	4	4	3
块号2			0	0	0	1	1	3	3	2	2	4
淘汰						0		1		3		2
缺页	√		√			√		√		√		√

采用LRU算法的页面置换情况如下表，共产生缺页中断7次。

走向	0	0	1	1	0	3	1	2	2	4	4	3
块号1	0	0	1	1	0	3	1	2	2	4	4	3
块号2			0	0	1	0	3	1	1	2	2	4
淘汰						1	0	3		1		2
缺页	√		√			√	√	√		√		√

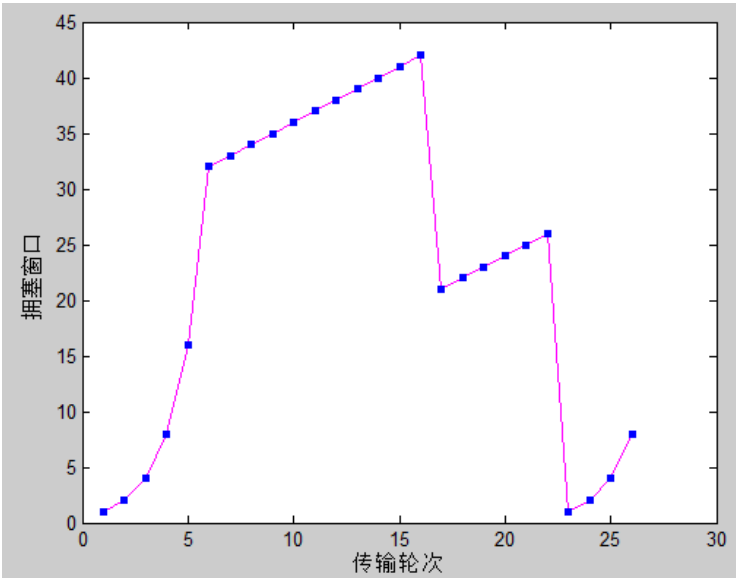
(3) 设可接受的最大缺页中断率为ρ。若要访问页面在内存中，一次访问的时间是10ms(访

问内存页表)+10ms(访问内存)=20ms。如果不在内存，所花时间为10ms(访问内存页表)+25ms(中断处理)+10ms(访问内存页表)+10ms(访问内存)=55ms。平均有效访问时间：
 $20ms \times (1-p) + 55ms \times p \leq 22ms$ ，得可接受的最大缺页中断率 p 为5.7%。

47. 解析：

本题考查 TCP 的拥塞控制算法。在画出拥塞窗口与传输轮次的曲线后，根据四种拥塞控制算法的特点，以图像的拐点进行分段分析。初始时，拥塞窗口置为 1，即 $cwnd=1$ ，慢开始门限置为 32，即 $ssthresh=32$ 。慢开始阶段， $cwnd$ 初值为 1，以后发送方每收到一个确认 ACK， $cwnd$ 值加 1，也即经过每个传输轮次（RTT）， $cwnd$ 呈指数规律增长。当拥塞窗口 $cwnd$ 增长到慢开始门限 $ssthresh$ 时（即当 $cwnd=32$ 时），就改用拥塞避免算法， $cwnd$ 按线性规律加性增长。当 $cwnd=42$ 时，收到三个重复的确认，启用快恢复算法，更新 $ssthresh$ 值为 21（即变为超时时 $cwnd$ 值 42 的一半）。 $cwnd$ 重置 $ssthresh$ 减半后的值，并执行拥塞避免算法。当 $cwnd=26$ 时，网络出现拥塞，改用慢开始算法， $ssthresh$ 置为拥塞时窗口值得一半，即 13， $cwnd$ 置为 1。

(1) 拥塞窗口与传输轮次的关系曲线如图所示：



(2) 慢开始的时间间隔：[1, 6]和[23, 26]。拥塞避免的时间间隔：[6, 16]和[17, 22]。

(3) 在第 16 轮次之后发送方通过收到三个重复的确认检测到丢失的报文段。在第 22 轮次之后发送方是通过超时检测到丢失的报文段。

(4) 在第 1 轮次发送时，门限 $ssthresh$ 被设置为 32。

在第 18 轮次发送时，门限 $ssthresh$ 被设置为发生拥塞时的一半，即 21。

在第 24 轮次发送时，门限 $ssthresh$ 是第 22 轮次发生拥塞时的一半，即 13。

(5) 第 70 报文段在第 7 轮次发送出。

(6) 拥塞窗口 $cwnd$ 和门限 $ssthresh$ 应设置为 8 的一半，即 4。

王道模拟试题（六）答案

一、单项选择题

1. C。

【解析】考查时间复杂度。在程序中，执行频率最高的语句为“ $i=i*3$ ”。设该基本语句一共执行了 k 次，根据循环结束条件，有 $n>2*3^k\geq n/3$ ，由此可得算法的时间复杂度为 $O(\log_3 n)$ 。

2. A。

【解析】考查出入栈操作的性质。当 $p_1=3$ ，表示 3 最先出栈，前面 1、2 应在栈中，此时若出栈操作，则 p_2 应为 2；此时若进栈操作（进栈 1 次或多次），则 p_2 为 4、5、 \cdots 、 n 都有可能，故选 A。

3. C。

【解析】考查双端队列的操作。输入受限的双端队列是两端都可以删除，只有一端可以插入的队列；输出受限的双端队列是两端都可以插入，只有一端可以删除的队列。对于 A，可由输入受限的双端队列、也可由输出受限双端队列得到。对于 BCD，因为 4 第一个出队，所以之前输入序列必须全部进入队列。对于 B，在输入受限的双端队列中，输入序列是 1234，全部进入队列后的序列也为 1234，两端都可以出，所以可以得到 4132；在输出受限双端队列中，输入序列全部入队，1 肯定和 2 挨着，所以得不到 4132。对于 C，在输入受限的双端队列中，输入序列是 1234，全部进入队列后的序列也为 1234，在 4 出队后不可以把 2 直接出队。在输出受限双端队列中，也是 1 和 2 在序列进入队列中后必须挨着。所以也得不到。对于 D，在输入受限的双端队列中，输入序列是 1234，全部进入队列后的序列也为 1234，输出 4 后，应该是 1 和 3，所以得不到；在输出受限双端队列中，输入序列 1234，一端进 1，另一端进 2，再一端进 3，另一端进 4，可得到 4213 的输出序列。因此选 C。

4. C。

【解析】考查平衡二叉树的性质。在平衡二叉树的结点最少情况下，递推公式为 $N_0=0$ ， $N_1=1$ ， $N_2=2$ ， $N_h=1+N_{h-1}+N_{h-2}$ （ h 为平衡二叉树高度， N_h 为构造此高度的平衡二叉树所需最少结点数）。通过递推公式可得，构造 5 层平衡二叉树至少需 12 个结点，构造 6 层至少需要 20 个。

5. C。

【解析】考查二叉排序树的构造过程。画出三个选项 ABC 构造的二叉排序树的草图即可知道答案，C 和 AB 构造的树形不同；再画出最后一个选项 D 构造的二叉排序树即可验证答案，D 和 AB 两项的相同。

6. C。

【解析】考查哈夫曼树的构造。将 16 个权值相等（设为 m ）的字母看成 16 个独立的结点；从中任选两个结点构成一棵新的二叉树（共 8 棵），新树的权值为 $2m$ ；再从 8 棵树中任选 2 棵构成新的二叉树（共 4 棵），新树的权值为 $4m$ ， \cdots ，如此继续，刚好能构成一棵满二叉树。

7. C。

【解析】考查图的基本性质。强连通有向图的任何顶点到其他所有顶点都有路径，但未必有弧，A 错误。图与树的区别是逻辑上的，而不是边数的区别，图的边数也可能小于树的边数。若 E' 中的边对应的顶点不是 V' 中的元素时，则 V' 和 $\{E'\}$ 无法构成图，D 错误。

8. D。

【解析】考查邻接矩阵的定义。一个含有 n 个顶点和 e 条边的简单无向图的邻接矩阵为 $n \times n$ 矩阵，共有 n^2 个元素，其中非零元素个数为 $2e$ ，则零元素个数为 $n^2 - 2e$ 。

9. C。

【解析】考查散列表的性质。不同冲突处理方法对应的平均查找长度是不同的，I 错误。散列查找的思想是通过散列函数计算地址，然后再比较关键字确定是否查找成功，II 正确。平均查找长度与填装因子（即表中记录数与表长之比）有关，III 错误。在开放定址的情况下，不能随便删除表中的某个元素（只能标记为删除状态），否则可能会导致搜索路径被中断，IV 错误。

10. C。

【解析】考查各种内部排序算法的性能。选择排序在最好、最坏、平均情况下的时间性能均为 $O(n^2)$ ，归并排序在最好、最坏、平均情况下的时间性能均为 $O(n \log_2 n)$ 。各种排序方法对应的时间复杂度见下表。

时间复杂度	直接插入	冒泡排序	简单选择	希尔排序	快速排序	堆排序	二路归并
平均情况	$O(n^2)$	$O(n^2)$	$O(n^2)$	-	$O(n \log_2 n)$	$O(n \log_2 n)$	$O(n \log_2 n)$
最好情况	$O(n)$	$O(n)$	$O(n^2)$	-	$O(n \log_2 n)$	$O(n \log_2 n)$	$O(n \log_2 n)$
最坏情况	$O(n^2)$	$O(n^2)$	$O(n^2)$	-	$O(n^2)$	$O(n \log_2 n)$	$O(n \log_2 n)$

11. C。

【解析】考查初始堆的建立。首先对以第 $\lfloor n/2 \rfloor$ 个结点为根的子树（也即最后一个结点的父结点为根的子树）筛选，使该子树成为堆，之后向前依次对各结点为根的子树进行筛选，直到筛选到根结点。从 $\lfloor n/2 \rfloor \sim 1$ 依次筛选堆的过程如下图所示：

12. C。

【解析】本题考查计算机的性能指标。CPI 指执行一条指令所需的时钟周期， $CPI = 15\text{MHz} / 10 \times 10^6 = 1.5$ 。这里的存储器延迟为迷惑项，与 CPI 的计算无关。

表 1 几种经常考查的性能指标

CPU 时钟周期	通常为节拍脉冲或 T 周期，即主频的倒数，它是 CPU 中最小的时间单位。
主频	机器内部主时钟的频率，主频的倒数是 CPU 时钟周期。 CPU 时钟周期 = $1 / \text{主频}$ ，主频通常以 MHz 为单位，1Hz 表示每秒一次。
CPI	执行一条指令所需的时钟周期数。
CPU 执行时间	运行一个程序所花费的时间。 CPU 执行时间 = CPU 时钟周期数 / 主频 = (指令条数 \times CPI) / 主频
MIPS	每秒执行多少百万条指令， $\text{MIPS} = \text{指令条数} / (\text{执行时间} \times 10^6) = \text{主频} / \text{CPI}$ 。
MFLOPS	每秒执行多少百万次浮点运算， $\text{MFLOPS} = \text{浮点操作次数} / (\text{执行时间} \times 10^6)$ 。

13. C。

【解析】本题考查补码的表示。因求最小值，故符号位取 1，为负数。补码负数的绝对值是数值部分按位取反，末位加 1，故剩下的两个“1”放在末位时，补码的绝对值最大，本题中对应最小负数，因此补码形式为 1000 0011，转换为原码为 1111 1101 =-7DH=-123。故选 C。

原码和补码的相互转换的规则。

对于正数（符号位为 0）：补码与原码的表示相同， $[x]_{补}=[x]_{原}$ 。

对于负数（符号位为 1）：符号位不变，数值部分按位取反，末位加 1。

14. B。

【解析】本题考查补码数的符号扩展。将 16 位有符号数扩展成 32 位有符号数，符号位不变，附加位是符号位的扩展。这个数是一个负数，而选项 A 表示正数，选项 C 数值部分发生变化，选项 D 用 0 来填充附加位，所以只有选项 B 正确。

注意：符号扩展的方法根据机器数的不同而不同，见下表所示。

正数		原符号位移动到新符号位上，新表示形式的所有附加位都用 0 进行填充。
负数	原码	原符号位移动到新符号位上，新表示形式的所有附加位都用 0 进行填充。
	反码、补码	原符号位移动到新符号位上，新表示形式的所有附加位都用 1 进行填充。

15. C。

【解析】本题考查交叉存储器的性能分析。低位交叉存储器连续读出 4 个字所需的时间为： $t = T + (m-1) \times r = 200 \text{ ns} + 3 \times 50 \text{ ns} = 350 \text{ ns} = 3.5 \times 10^{-7} \text{ s}$ 。故带宽为： $W = 64 \times 4 \text{ b} / (3.5 \times 10^{-7} \text{ s}) = 73 \times 10^7 \text{ b/s}$ 。

注意：在低位交叉存储器中，连续的地址分布在相邻的块中，而同一模块内的地址都是不连续的。这种存储器采用分时启动的方法，可以在不改变每个模块存取周期的前提下，提高整个主存的速度。

16. C。

【解析】本题考查页式存储器中地址映射的计算。对于本类题，先将物理地址转换为“物理页号+页内地址”的形式，然后查找页表以找出物理页号对应的逻辑页号，然后将“逻辑页号+页内地址”转换为对应的十进制数即可。 $32773 = 32768 + 5 = 1000 \text{ } 0000 \text{ } 0000 \text{ } 0000 \text{ B} + 101 \text{ B} = 1000 \text{ } 0000 \text{ } 0000 \text{ } 0101 \text{ B}$ ，后 12 位为页内地址，前 4 位为页号。物理页号为 8，对应逻辑页号为 3=11B。则逻辑地址=11 0000 0000 0101B=3×4K+5=10240+2048+5=12288+5=12293。

17. D。

【解析】本题考查各种数据寻址方式的原理。直接寻址 200 中，200 就是有效地址，所访问的主存地址 200 对应的内容是 300，I 错误。寄存器间接寻址（R）的访问结果与 I 一样，II 错误。存储器间接寻址（200）表示主存地址 200 中的内容为有效地址，所以有效地址为 300，访问的操作数是 400，III 错误。寄存器寻址 R 表示寄存器 R 的内容即为操作数，所以只有 IV 正确。此类题建议画出草图。

18. C。

【解析】本题考查控制器的组成。程序状态字（PSW）寄存器属于运算器的组成部分。PSW 包括两个部分：一是状态标志，如进位标志（C）、结果为零标志（Z）等，大多数指令的执行

将会影响到这些标志位；二是控制标志，如中断标志、陷阱标志等。

注意：控制器由程序计数器、指令寄存器、存储器地址寄存器、存储器数据寄存器、指令译码器、时序电路和微操作信号发生器等组成。

19. A。

【解析】本题考查字段直接编码的特点。互斥性微命令是指不能同时或不能在同一个 CPU 周期内并行执行的微命令，反之则是可以并行执行的微命令。字段直接编码将微指令的操作控制字段分成若干段，将一组互斥的微命令放在一个字段内，通过对这个字段的译码，便可对应每一个微命令。这样，各个字段的译码输出都是可以并行执行的微命令，这种编码方式提高了微指令的并行执行能力。

20. B。

【解析】本题考查总线的性能指标。总线的最大数据传输率又称总线带宽，即每秒传输的字节数。由于传送 4 个字节的数据需要 5 个时钟周期，总线带宽=总线宽度×总线频率=4B×500MHz÷5=400MB/s。

21. C。

【解析】本题考查总线的性能指标。总线带宽定义为总线的数据传输率，即单位时间内总线上传输数据的位数。I 和 III 直接决定总线带宽的计算结果，IV 间接影响总线的性能。

22. C。

【解析】考查中断方式的原理。中断周期中关中断是由隐指令完成，而不是关中断指令，I 错误。最后一条指令是中断返回指令，II 错误。CPU 通过 I/O 指令来控制通道，III 错误。

23. D。

【解析】本题考查用户态与和心态。打开定时器属于时钟管理的内容，对时钟的操作必须加以保护，否则，一个用户进程可以在时间片还未到之前把时钟改回去，从而导致时间片永远不会用完，那么该用户进程就可以一直占用 CPU，这显然不合理。从用户模式到内核模式是通过中断实现的，中断的处理过程很复杂，需要加以保护，但从内核模式到用户模式则不需要加以保护。读取操作系统内核的数据和指令是静态操作，显然无需加以保护。

24. A。

【解析】本题考查调度算法的性质。基于时间片的调度算法在执行过程中，进程的执行是以时间片为单位的。多级反馈队列调度算法在各个队列内以 FCFS 原则依次执行时间片，在最底层队列中按照时间片轮转算法执行。

25. D。

【解析】本题考查信号量机制。互斥信号量的初值为 1，P 操作成功则将其改成 0，V 操作成功将其改成 1。实现同步时，信号量的初值应根据具体情况来确定，若期望的消息尚未产生，则对应的初值应为 0；若期望的消息已经存在，则信号量的初值应设为一个非 0 的正整数。

注意互斥信号量和同步信号量的区别。信号量机制是每年考题的重点，这就要求考生能在理解的基础上熟练应用和掌握信号量。

26. B。

【解析】本题考查 PV 操作与死锁以及饥饿的关系。仔细考察程序代码，我们似曾相识，可以看出是一个扩展的单行线问题。也就是说，某单行线只允许单方向的车辆通过，在单行线的入口设置信号量 y ，在告示牌上显示某一时刻各方向来车的数量 $c1$ 和 $c2$ ，要修改告示牌上的车辆数量必须互斥进行，为此设置信号量 $x1$ 和 $x2$ 。若某方向的车辆需要通过时，首先要将该方向来车数量 $c1$ 或 $c2$ 增加 1，并查看自己是否是第一个进入单行线的车辆，若是，则获取单行线的信号量 y ，并进入单行线。通过此路段以后出单行线时，将该方向的车辆数 $c1$ 或 $c2$ 减 1（当然是利用 $x1$ 或 $x2$ 来互斥修改），并查看自己是否是最后一辆车，若是释放单行线的互斥量 y ，否则保留信号量 y ，让后继车辆继续通过。双方的操作如出一辙。考虑出现一个极端情况，即当某方向的车辆首先占据单行线并后来者络绎不绝时，另一个方向的车辆就再没有机会通过该单行线了。从而造成饥饿。由于有信号量的控制，死锁的可能性没有了（即双方同时进入单行线，在中间相遇，造成双方均无法通过的情景）。

27. D。

【解析】本题考查系统的安全状态和安全序列。当 Available 为 (2,3,3) 时，可以满足 P4,P5 中任一进程的需求；这两个进程结束后释放资源，Available 为 (7,4,11) 此时可以满足 P1,P2,P3 中任一进程的需求，故该时刻系统处于安全状态，安全序列中只有 D 满足条件。

28. D。

【解析】本题考查覆盖和交换的作用。内存管理是为了提高内存利用率，引入覆盖和交换技术，就是为了在较小的内存空间中用重复使用的方法来节省存储空间。覆盖和交换技术付出的代价是，需要消耗更多的处理机时间，它实际上是一种以时间换空间的技术。为此，从节省处理机时间来讲，换入、换出速度越快，付出的时间代价就越小，反之就越大，当时间代价大到一定程度时，覆盖和交换技术就没有意义了。

29. C。

【解析】本题考查请求分页系统的性能分析。设最大缺页中断率为 p ，则有效存储时间：
$$(1-p)*1us+0.3p*8us+0.7p*20us \leq 2us$$

即：
$$-p+2.4p+14p \leq 1$$

解得：
$$p \leq 6.5\%$$

30. B。

【解析】本题考查磁盘的性质。1569=512*3+33，故要访问字节位于第 4 个磁盘块上，对应的盘块号为 80。

31. C。

【解析】本题考查目录检索的原理。实现用户对文件的按名存取，系统先利用用户提供的文件名形成检索路径，对目录进行检索。在顺序检索中，路径名的一个分量未找到，说明路径名中的某个目录或文件不存在，就不需要继续再检索了。目录的查询方式有两种：顺序检索法和 Hash 法，通常还是采用顺序检索法，A 错误。在树型目录中，为了加快文件检索速度，可设置当前目录，于是文件路径可以从当前目录开始查找，B 错误。在顺序检索法查找完成后，得到的是文件的逻辑地址，D 错误。

32. C。

【解析】本题考查通道控制方式。CPU 启动通道时不管启动成功与否，通道都要回答 CPU，通过通过执行通道程序来实现数据的传送。通道在执行通道程序时，CPU 与通道并行，当通道完成通道程序的执行（即数据传送结束），便产生 I/O 中断向 CPU 报告。

33. B。

【解析】本题考查 OSI 参考模型各层的特点和功能。解题时，应注意题干中隐含的协议数据单元 PDU，以及各层次特定的功能。题干中的“二进制信息块”实际上就是指数据链路层封装的帧，数据链路层的可靠传输协议能够提供可靠传输服务。虽然传输层也能提供可靠传输服务，但它的可靠传输服务是可选的，而且它的 PDU 是报文。

34. D。

【解析】本题考查香农定理的应用。在一条带宽为 W Hz、信噪比为 S/N 的有噪声信道的最大数据传输率 V_{\max} 为 $W \log_2(1+S/N)$ bps。

先计算信噪比 S/N ：由 $30\text{db}=10 \log_{10} S/N$ ，得 $\log_{10} S/N = 3$ ，所以 $S/N=10^3=1000$ 。

计算 V_{\max} ： $V_{\max}=W \log_2(1+S/N)$ bps= $4000 \log_2(1+1000)$ bps $\approx 4000 \times 9.97$ bps $< 40\text{Kbps}$ 。

35. B。

【解析】本题考查二进制指数退避算法。如果发生冲突，采用该算法需要从 $[0, 1, 2, \dots, (2^k-1)]$ 中随机选取一个数，记为 r 。重传应推后的时间就是 r 倍的争用期。而上面所述的 k 值即为重传次数，但不应该超过 10。即： $k=\min[10, \text{重传次数}]$ 。在本题中重传次数为 4（第 5 次碰撞前则一共发送了 5 次，第一次以后的均为重传次数），因此本题答案为 $1/2^4=1/16$ 。

36. D。

【解析】本题考查路由选择的原理。对于该问题，我们可以从路由协议的原理以及路由表的构成上来考虑。

对于 IP 网络，是采用数据报方式，因此对于源主机和中途路由器都不会知道数据报经过的完整路径，路由器仅知道到达目的地址的下一跳地址（由路由表亦可知），主机仅知道到达本地网络的路径，到达其他网络的数据报均转发到路由器。

37. B

【解析】目的地址 195.26.17.4 转换为二进制的表达方式为：11000011.00011010.00010001.00000100。对该 IP 取 20、21、22 位的子网掩码，就可以得到该 IP 所对应的子网：195.26.16.0/20、195.26.16.0/21、195.26.16.0/22。从而可以得出该地址属于 195.26.16.0/20 的子网。

38. A。

【解析】ARP 请求分组是广播发送的，但 ARP 响应分组却是普通的单播，即从一个源地址发送到另一个目的地址。

39. C。

【解析】路由器是第三层设备，要处理的内容比第二层设备交换机要多，因而转发速度比交换机慢。虽然一些路由协议可以将延迟作为参数进行路由选择，但路由协议使用最多的参数是传输距离，此外还有其他的一些参数。路由器只能根据 IP 地址进行转发。

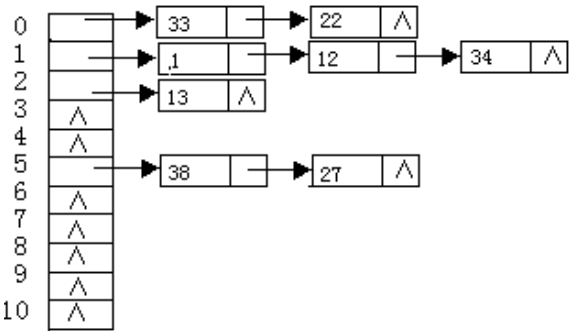
40. D

【解析】本题考查 TCP 协议的流量控制方式。TCP 协议采用滑动窗口机制来实现流量控制，同时根据接收端给出的接收窗口的数值发送方来调节自己的发送窗口，即使用可变大小的滑动窗口协议。

二、综合应用题

41. 解析：

(1) 采用链地址法构造散列表时，在直接计算出关键字对应的哈希地址后，将关键字结点插入到此哈希地址所在的链表中。由 $\text{hashf}(x)=x \bmod 11$ 可知，散列地址空间是 0 到 10。链地址法构造的表如下：



(2) 在链地址表中查找成功时，查找关键字为 33 的记录需进行 1 次探测，查找关键字为 22 的记录需进行 2 次探测，依此类推。因此：

$$ASL_{成功}=(1\times4+2\times3+3)/8=13/8$$

查找失败时，假设对空结点的查找长度为 0，则对于地址 0，查找失败的探测次数为 2；对于地址 1，查找失败的探测次数为 3，则平均探查长度为：

$$ASL_{失败}=(2+3+1+0+2+0+0+0+0+0)/11=8/11。$$

(3) 由 (1) 可知，查找关键字 34，需要依次与关键字 1,12,34 进行比较。

【扩展】对同样一组关键字，设定相同的散列函数，则不同处理冲突方法将得到不同的散列表，它们的平均查找长度也不同，本题若采用线性探查法处理冲突，题目应如何解答？

42. 解析：

(1) 算法基本设计思想：

①把数组的前 m 个元素看成一个归并段，后 n 个元素看成一个归并段，增加一个临时数组 B[1..m+n]存储临时归并结果。分别设置两个指针 k1, k2，指向两个归并段首元素，再设置一个指针 k3 指向临时数组下一个结果位置。

②如果 $1\leq k1\leq m$ 而且 $m+1\leq k2\leq m+n$ 执行③；否则执行④。

③比较两个归并段指针所指元素的大小。如果 $A[k1]\leq A[k2]$ ，那么 $B[k3++]=A[k1++]$ ；否则 $B[k3++]=A[k2++]$ 。执行②。

④如果 $k1>m$ ，则第二个归并段的元素还未比较完，把第二个归并段的剩余元素复制到数组 B。如果 $k2>m+n$ ，则第一个归并段的元素还未比较完，把第一个归并段的剩余元素复制到数组 B。最后把数组 B 复制到数组 A。

(2) 算法的实现如下:

```
void Merge(int[] A) {           //实现数组 1-m 和 m+1-m+n 两个归并段归并。
    int B[m+n+1];              //临时辅助数组 B[1...m+n]
    int k1,k2,k3;
    k1=1;k2=m+1;k3=1;          //3 个指针
    while(k1<=m&&k2<=m+n) { //如果两个归并段都没有比较完
        if(A[k1]<=A[k2]) B[k3++]=A[k1++]; //第一个归并段指针指向元素较小
        else B[k3++]=A[k2++];           //第二个归并段指针指向元素较小
    }
    if(k1>m)                      //把没有比较完的归并段中的剩余元素复制到数组 B
        while(k2<=m+n) B[k3++]=A[k2++];
    else
        while(k1<=m) B[k3++]=A[k1++];
    for(int i=1;i<=m+n;i++)        //把临时数组 B 复制到 A 数组
        A[i]=B[i];
}
```

(3) 总共遍历了 A 数组两遍, 第一遍合并, 第二遍复制结果, 时间复杂度为 $O(m+n)$ 。
临时数组 B 空间大小 $m+n$, 所以空间复杂度为 $O(m+n)$ 。

【解析 2】(1) 算法的基本设计思想:

将数组 A 看成是两个长度分别为 m 和 n 的有序表 L1 和 L2, 只需要将 L2 中的每个元素依次向前插入到前面有序数组部分中的合适位置即可。插入过程如下:

①取表 L2 中的第一个元素 $A[m+1]$, 暂存在 temp 中, 让 temp 前插到合适的位置。

②重复过程①, 继续插入 $A[m+2], A[m+3], \dots, A[m+n]$, 直到数组 A 整体有序。

(2) 算法的实现如下:

```
void InsertSort(int A[],int m,int n){
    int temp;                    //辅助变量, 暂存待插入元素
    for(int i=m+1;i<=m+n;i++){ //将 L2 中的元素依次插入到前面有序部分
        temp=A[i];
        for(int j=i-1;j>=1&&temp<A[j];j--) //向前查找待插入位置
            A[j+1]=A[j];                //向后挪位
        A[j+1]=temp;                    //复制到插入位置
    }
}
```

(3) 本算法的时间复杂度由 m 和 n 共同决定, 最内层循环的 $A[j+1]=A[j]$ 为基本语句。在最坏情况下, 即 L2 中的所有元素均小于 L1 中的最小元素, 则对于 L2 中的每个元素, 为了找到其插入位置都需要做 m 次移动, 故时间复杂度为 $O(mn)$ 。空间复杂度为 $O(1)$ 。

43. 解析:

本题考查补码的机内表示、补码的运算和溢出判断。

(1) 因 $x=-68=-(100\ 0100)_2$, 则 $[-68]_{\text{补}}=1011\ 1100=\text{BCH}$; 因 $y=-80=-(101\ 0000)_2$, 则 $[-80]_{\text{补}}=1011$

0000=B0H，所以寄存器 A 和 B 中的内容分别是 BCH、B0H。

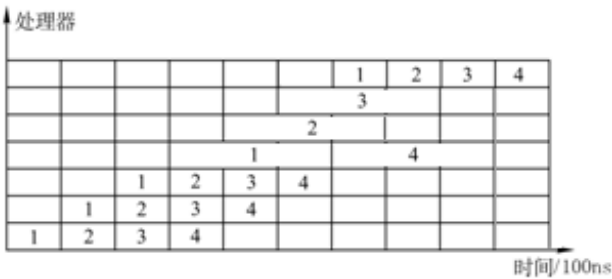
(2) $[x+y]_{补}=[x]_{补}+[y]_{补}=1011\ 1100+1011\ 0000=1\ 0110\ 1100=6CH$ ，所以寄存器 C 中的内容是 6CH，其真值为 108。此时，溢出标志位 OF 为 1，表示溢出，即说明寄存器 C 中的内容不是真正的结果；符号标志位 SF 为 0，表示结果为正数(溢出标志为 1，说明符号标志有错)；进位标志位 CF 为 1，仅表示加法器最高位有进位，对运算结果不说明什么。

(3) $[x-y]_{补}=[x]_{补}+[-y]_{补}=1011\ 1100+0101\ 0000=1\ 0000\ 1100=0CH$ ，最高位前面的一位被丢弃(取模运算)，结果为 12，所以寄存器 D 中的内容是 0CH，其真值为 12。此时，溢出标志位 OF 为 0，表示不溢出，即：寄存器 D 中的内容是真正的结果；符号标志位 SF 为 0，表示结果为正数；进位标志位 CF 为 1，仅表示加法器最高位有进位，对运算结果不说明什么。

44. 解析：

本题考查用时空图描述流水线的工作过程和流水线性能的计算方法。本题中的流水线使用重复设置瓶颈段的方法来消除瓶颈。B1、B2 和 B3 段是本题的关键，分为 3 条路径，每条都是 300ns，完全可以满足流水线的输入。

(1) 在流水线的 B 段，可以同时并行执行 3 条指令。流水线的时空图如下所示。



(2) 完成 4 个任务的周期数为 $T=(100+100+100+300+100+300)ns=1000ns$ ；任务数为 $N=4$ ；则有吞吐率为：

$$TP=N/T=(4/1000)\times10^9=0.4\times10^7 \text{ (条指令/秒)}$$

流水线的效率为：

45. 解析：

本题考查PV操作实现进程的互斥。

(1) 哥哥存两次钱后，共享变量amount的值为20。哥哥的第三次存钱与弟弟的取钱同时进行，如果两者顺序执行，则最后amount的值为20；如果在一个进程的执行过程中，进行CPU调度，转去执行另一进程，则最后amount的值取决于amount=m1及amount=m2的执行先后次序，若前者先执行，则最后amount的值为10，若后者先执行，则最后amount的值为30。因此，最后账号amount上面可能出现的值有10、20、30。

(2) 在上述问题中，共享变量amount是一个临界资源，为了实现两并发进程对它的互斥访问，可为它设置一初值为1的互斥信号量mutex，并将上述算法修改为：

```
int amount=0;

semaphore mutex=1;          //互斥访问amount变量的信号量
cobegin{
    process SAVE(){
```

```
int m1;
P(mutex);
m1=amount;
m1=m1+10;
amount=m1;
V(mutex);
}
process TAKE() {
int m2;
P(mutex);
m2=amount;
m2=m2-10;
amount=m2;
V(mutex);
}
} coend
```

46. 解析：

本题考查磁道的性能指标和特点。

(1) 读一个扇区的平均等待时间为旋转半周的时间，即为 $(60/5400)/2=5.55\text{ms}$ ，传输时间为 $(60/5400)/63=0.18\text{ms}$ ，因此读一个扇区的平均时间为 $5.55+0.18+10=15.73\text{ms}$ 。

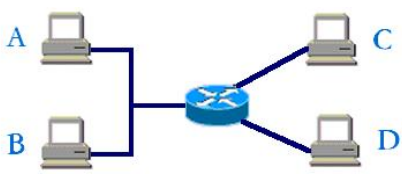
(2) 换出页时间为 15.73ms ，换入页时间 $1+5.55+0.18=6.73$ ，传输这两个页的平均时间为 $6.73+15.73=22.46\text{ms}$ 。

(3) 可能会产生两个后果，第一个后果是“饥饿”，这是由于请求磁盘I/O操作的应用程序得不到满足而长时间在阻塞队列等待，从而导致“饥饿”；第二个后果是“抖动”，由于每次磁盘I/O操作完成后，都要进行磁盘的换入换出，从而导致“抖动”。

47. 解析：

本题考查 IPv4 的地址特点、子网划分方法及网络设备的应用。子网掩码为 255.255.255.224，仅和第四字节有关，转换为二进制 255.255.255.11100000。把主机的地址转换为二进制，并和子网掩码做“与”运算，就可求出其网络地址。

(1) 只有处于同一个网络的主机之间才能直接通信，因此，只有主机 A 和主机 B 之间才可以直接通信。主机 C 和主机 D，以及它们同 A 和 B 的通信必须经过路由器。



A	主机地址	192.155.28.112	子网地址	192.155.28.96
B	主机地址	192.155.28.120	子网地址	192.155.28.96
C	主机地址	192.155.28.135	子网地址	192.155.28.128

D 主机地址 192.155.28.202 子网地址 192.155.28.192

(2) 若要加入第 5 台主机 E，使它能与 D 直接通信，那么主机 E 必须位于和 D 相同的网络中，即 192.155.28.192，这样地址范围是 192.155.28.193 到 192.155.28.222，注意要除掉 192.155.28.202。

(3) 主机 A 地址改为 192.155.28.168，那么它所处的网络为 192.155.28.160。由定义，直接广播地址是主机号各位全为“1”，用于任何网络向该网络上所有的主机发送报文，每个子网的广播地址则是直接广播地址。本地广播地址，又称为有限广播地址，它的 32 位全为“1”，用于该网络不知道网络号时内部广播。因此，主机 A 的直接广播地址为 192.155.28.191，本地广播地址是 255.255.255.255，若使用本地广播地址发送信息，所有主机都能够收到。

(4) 若希望 4 台主机直接通信，可以修改子网掩码为 255.255.255.0，这样 4 台主机就处于一个网络中，可以直接通信。

王道模拟试题（七）答案

一、单项选择题

1. B。

【解析】考查队列的应用。图的广度优先搜索类似于树的层序遍历（形象地想象成一个扇形，以搜索起点为中心，逐层向相连的外圈搜索），同样需要借助于队列。前序遍历二叉树是一个递归的过程，通常可以借助于栈，将递归算法转换为非递归算法。图的深度优先搜索类似于树的前序遍历，也是一个递归的过程，通常也可以借助栈来实现。

2. C。

【解析】考查出入栈序列。对于 A，可能的顺序是：1 入,1 出,2 入,2 出,3 入,3 出,4 入,4 出。对于 B，可能的顺序是：1 入,2 入,3 入,3 出,2 出,4 入,4 出,1 出。对于 D，可能的顺序是：1 入,1 出,2 入,3 入,3 出,2 出,4 入,4 出。C 则没有对应的序列。

【另解】对于 D 选项，p2 为最后一个入栈元素 4，则只有 p1 或 p3 出栈的元素有可能为 3（请读者分两种情况自行思考），而 p4 绝不可能为 3。读者在解答此类题时，一定要注意出栈序列中的“最后一个入栈元素”，这样可以节省答题的时间。

3. B。

【解析】考查二叉排序树的性质。在二叉排序树的存储结构中，每个结点由三部分构成，其中左（或右）指针指向比该结点的关键字值小（或大）的结点。关键字值最大的结点一定位于二叉排序树的最右位置上，因此它的右指针一定为空。

4. B。

【解析】考查森林与二叉树的转换。将这四棵树转换为二叉树后，第一棵树的根结点变成二叉树的根结点，第二棵树的根结点变成了根结点的右孩子，第二棵树中剩下的结点变成了其根结点的左子树。

5. D。

【解析】考查各种遍历算法的特点。先序、中序和后序遍历算法访问叶结点的顺序都一样，而层序遍历算法在二叉树的叶结点不在同一层上时，可能先遍历后面的叶结点。因此选 D。

6. A

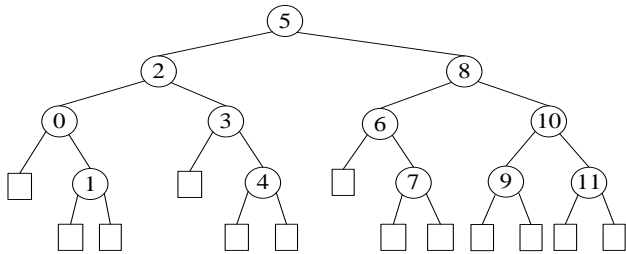
【解析】考查中序遍历。根据中序遍历的定义可知，在输出根结点后，才去中序递归地遍历根结点的右子树，因此根结点右边只有右子树上的所有结点。

7. B。

【解析】考查图的生成树。 n 个顶点的生成树是具有 $n-1$ 条边的极小连通子图， n 个顶点构成的环具有 n 条边，去掉任一条边后剩下的图依然连通的。因为 n 个顶点构成的环共有 n 条边，去掉其中任意一条便是一棵生成树，共有 n 种情况，所以可以有 n 棵不同的生成树（如，以 $n=3$ 为例）。

8. A。

【解析】考查折半查找的平均查找长度。假设有有序表中元素为 $A[0 \dots 11]$ ，不难画出对它所对应的折半查找判定树如下图所示，圆圈是查找成功结点，方形是虚构的查找失败结点。从而可以求出查找成功的 $ASL = (1 + 2 \times 2 + 3 \times 4 + 4 \times 5) / 12 = 37 / 12$ ，查找失败的 $ASL = (3 \times 3 + 4 \times 10) / 13$ 。

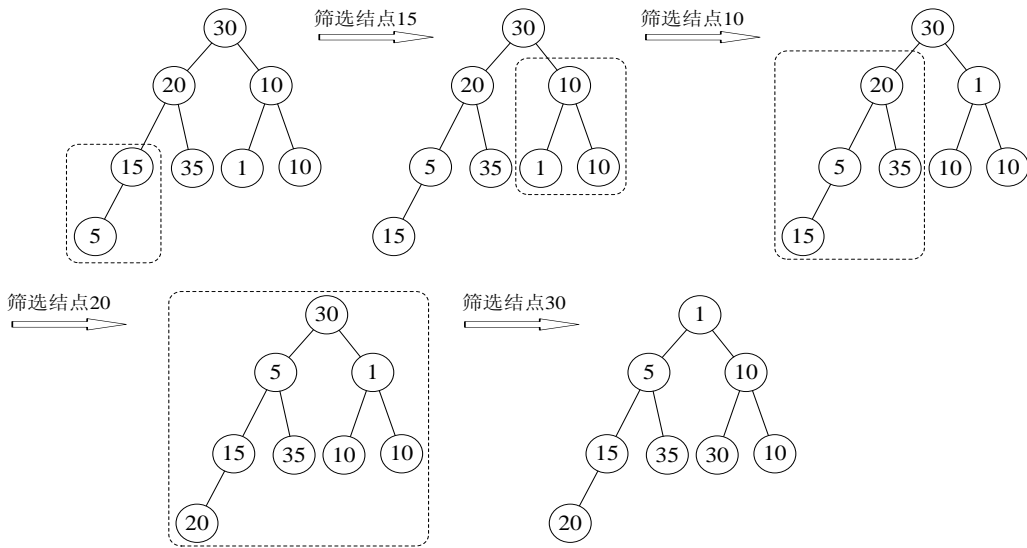


9. C。

【解析】考查 B 树的形状与磁盘读取次数的关系。在 B 树上进行查找需比较的结点数最多为 B 树的高度，B 树的高度与 B 树的阶 m 和关键字总数 n 有关。根据 B 树的定义，第一层（根结点）至少有 1 个结点；第二层至少有 2 个结点；除根结点以外的每个非终端结点至少有 $\lceil m/2 \rceil$ 棵子树，则第三层至少有 $2\lceil m/2 \rceil$ 个结点……第 $h+1$ 层至少有 $2(\lceil m/2 \rceil)^{h-1}$ 个结点，注意到第 $h+1$ 层是不包含任何信息的叶结点。对于关键字数为 n 的 B 树，叶结点即查找不成功的结点为 $n+1$ ，由此有 $n + 1 \geq 2(\lceil m/2 \rceil)^{h-1}$ ，即 $h \leq \log_{\lceil m/2 \rceil}((n + 1)/2) + 1$ 。

10. D。

【解析】考查各种排序算法的排序过程。观察序列变化，发现第 1 趟排序序列位置变化很大，所以不可能是选择排序和归并排序。又发现第 2 趟排序 15 和 20 交换了位置，所以不可能是希尔排序。只可能是快速排序。



11. B。

【解析】考查归并排序的执行过程。第一趟归并时，将每个关键字看成一个有序表，两两进行归并；第二趟归并时，将第一趟结果的 5 个长度为 2 的有序表归并，得到 2 个长度为

4 的有序表和 1 个长度为 2 的有序表。由于这里是采用 2-路归并，而且是第二趟排序，所以每 4 个元素放在一起归并，可将序列划分为{25,50,15,35}，{80,85,20,40}和{36,70}，分别对它们进行排序为{15,25,35,50}，{20,40,80,85}和{36,70}。

12. B。

【解析】本题考查根据时钟频率、指令条数和 CPI 来计算程序执行时间。程序的执行时间=(指令条数×CPI)/主频= $1.2 \times 4 \times 10^9 / 2\text{GHz} = 2.4\text{s}$ ，所占百分比为 $(2.4/4) \times 100\% = 60\%$ 。

13. D。

【解析】本题考查 IEEE754 标准的浮点数。在单精度浮点数中，最高位为数符位；其后是 8 位阶码，以 2 为底，用移码表示，阶码的偏置值为 127；其后 23 位是尾数数值位。对于规格化的二进制浮点数，数值的最高位总是“1”，为了能使尾数多表示一位有效值，将这个“1”隐含，因此尾数数值实际上是 24 位。隐含的“1”是一位整数。在浮点格式中表示出来的 23 位尾数是纯小数，用原码表示。41A4C000H 写成二进制为 0100 0001 1010 0100 1100 0000 0000 0000，第一位为符号位 0，表示是正数。之后的 8 位 1000 0011 表示阶码，真值为 $(100)_B$ ，即 4。剩下的是隐含了最高 1 的尾数，故而为 1.010 0100 1100 0000 0000 0000，数值左移四位后整数部分 10100 表示为 20。

注意：在 IEEE754 中，单精度浮点数（float）与双精度浮点数（double）都采用隐含尾数最高数位的方法，故可多表示一位尾数。临时浮点数又称为扩展精度浮点数，无隐含位。

14. D。

【解析】考查补码和浮点数运算的特点。补码定点运算，符号位参与运算，I 显然错误。浮点数的正负由尾数的符号决定，而阶码决定浮点数的表示范围，当阶码为负数时，浮点数小于 1，IV 错误。浮点数作加减运算时，尾数进行的是加减运算，V 错误。正确的选项为 II 和 III，故选 D。

15. B。

【解析】本题考查存储芯片的扩展。RAM区的地址范围为：0000 1000 0000 0000 0000 ~ 1111 1111 1111 1111 1111，由此可知RAM区的大小为 $31 \times 32\text{KB}$ ， $(31 \times 32\text{KB}) / 16\text{KB} = 62$ 。

16. D。

【解析】本题考查Cache命中率的相关计算。设Cache命中率为a，则 $(1000+100)(1-a)+100a \leq 115$ ，解得 $a \geq 0.985$ ，故至少为99%。

注意：虽然也可以采用同时访问Cache和主存的方式，此时不命中的访问时间为1000ns，但若题设中没有说明，默认Cache不命中的时间为访问Cache和主存的时间之和。

17. D。

【解析】本题考查快表和慢表的关系。快表又称TLB，采用高速相联存储器来存储可能需要使用的页的对应表项。而慢表存储在内存中。快表采用的是相联存储器，而不是依赖搜索算法来查找的，慢表通常是依赖于查找算法，故A和B错误。快表的命中率有可能高于慢表，但快表仅是慢表的一个部分拷贝，不能够得到比慢表更多的结果，因此C错误。

18. D。

【解析】本题考查 CALL 指令的执行。执行子程序调用 CALL 指令时，需要将程序断点即 PC 的内容保存在栈中，然后将 CALL 指令的地址码送入 PC。取出 CALL 指令后，PC 的值加 2 变为 10002H，CALL 指令执行后，程序断点 10002H 进栈，此时 SP=00FFH，栈顶内容为 1002H。

注意：PC 自增的数量，取决于指令长度。

19. C。

【解析】本题考查微程序方式的工作原理。当执行完公共的取指令微操作（送至指令寄存器 IR）后，由机器指令的操作码字段形成其对应微程序的入口地址。

20. D。

【解析】本题考查 PCI 总线。PCI 的特点主要有：与 CPU 及时钟频率无关；即插即用；采用猝发传送方式；扩展性好，可以采用多级 PCI 总线，可知 I 和 II 正确、IV 错误。总线连接的既然有主设备，就肯定有从设备，从设备主设备并不是固定的，III 错误。

注意：PCI 总线是常见的总线标准，如声卡、显卡、网卡等常用的插口。

21. A。

【解析】考查 DMA 方式中的中断与中断传输方式的区别。前者是向 CPU 报告数据传输结束，后者是传送数据，另外 DMA 方式中的中断不包括检查是否出错，而是报告错误。

22. D。

【解析】考查通道的工作过程。通道的基本工作过程（以一次数据传送为例）如下：

- ① 在用户程序中使用访管指令进入操作系统管理程序，由 CPU 通过管理程序组织一个通道程序，并使用 I/O 指令启动通道（此后 CPU 并行运行应用程序）。
- ② 通道处理器执行 CPU 为其组织的通道程序，完成指定的数据的输入/输出工作。
- ③ 通道程序结束后，向 CPU 发出中断请求。CPU 响应此中断请求后，第二次进入操作系统，调用管理程序对输入/输出中断进行处理。

23. C。

【解析】本题考查中断的处理过程和作用。当中断或异常发生时，通过硬件实现将运行在用户态的 CPU 立即转入到核心态。中断发生时，若被中断的是用户程序，系统将从目态转入管态，在管态下进行中断的处理；若被中断的是低级中断，则仍保留在管态，而用户程序只能在目态下运行，因此进入中断处理的程序只能是 OS 程序。

24. B。

【解析】本题考查信号量机制的应用。申请资源用 P 操作，执行完后若 $S < 0$ 时，表示资源申请完毕，需要等待， $|S|$ 表示等待该资源的进程数；释放资源用 V 操作，当 V 操作后， S 仍 ≤ 0 。在某时刻，信号量 S 的值为 0，然后对信号量 S 进行了 3 次 V 操作，即 $S = S + 3$ ，此时 $S > 0$ ，表示没有进程在队列中等待。

【注意】之前对 S 进行了 28 次 P 操作和 18 次 V 操作，并不会影响到计算的结果。

25. C。

【解析】本题考查并发进程的特点，并结合信号量进行同步的原理。由于进程并发，所以进程的执行具有不确定性，在 P1、P2 执行到第一个 P、V 操作前，应该是相互无关的。现

在考虑第一个对 s1 的 P、V 操作, 由于进程 P2 是 P(s1)操作, 所以它必须等待 P1 执行完 V(s1)操作以后才可继续运行, 此时的 x、y、z 值分别是 3,3,4, 当进程 P1 执行完 V(s1)以后便在 P(s2)上阻塞, 此时 P2 可以运行直到 V(s2), 此时的 x、y、z 值分别是 5,3,9, 进程 P1 继续运行直到结束, 最终的 x、y、z 值分别为 5,12,9。

26. B。

【解析】本题考查银行家算法。安全性检查一般要用到进程所需的最大资源数, 减去进程占用的资源数, 得到进程为满足进程运行尚需要的可能最大资源数, 而系统拥有的最大资源数减去已分配掉的资源数得到剩余的资源数, 比较剩余的资源数是否满足进程运行尚需要的可能最大资源数就可以得到当前状态是否安全的结论。而满足系统安全的最少资源数并没有这么一个说法。

27. C。

【解析】本题考查虚拟分页存储管理中缺页中断的处理过程。在内存管理中, 一定要特别注意区分基本分页与请求分页、基本分段与请求分段的管理方式下, 具体的地址变换过程。缺页中断的处理流程为: 产生缺页中断后; 首先去内存寻找空闲物理块, 若内存没有空闲物理块, 使用页面置换算法决定淘汰页面, 然后调出该淘汰页面; 最后再调入该进程欲访问的页面。整个流程可归纳为: 缺页中断->决定淘汰页->页面调出->页面调入。

28. D。

【解析】考查内存保护。在地址变换过程中, 可能会因为缺页、操作保护和越界保护而产生中断, 但肯定不会发生内存溢出(内容容量不足)的现象, 故 IV 错误。

29. B。

【解析】本题考查虚拟页式存储管理中多级页表的计算。由题中所给的条件, 虚拟地址空间是 2^{48} , 即没有完全使用 64 位地址。页面大小为 2^{13} , 即 8KB, 则用于分页的地址线的位数为 $48-13=35$ 。下面计算每一级页表能容纳的最多数量。由题意, 每个页面为 8KB, 每个页表项为 8 字节, 那么一页中能容纳的页表项为 $8KB/8B=1K$, 即 1024 个页表项, 可以占用 10 位地址线来寻址, 故剩余的 35 位地址线可以分为 $35/10=3.5$, 向上取整为 4, 因此至少 4 级页表才能完成此虚拟存储的页面映射。

30. C。

【解析】本题考查文件的物理结构。第 22 个逻辑记录存放在第 5 个物理块中($22 \times 100 / 512 = 4$, 余 152), 由于文件采用的物理结构是链接文件, 因此需要从第一个物理块开始读取, 共需启动磁盘 5 次。

31. A。

【解析】本题考查设备管理的知识点。通道是一种硬件、或特殊的处理器, 它有自身的指令, 故 I 错误。通道没有自己的内存, 通道指令存放在主机的内存中, 也就是说通道与 CPU 共享内存, 故 II 正确。为了实现设备独立性, 用户使用逻辑设备号来编写程序, 故 III 错误。来自通道的 I/O 中断事件是属于输入/输出的问题, 故应该由设备管理负责, 故 IV 正确。综上所述, I、III 错误。

注意: 通道作为一种特殊的硬件或者处理器, 具有诸多特征, 它与一般处理器的区别、

以及与 DMA 方式的区别要认真理解。

32. B。

【解析】本题考查网络协议中的基本概念。协议、服务、对等层等基本概念是 OSI 参考模型的重要内容，与分层体系结构的思想相互渗透。对等层是指计算机网络协议层次中，将数据“直接”传递给对方的任何两个同样的层次，因此，对等层之间的通信必须有对等层之间的协议。选项 A 是相邻层之间通信所必需的。上层使用下层所提供的服务必须通过与下层交换一些命令，这些命令在 OSI 中称为服务原语，C 错误。选项 D 是物理层的内容。

33. A。

【解析】本题考查了有关 GBN 协议的相关机制问题。在 GBN 协议中，接收窗口尺寸被定为 1，从而保证了按序接收数据帧。如果接收窗口内的序号为 4 时，此时接收方需要接收到的帧即为 4 号帧，即便此时接收到正确的 5 号帧接收端也会自动丢弃该帧从而保证按序接收数据帧。注意，GBN 协议中接收端是没有缓存的，所以也不存在将 5 号帧缓存下来的说法。

34. B。

【解析】本题考查 CSMA/CD 协议的碰撞检测原理。最短帧长等于在争用期时间内发送出的比特数。站点在发送帧后至多经过 2τ (争用期) 就可以知道所发送的帧是否遭到了碰撞。因此，最小帧长=总线传播时延 \times 数据传输速率 $\times 2$ 。当传输速率提高时，为了有效地检测冲突，可采用减少电缆介质的长度，使争用期时间减少（即以太网端到端的时延减小），保持最小帧长不变；或增加最短帧长。

35. D。

【解析】本题考查 IP 分组的首部字段含义。如果题目没有说明不考虑 NAT，都认为源目的 IP 地址和目的 IP 地址都是可以改变的，否则都是不能改变的。A 选项：当 IP 分组的长度超过该网络的 MTU 时需要分片，总长度将改变，故 A 错误；B 选项：IP 分组每经过一跳，都会改变其首部检验和，故 B 错误；C 选项：每经过一个路由器，生存时间减 1，故 C 错误；D 选项：在不考虑 NAT 时，源 IP 地址和目的 IP 地址都不会变化。

36. A。

【解析】本题考查路由聚合的计算。路由聚合将网络前缀都相同的连续的 IP 地址组成一个地址块，它可以包括多个 A、B、C 类网络，并在网络地址后面指明网络前缀的位数。CIDR 虽然不使用子网但分配到一个 CIDR 地址块的组织，但仍然可以在本组织内根据需要划分出一些子网。由于前两个字节和最后一个字节都一样，则只比较第三个字节即可，转化为二进制：129 \rightarrow 10000001；130 \rightarrow 10000010；132 \rightarrow 10000100；133 \rightarrow 10000101。显然，这四个数字只有前五位是完全相同的，因此汇聚后的网络的第三个字节应该是 10000000 \rightarrow 128。聚合后的网络的掩码中 1 的数量应该有 $8+8+5=21$ ，因此答案是 172.18.128.0/21。

37. B。

【解析】考查各种协议的应用。刚开机时 ARP 表为空，想需要和其他主机进行通信时，数据链路层需要使用 MAC 地址，因此就会用到 ARP 协议。在校园网访问因特网时，肯定会使用到 IP 协议。而此时访问的是因特网，因特网为外网，所以就需要通过 DHCP 分配公网地址。而 ICMP 协议主要用于发送 ICMP 差错报告报文和 ICMP 询问报文，则不一定会用到。

38. D。

【解析】本题考查 PDU 在对等层间的处理。PDU 中装载的是哪一层的数据，就由哪一层来处理该数据，而 PDU 所在的层只负责传输该数据。IP 网络是分组交换网络，每个分组的首部都包含了完整的源地址和目的地址，以便途经的路由器为每个 IP 分组进行路由，即便是同一个源站点向同一个目的站点发出的多个 IP 分组也并不一定走同一条路径，亦即这些 IP 分组到达目的站点的顺序可能不一定按序到达，目的站点的传输层必须进行排序；而一个较大的 IP 分组在传输的过程中，由于途经物理网络的 MTU 可能比较小，一个 IP 分组可能将分成若干个分组，每个分组都有完整的首部，与普通的 IP 分组没有区别地传输。按照网络对等层通信的原则，接收站点的网络层收到的 IP 分组必须与发送站点发送的 IP 分组相同，所以接收站点的网络层必须把沿途被分片的分组进行重组，还原成原来的 IP 分组。所以重组工作是由网络层完成的。

39. B。

【解析】本题考查对 TCP 协议的理解。TCP 是在不可靠的 IP 层之上实现可靠的数据传输协议，它主要解决传输的可靠、有序、无丢失和不重复的问题，其主要特点是：①TCP 是面向连接的传输层协议。②每一条 TCP 连接只能有两个端点，每一条 TCP 连接只能是端对端的（进程—进程）。③TCP 提供可靠的交付服务，保证传送的数据无差错、不丢失、不重复且有序。④TCP 提供全双工通信，允许通信双方的应用进程在任何时候都能发送数据，为此 TCP 连接的两端都设有发送缓存和接收缓存。⑤TCP 是面向字节流的，虽然应用程序和 TCP 的交互是一次一个数据块（大小不等），但 TCP 把应用程序交下来的数据看成仅仅是一连串的无结构的字节流。I：IP 协议才是点到点的通信协议（也说是主机—主机），而 TCP 是端到端的协议，故 I 错误；II：TCP 提供面向连接的可靠数据传输服务，故 II 错误；III：IP 数据报不是由传输层来组织的，而应该由网络层加上 IP 数据报的首部来形成 IP 数据报，故 III 错误；IV：前面已经分析，正确。综上，I、II 和 III 都是错误的。

40. C。

【解析】本题考查 WWW 高速缓存。WWW 高速缓存将最近的一些请求和响应暂存在本地磁盘中，当与暂存的请求相同的新请求到达时，WWW 高速缓存就将暂存的响应发送出去，从而降低了广域网的带宽。

二、综合应用题：第 41~47 题，共 70 分。

41. 解析：

(1) 该图对应的邻接矩阵如下：

∞	2	3	∞	∞	∞	∞	∞
∞	∞	∞	5	∞	∞	∞	∞
∞	∞	∞	3	10	∞	∞	∞
∞	∞	∞	∞	∞	4	∞	∞
∞	∞	∞	∞	∞	∞	3	∞
∞	∞	∞	∞	2	∞	∞	6
∞	∞	∞	∞	∞	∞	∞	1
∞	∞	∞	∞	∞	∞	∞	∞

(2) 只有顶点 V1 的入度为 0，由此可以得到两个拓扑序列：V1, V2, V3, V4, V6, V5, V7, V8 和 V1, V3, V2, V4, V6, V5, V7, V8。

(3) 关键路径共有 3 条，长 17。依次为：V1->V2->V4->V6->V8, V1->V3->V5->V7->V8, V1->V2->V4->V6->V5->V7->V8。

事件	V1	V2	V3	V4	V5	V6	V7	V8
最早发生时间	0	2	3	7	13	11	16	17
最晚发生时间	0	2	3	7	13	11	16	17

活动	V1-V2	V1-V3	V2-V4	V3-V4	V3-V5	V4-V6	V6-V5	V5-V7	V6-V8	V7-V8
最早开始时间	0	0	2	3	3	7	11	13	11	16
最晚开始时间	0	0	2	4	3	7	11	13	11	16
时间余量	0	0	0	1	0	0	0	0	0	0

(4) 顶点 V1 到其它各顶点的最短路径和距离为：2(V1->V2), 3(V1->V3), 6(V1->V3->V4), 12(V1->V3->V4->V6->V5), 10(V1->V3->V4->V6), 15(V1->V3->V4->V6->V5->V7), 16(V1->V3->V4->V6->V5->V7->V8 或 V1->V3->V4->V6->V8)。

42 . 解析：

(1) 基本的基本设计思想：

设置二叉树的平衡标记 **balance**，以标记返回二叉树 **bt** 是否为平衡二叉树，若为平衡二叉树，则返回 1，否则返回 0；h 为二叉树 **bt** 的高度。采用前序遍历的递归算法：

- ①若 **bt** 为空，则高度为 0，**balance=1**。
- ②若 **bt** 仅有根结点，则高度为 1，**balance=1**。

③否则，对 **bt** 的左、右子树执行递归运算，返回左、右子树的高度和平衡标记，**bt** 的高度为最高子树的高度加 1。若左、右子树的高度差大于 1，则 **balance=0**；若左、右子树的高度差小于 1，且左、右子树都平衡时，**balance=1**，否则 **balance=0**。

(2) 算法的实现如下：

```
void Judge_AVL(BiTree bt,int &balance,int &h){
    int bl,br,hl,hr;           //左、右子树的平衡标记和高度
    if(bt==NULL){              //空树，高度为 0
        h=0;
        balance=1;
    }
    else if(p->lchild==NULL&&p->rchild==NULL){ //仅有根结点，则高度为 1
        h=1;
        balance=1;
    }
    else{
        Judge_AVL(bt->lchild,bl,hl); //递归判断左子树
        Judge_AVL(bt->rchild,br,hr); //递归判断右子树
        h=(hl>hr?hl:hr)+1;
```

```
if(abs(hl,hr)<2)           //若高度绝对值小于2，则看左、右子树是否都平衡
    balance=bl&br;         //&为逻辑与，即左、右子树都平衡时，二叉树平衡
else
    balance=0;
}
```

43. 解析：

本题考查存储器的扩展。根据各个存储区所要求的容量和选定的存储芯片的容量，就可以计算出各种芯片的数目，即：总片数=总容量/每片的容量。

将多个芯片组合起来常采用位扩展法、字扩展法、字和位同时扩展法。位扩展是指只在位数方向扩展（加大字长），而芯片的字数和存储器的字数是一致的；字扩展是指仅在字数方向扩展，而位数不变，字扩展将芯片的地址线、数据线、读写线并联，由片选信号来区分各个芯片。本题需采用字和位同时扩展，即在字数方向和位数方向上同时扩展。

(1) 已知数据总线为 8 位，ROM 区为 3000H~4FFFFH，故 ROM 的容量为 8K×8；ROM 芯片数=8K×8/4K×2=8 片（分为 2 组，每组 4 片）。RAM 区为 5000H~67FFH，故 RAM 的容量为 6K×8；SRAM 芯片数=6K×8/2K×4=6 片（分为 3 组，每组 2 片）。

(2) ROM 芯片的容量为 4K×2，具有 12 根地址线、2 根数据线，因此 ROM 芯片的地址线连接 CPU 地址线的低 12 位 A₁₁~A₀，每组 ROM 内的 4 片芯片分别连接 CPU 数据线的 D₇D₆、D₅D₄、D₃D₂、D₁D₀。SRAM 芯片的容量为 2K×4，具有 11 根地址线、4 根数据线，因此 SRAM 芯片的地址线连接 CPU 地址线的低 11 位 A₁₀~A₀，每组 SRAM 内的 2 片芯片分别连接 CPU 数据线的 D₇D₆D₅D₄、D₃D₂D₁D₀。

(3) ROM 区有 2 个片选信号，RAM 区有 3 个片选信号，共需 5 个片选信号，根据地址分配的要求，各片选信号的逻辑表达式如下：

$$CS_0 = \overline{A_{15}} \overline{A_{14}} \overline{A_{13}} A_{12}, \quad CS_1 = \overline{A_{15}} A_{14} \overline{A_{13}} \overline{A_{12}}, \quad CS_2 = \overline{A_{15}} A_{14} \overline{A_{13}} A_{12} \overline{A_{11}}$$
$$CS_3 = \overline{A_{15}} A_{14} \overline{A_{13}} A_{12} A_{11}, \quad CS_4 = \overline{A_{15}} A_{14} A_{14} \overline{A_{12}} \overline{A_{11}}$$

44. 解析：

本题考查流水线的技术。编者以为今年组成原理部分的大题极有可能会来自中央处理器这章，而且考到流水线的概念很大，因此读者务必牢固掌握。

- (1) 流水线操作的时钟周期 t 应按四步操作中的最长时间来考虑，所以 t=100ns。
- (2) 两条指令发生数据相关冲突情况如下：

```
ADD R1,R2,R3           # R2+R3 -> R1
SUB R4,R1,R5           # R1-R5 -> R4
```

两条指令在流水线中执行情况如下表所示：

时钟 指令	1	2	3	4	5	6	7
ADD	取指	指令译码并取数	运算	写回			

SUB		取指	指令译码并取数	运算	写回		
-----	--	----	---------	----	----	--	--

ADD 指令在时钟 4 时将结果写入寄存器堆(R1),但 SUB 指令在时钟 3 时读寄存器堆(R1)。本来 ADD 指令应先写入 R1, SUB 指令后读 R1, 结果变成 SUB 指令先读 R1, ADD 指令后写 R1, 因而发生两条指令间数据相关。如果硬件上不采取措施, 第 2 条指令 SUB 至少应推迟 2 个操作时钟周期 (2×100ns), 即将 SUB 指令中的指令译码并取数阶段推迟到 ADD 指令的写回阶段之后才能保证不会出错。如下表所示:

时钟 指令	1	2	3	4	5	6	7
ADD	取指	指令译码并取数	运算	写回			
SUB		取指			指令译码并取数	运算	写回

(3) 如果硬件上加以改进, 可只延迟 1 个操作时钟周期 (100ns)。因为在 ADD 指令中, 运算阶段就已经得到结果了, 因此可以通过数据旁路技术在运算结果一得到的时候将结果快速送入寄存器 R1, 而不需要等到写回阶段完成。流水线中执行情况如下图所示:

时钟 指令	1	2	3	4	5	6	7
ADD	取指	指令译码并取数	运算（并采用数据旁路技术写入寄存器 R1）	写回			取指
SUB		取指		指令译码并取数	运算	写回	

45 . 解析：

 本题考查各种调度算法的执行以及性能分析。

(1) 采用先来先服务调度时，执行作业的次序为P₁、P₂、P₃、P₄、P₅，如下表所示。

作业号	就绪时刻	服务时间	等待时间	开始时刻	结束时刻	周期时间	带权周转时间
P ₁	0	3	0	0	3	3	3/3=1.0
P ₂	2	6	1	3	9	7	7/6=1.17
P ₃	4	4	5	9	13	9	9/4=2.25
P ₄	6	5	7	13	18	12	12/5=2.4
P ₅	8	2	10	18	20	12	12/2=6.0
平 均						8.6	2.56

(2) 采用短作业优先调度时，执行作业的次序为P₁、P₂、P₅、P₃、P₄，如下表所示。

作业号	就绪时刻	服务时间	等待时间	开始时刻	结束时刻	周期时间	带权周转时间
P ₁	0	3	0	0	3	3	3/3=1.0
P ₂	2	6	1	3	9	7	7/6=1.17

P ₅	8	2	1	9	11	3	3/2=1.5
P ₃	4	4	7	11	15	11	11/4=2.75
P ₄	6	5	9	15	20	14	14/5=2.8
平 均						7.6	1.84

(3) 采用高响应比优先调度时，响应比=(等待时间+服务时间)/运行时间。在时刻0，只有进程P₁就绪，执行P₁，在时刻3结束。此时只有P₂就绪，执行P₂，在时刻9结束。此时P₃、P₄、P₅均就绪，计算它们的响应比分别为2.25、1.6、1.5，则选择执行P₃，在时刻13结束。此时P₄、P₅均就绪，计算它们的响应比分别为2.4、3.5，则选择执行P₅，在时刻15结束。此时只有P₄就绪，执行P₄，在时刻20结束。整个执行作业的次序为P₁、P₂、P₃、P₅、P₄，如下表所示。

作业号	就绪时刻	服务时间	等待时间	开始时刻	结束时刻	周期时间	带权周转时间
P ₁	0	3	0	0	3	3	3/3=1.0
P ₂	2	6	1	3	9	7	7/6=1.17
P ₃	4	4	5	9	13	9	9/4=2.25
P ₅	8	2	5	13	15	7	7/2=3.5
P ₄	6	5	9	15	20	14	14/5=2.8
平 均						8.0	2.14

46 . 解析：

 本题考查文件目录的结构。

(1) 因为磁盘块大小为 512B，所以索引块大小也为 512B，每个磁盘地址大小为 2B。因此，一个一级索引表可容纳 256 个磁盘地址。同样地，一个二级索引表可容纳 256 个一级索引表地址，一个三级索引表可容纳 256 个二级索引表地址。这样，一个普通文件最多可有文件页数为 10+256+256×256+256×256×256=16843018 页。

(2) 由图可知，目录文件 A 和 D 中的目录项都只有两个，因此这两个目录文件都只占用一个物理块。要读文件 J 中的某一页，先从内存的根目录中找到目录文件 A 的磁盘地址，将其读入内存（已访盘 1 次）。然后从目录 A 中找出目录文件 D 的磁盘地址读入内存（已访盘 2 次）。再从目录 D 中找出文件 J 的文件控制块地址读入内存（已访盘 3 次）。在最坏情况下，该访问页存放在三级索引下，此时需要一级一级地读三级索引块才能得到文件 J 的地址（已访盘 6 次）。最后读入文件 J 中的相应页（共访盘 7 次）。所以，若要读文件 J 中的某一页，最多启动磁盘 7 次。

(3)由图可知，目录文件 C 和 U 的目录项较多，可能存放在多个链接在一起的磁盘块中。在最好情况下，所需的目录项都在目录文件的第一个磁盘块中。先从内存的根目录中找到目录文件 C 的磁盘地址读入内存（已访盘 1 次）。在 C 中找出目录文件 I 的磁盘地址读入内存（已访盘 2 次）。在 I 中找出目录文件 P 的磁盘地址读入内存（已访盘 3 次）。从 P 中找到目录文件 U 的磁盘地址读入内存（已访盘 4 次）。从 U 的第一个磁盘块中找出文件 W 的文件控制块地址读入内存（已访盘 5 次）。在最好情况下，要访问的页在文件控制块的前 10 个直接块中，按照直接块指示的地址读文件 W 的相应页（已访盘 6 次）。所以，若要读文件 W 中的某一页，最少启动磁盘 6 次。

(4) 为了减少启动磁盘的次数，可以将需要访问的 W 文件挂在根目录的最前面的目录项

中。此时，只需读内存中的根目录就可以找到 W 的文件控制块，将文件控制块读入内存（已访盘 1 次），最差情况下，需要的 W 文件的那个页挂在文件控制块的三级索引下，那么读 3 个索引块需要访问磁盘 3 次（已访盘 4 次）得到该页的物理地址，再去读这个页即可（已访盘 5 次）。此时，磁盘最多启动 5 次。

47. 解析：

（1）在使用 CIDR 时会有多个匹配结果，应从匹配结果选择具有最长网络前缀的路由。首先 142.150.0.0/16 和 142.150.71.132 是相匹配的，前面 16 位相同，下面分析其他项：

- ①142.150.64.0/24 和 142.150.71.132 不匹配，因为前 24 位不相同。
- ②142.150.71.128/28 和 142.150.71.132 的前 24 位是匹配的，只需看后面 4 位是否一样，128 的二进制为 1000 0000，132 的二进制为 1000 0100，前 4 位相同，故匹配了 28 位。
- ③142.150.71.128/30 和 142.150.71.132 的前 24 位是匹配的，但后面的 6 位中第 6 位不一样，故不匹配。

因此，根据最长网络前缀的匹配原则，应根据第 2 个路由表项转发，下一跳路由为 B。

（2）欲达到题目的要求，只需构造一个网络前缀和该地址匹配 32 位就行了，即针对 142.150.71.132 的特定主机路由，增加的表项为：网络前缀 142.150.71.132/32、下一跳 A。

（3）增加 1 条默认路由：网络前缀 0.0.0.0/0、下一跳 E。

（4）要划分成 4 个规模尽可能大的子网，则需要从主机位中划出 2 位作为子网位（ $2^2=4$ ，CIDR 广泛使用之后允许子网位可以全 0 和全 1）。

各子网网地址分别为：142.150.64.0000 0000；142.150.64.0100 0000；142.150.64.1000 0000；142.150.64.1100 0000。子网掩码应该为 255.255.255.192。可分配地址范围需将主机号中全 0 和全 1 的都去掉。因此各子网的地址分配方案如下：

子网地址：142.150.64.0/26	地址范围：142.150.64.1~142.150.64.62
子网地址：142.150.64.64/26	地址范围：142.150.64.65~142.150.64.126
子网地址：142.150.64.128/26	地址范围：142.150.64.129~142.150.64.190
子网地址：142.150.64.192/26	地址范围：142.150.64.193~142.150.64.254

王道模拟试题（八）答案

一、单项选择题

1. C.

【解析】考查栈的操作。初始时栈顶指针 $\text{top} = n+1$ ，所以该栈应该是从高地址向低地址生长。且 $n+1$ 不在向量的地址范围，因此应该先将 top 减 1，再存储。即选 C。

注意：对于顺序存储的栈（对于队列也类似），如果存储的定义不同，则出入栈的操作也不相同（并不是固定的），这要看栈顶指针指向的是栈顶元素，还是栈顶元素的下一位置。

2. B.

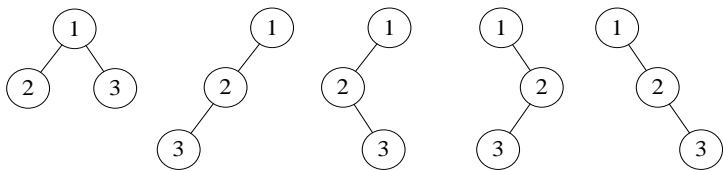
【解析】考查循环队列的插入和删除，头尾指针的变化。队列的特点是先进先出，队头删除元素，队尾插入元素。删除一个元素，队头指针 $\text{front} = (\text{front}+1) \bmod 6 = 4$ ，队尾指针不变。插入两个元素，队尾指针 $\text{rear} = (\text{rear}+2) \bmod 6 = 2$ ，队头指针不变。所以 rear 和 front 分别为 2 和 4，选 B。

3. B.

【解析】考查几种特殊二叉树的特点。二叉判定树描述了折半查找的过程，肯定是高度平衡的，因此不可能是 A。对于 B，此图中所有结点的关键值均大于左子树中结点关键值，且均小于右子树中所有结点的关键值，B 符合。对于 C，此图中存在不平衡子树，错误。对于 D，此图不符合小根堆或大根堆的定义。

4. D.

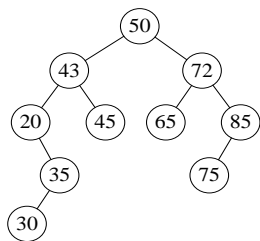
【解析】考查由遍历序列构造二叉树。由遍历序列构造二叉树的思想就是找到根结点，然后将序列划分成左、右子树，如此递归地进行下去。前序序列和中序序列、后序序列和中序序列、或中序序列和层序序列可唯一确定一个二叉树。先序序列和层序序列不能唯一的确定一棵二叉树，层序序列第 1 次访问根结点，先序序列为 NLR，虽然能找到根结点，但无法划分左、右子树。



如上图所示的 5 棵不同的二叉树，其对应的先序序列和层序序列是相同的。

5. B.

【解析】考查二叉排序树的构造和查找。按题中数据的输入次序，建立的二叉排序树如下图所示。查找元素 30 的比较次数为 5 次。



6. D。

【解析】考查图的基本性质。 n 个顶点构成连通图至少需要 $n-1$ 条边（生成树），但若再增加 1 条边，则必然会构成环。如果一个无向图有 n 个顶点和 $n-1$ 条边，可以使它连通但没有环（即生成树），但再加一条边，在不考虑重边的情形下，就必然会构成环。

7. C。

【解析】考查深度优先遍历。深度优先遍历是找到新的访问结点后，就从新结点开始找新的访问结点，如果没有找到，回溯到上一个找到的新的访问结点继续查找。从顶点 1 出发，下一个新访问结点 3，从 3 开始，找到 4，从 4 开始，没有新结点，回溯到 3，找到新访问结点 5，从 5 开始，找到 2，从 2 开始没有新结点，回溯到 5，没有新结点，回溯到 3，没有新节点，回溯到 1，没有新结点，访问结束。所以得到的顶点序列为 1,3,4,5,2。

8. B。

【解析】考查各种查找方法的特点。顺序查找平均查找长度的数量级是 $O(n)$ ；折半查找平均查找长度的数量级是 $O(\log_2 n)$ 。分块查找平均查找长度的数量级是 $O(\log_2 K + n/K)$ 。散列查找的平均查找长度跟装填因子和采用的冲突解决方法有关。二分查找树在最坏情况下的平均查找长度为 $O(n)$ ，但在关键字随机分布的情况下，用二分查找树的方法进行查找的平均查找长度的数量级为 $O(\log_2 n)$ 。

9. D。

【解析】考查堆排序的执行过程。筛选法初始建堆为 {8,17,23,52,25,72,68,71,60}，输出 8 后重建的堆为 {17,25,23,52, 60,72,68,71}，输出 17 后重建的堆为 {23,25,68,52,60,72,71}。建议读者在解题时画草图。

10. B。

【解析】考查快排过程。以 28 为基准元素，首先从后向前扫描比 28 小的元素，此元素位置为 L_0 ，把此元素放到前面基准元素位置，而后再从前向后扫描比 28 大的元素，此元素位置 L_1 ，并将其放到 L_0 位置，从而得到了 (5,16, L_1 ,12,60,2,32,72)。而后继续重复从后向前扫描，记录找到的比 28 小的元素位置 L_2 ，把此元素放到 L_1 ，再从前往后扫描的操作找到比 28 大的元素，此元素位置 L_3 ，并将其放到 L_2 位置，直到扫描到相同元素，一趟排序完毕。最后得到 (5,16,2,12)28(60,32,72)。

11. C。

【解析】考查各种排序算法的性质。插入排序为和选择排序的排序趟数始终为 $n-1$ ，与序列的初态无关。对于冒泡排序，如果序列初态基本有序，可以在一趟排序后检查是否有元素交换，如果没有说明已排好序，不用再继续排序。对于快速排序，每个元素要确定它的最终

位置都需要一趟排序，所以无论序列原始状态如何，都需要 n 趟排序，只不过对于不同的初态，每一趟处理的时间效率不同，初试状态约接近有序，效率越低。

12. C。

【解析】本题考查计算机的性能指标。整机的速度是由多个指标综合衡量的，某个指标的高低并不能完全决定机器的速度，故 A、B 错误。在一个用户程序执行过程中，可能会插入运行其他程序，所以观测到用户程序的执行时间要大于其真正的 CPU 执行时间，故 D 错误。在不同的程序中，各类指令所占的比例是有可能不同的，因此测得的 CPI 有可能不同，C 正确。

13. C。

【解析】考查规格化形式。基数为 4，尾数用原码表示，则小数点后面两位不全为 0 即为规格化数。

注意：对于基数为 4 的原码尾数，每右（或左）移 2 位，阶码加（或减）1。

14. A。

【解析】考查 IEEE754 单精度浮点数的表示。IEEE754 规格化单精度浮点数的阶码范围为 $1 \sim 255$ ，尾数为 1.f。最接近 0 的负数的绝对值部分应最小，因此尾数取 1.0；阶码取最小 1，故最接近 0 的负数为 $-1.0 \times 2^{1-127} = -2^{-126}$ 。即选 A。

15. C。

【解析】本题考查双口存储器和交叉存储器的特点。双端口 RAM 的两个端口具有 2 组相互独立的地址线、数据线和读写控制线，因此可以同时访问同一区间、同一单元，I 正确；当两个端口同时对相同的单元进行读操作时，则不会发生冲突，II 错误。高位多体交叉存储器由于是在单个存储器中字是连续存放的，所以不能保证程序的局部性原理；而低位多体交叉存储器由于是交叉存放，所以能很好的满足程序的局部性原理，III 错误。高位四体交叉存储器虽然不能满足程序的连续读取，但仍可能一次连续读出彼此地址相差一个存储体容量的 4 个字，只是这么读的概率较小，IV 正确。

注意：高位多体交叉存储器仍然是顺序存储器。

16. D。

【解析】考查存储器的多个知识点。虚拟存储器进行虚实地址转换，需要多次访存（先查找页表），增加了延迟，降低了计算机速度，I 错误。II 描述的是存取周期的概念，II 错误。Cache 有自己独立的地址空间，通过不同的映射方式映射到主存的地址空间，III 错误。主存也可以由 ROM 组成，如可用于部分操作系统的固话、自举程序等，IV 错误。

17. C。

【解析】本题考查各种寻址方式的原理。因此访问寄存器的速度通常访问主存的数十倍，因此获取操作数快慢主要取决于寻址方式的访存次数。立即寻址操作数在指令中，不需要任何访问寄存器或内存，取数最快，I 正确。堆栈寻址可能是硬堆栈（寄存器）或软堆栈（内存），采用软堆栈时比寄存器寻址慢，II 错误。寄存器一次间接寻址先访问寄存器得到地址，然后再访问主存；而变址寻址访问寄存器 IX 后，还要将 A 和 (IX) 相加（相加需要消耗时间），在根据相加的结果访存，显然后者要慢一点，III 错误。一次间接寻址需要两次访存，显然慢于

变址寻址，IV正确。

18. C。

【解析】考查 RISC 的特点。选项 A 明显错误。RISC 选择那些常用的、寄存器型的指令，并不是为了兼容 CISC，RISC 也不可能与 CISC 兼容，B 错误。RISC 中复杂指令是通过简单指令的组合来实现的，D 错误。

19. A。

【解析】本题考查微指令的编码方式。编码的是对微指令的控制字段进行编码，以形成控制信号；目的是在保证速度的情况下，尽量缩短微指令字长。微命令个数为 8 时，需要 4 位，假设只用 3 位，将会造成每个编码都会输出一个微命令，事实上，微命令的编码需要预留一个字段表示不输出，I 错误。垂直型微指令的缺点是微程序长、执行速度慢、工作效率低，III 错误。字段间接编码中的一个字段的某些微命令还需由另一个字段中的某些微命令来解释，即受到某一个字段的译码输出，IV 错误。

20. B。

【解析】本题考查总线仲裁。独立请求方式每个设备均由一对总线请求线和总线允许线，总线控制逻辑复杂，但响应速度快，I 正确。计数器定时方式采用一组设备地址线(约 $\log_2 n$)，II 错误。链式查询方式对硬件电路故障敏感，且优先级不能改变，III 正确。分布式仲裁方式不需要中央仲裁器，每个主模块都有自己的仲裁号和仲裁器，多个仲裁器竞争使用总线，IV 正确。

21. D。

【解析】考查总线的分类与特点。地址、控制和状态信息都是单向传输的，数据信息是双向传输的。

22. B。

【解析】本题考查中断屏蔽字的设置。屏蔽字可以改变中断处理优先级，利用中断屏蔽字可以不改变中断响应次序的情况下改变中断处理的次序。在屏蔽字中，1 表示屏蔽该中断，0 表示响应该中断。1 级中断的屏蔽字为 1101，表示屏蔽 1 级、2 级和 4 级中断；2 级中断的屏蔽字为 0100，表示屏蔽 2 级中断（即其自身，故可知优先级最低）；3 级中断的屏蔽字为 1111，表示能屏蔽所有级中断（优先级最高）；4 级中断的屏蔽字为 0101，表示能屏蔽 2 级和 4 级中断。此外，还有一个简单方法：1 越多优先级就越高，因为屏蔽其他中断源数就越多。

23. B。

【解析】本题考查线程的实现方式。考生要注意掌握进程与线程的区别和联系，以及在具体执行中线程与进程扮演的角色和线程的属性。在多线程模型中，进程依然是资源分配的基本单元，而线程是最基本的 CPU 执行单元，它们共享进程的逻辑地址空间，但各个线程有自己的栈空间。故 I 对、II 错。在多对一线程模型中，一个线程被阻塞了，则整个进程都将被阻塞，III 对、IV 错。假如 IV 对的话，凡是遇到等待 I/O 输出的线程，都被撤销，这显然是不合理的。

24. B。

【解析】本题考查各种调度算法的特点。FCFS 调度算法比较有利于长作业，而不利于短作业。所谓 CPU 繁忙型的作业，是指该类作业需要大量的 CPU 时间进行计算，而很少请求 I/O 操作，故采用 FCFS 可从容完成计算。I/O 繁忙型的作业是指 CPU 处理时，需频繁的请求 I/O 操作，所以 CPU 繁忙型作业更接近于长作业，若采用 FCFS 则等待时间过长。

25. C。

【解析】本题考查并发执行的特点。根据进程的一次执行和并发执行的区别来分析影响进程推进速度的因素。在进程的一次运行过程中其代码的执行序列是确定的，即使有循环、转移、或等待，对于进程来讲，其运行的轨迹也是确定的。当进程存在于一个并发系统中时，这种确定性就被打破了。由于系统中存在大量的可运行的进程，操作系统为了提高计算机的效率，会根据用户的需求和系统资源的数量来进行进程调度和切换。此时，进程由于被调度，打破了原来的固执执行速度，因此，进程的相对速度就不受进程自己的控制，而是取决于进程调度的策略。

26. B。

【解析】本题考查互斥信号量的性质。mutex 初值为 1，表示允许一个进程进入临界区，当由一个进程进入临界区且没有进程等待进入时，mutex 减 1，变为 0。|mutex|为等待进入的进程数。

27. D。

【解析】并非所有的不安全状态都是死锁状态，但当系统进入不安全状态后，便可能进入死锁状态；反之，只要系统处于安全状态，系统便可以避免进入死锁状态；死锁状态必定不是安全状态。

28. C。

【解析】本题考查各种分配策略。采用首次适应算法得到的首地址为 100K，采用最佳适应算法得到的首地址为 350K，采用最坏适应算法得到的首地址为 410K，采用临近适应算法得到的首地址是 200K，所以采用最坏适应算法得到的首地址最大。

29. D。

【解析】本题考查系统抖动。要通过对存储分配的理解来推断系统是否会发生抖动，所以本题同时也需要了解不同的存储分配方案的内容。抖动现象是指刚刚被换出的页很快又要被访问，为此，又要换出其他页，而该页又很快被访问，如此频繁地置换页面，以致大部分时间都花在页面置换上。对换的信息量过大，内存容量不足不是引起系统抖动现象的原因，而选择的置换算法不当才是引起抖动的根本原因，例如，先进先出算法就可能会产生抖动现象。本题中只有虚拟页式和虚拟段式才存在换入换出的操作，简单页式和简单段式因已经全部将程序调入内存，因此不需要置换，也就没有了抖动现象。

30. D。

【解析】本题考查 FCB 的存储。考虑到存取的效率，FCB 的存放是不能分开的，所以 1KB 大小的盘块能存放的 FCB 数为 $1024/64=16$ 。

31. C。

【解析】本题考查位示图。先求出块号为 100 所在的字号，0~31 在字号 0，32~63 在字号 1，64~95 在字号 2，96~127 在字号 3，所以块号 100 在字号 3。接下来求出第 100 块在字号 3 的哪一位，字号 3 的第 0 位是第 96 块，以此类推第 100 块在字号 3 的第 4 位。或者，字号=100/32=3，位号=100%32=4。对于此类题，为了避免出错，建议画出草图求解。

32. B。

【解析】本题考查设备独立性的定义。设备独立性的定义就是指用户程序独立于具体物理设备的一种特性，引入设备的独立性是为了提高设备分配的灵活性和设备的利用率等。

33. A。

【解析】本题考查网络体系结构的原则和特点。网络体系结构是抽象的，它不包括各层协议及功能的具体实现细节，若规定层次名称和功能，则难以保持网络的灵活性。分层使得各层次之间相对独立，各层仅需关注该层需要完成的功能，保持了网络的灵活性和封装性，但网络的体系结构并没有规定层次的名称和功能必须一致。

注意：典型的如 OSI 参考模型，就很好地体现了网络体系结构设计的初衷。

34. D。

【解析】本题考查几种交换技术。电路交换、分组交换、报文交换及数据报服务、虚电路服务这些易混淆知识点容易出串联性选择题，要在对比中加深理解和记忆。电路交换是真正的物理线路交换，例如电话线路；虚电路交换是多路复用技术，每一条物理线路可以进行多条连接，是逻辑上的连接，因此 A 错误。虚电路不只是临时性的，它提供的服务包括永久性虚电路（PVC）和交换型虚电路（SVC）。其中前者是一种提前定义好的，基本上不需要任何建立时间的端点之间的连接，而后者是端点之间的一种临时性连接，这些连接只持续所需的时间，并且当会话结束时就取消这种连接，因此 B 错误。数据报服务是无连接的，不提供可靠性保障，也不保证分组的有序到达，虚电路服务提供可靠性，且保证分组的有序到达，因此 C 错误。数据报服务中，每个分组在传输过程中都必须携带源地址和目的地址；而虚电路服务中，在建立连接后，分组只需携带虚电路标识，而不必带有源地址和目的地址。

35. C。

【解析】本题考查 CSMA/CD 协议中冲突时间的概念。以太网端到端的往返时延称为冲突时间。为了确保站点在发送数据的同时能检测到可能存在的冲突，CSMA/CD 总线网中所有数据帧都必须大于一个最小帧长。任何站点收到帧长小于最小帧长的帧就把它当做无效帧立即丢弃。站点在发送帧后至多经过 2τ （争用期）就可以知道所发送的帧是否遭到了碰撞。因此，最小帧长的计算公式为：最小帧长=数据传输速率×争用期。而最大帧碎片长度不得超过最小帧长。冲突时间就是能够进行冲突检测的最长时间，它决定了最小帧的长度和最大帧碎片的长度，而最大帧的长度受限于数据链路层的 MTU。

36. C。

【解析】本题考查 CSMA/CD 协议。以太网采用 CSMA/CD 技术，当网络上的流量越多，负载越大时，发生冲突的机率也会越大。当工作站发送的数据帧因冲突而传输失败时，会采用二进制指数后退算法后退一段时间再重发数据帧。二进制指数后退算法可以动态地适应发送站点的数量，后退延时的取值范围与重发次数 n 形成二进制指数关系。当网络负载小时，

后退延时的取值范围也小；而当负载大时，后退延时的取值范围也随着增大。二进制指数后退算法的优点正是把后退延时的平均取值与负载的大小联系起来。所以，二进制指数后退算法考虑了网络负载对冲突的影响。

37. A。

【解析】本题考查 CIDR 的作用。CIDR，即无分类域间路由选择，是一种无分类的编址方法，它消除了传统的 A、B、C 类地址和子网划分的概念，并实现地址聚合，即构成超网。所以可以说 CIDR 技术的作用是把小的网络汇聚成大的超网。

38. A。

【解析】路由器可以隔离广播域和冲突域，要屏蔽数据链路层的广播帧，当然应该是网络层设备，因此只能选路由器。而以太网交换机和网桥是数据链路层的设备，只能隔离冲突域，不能隔离广播域。此外，集线器作为物理层设备，既不能隔离广播域，又不能隔离冲突域。

39. B。

【解析】本题考查以太网中 IP 数据报的分片。因为 IP 数据报被封装在链路层数据报中，故链路层的 MTU(最大传输单元)严格地限制着 IP 数据报的长度。以太网帧的 MTU 是 1500B，IP 头部长度为 20B，因此以太网的最大数据载荷是 1480B，因此 3000B 的数据必须进行分片， $3000=1480+1480+40$ 共 3 片。

40. B。

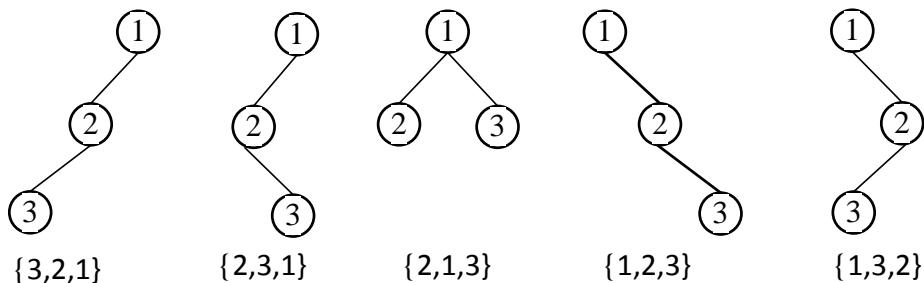
【解析】本题考查对 WWW 服务的理解。如果用户直接使用域名去访问一个 WWW 服务器，其过程是：域名解析、建立 TCP 连接、传输数据、释放连接。只有获得服务器的 IP 地址后，WWW 浏览器才能与 WWW 服务器建立连接开始后续的交互。因此从协议执行过程来说，访问 WWW 服务器的第一步是域名解析。

二、综合应用题

41. 解析：

由于二叉树前序遍历序列和中序遍历序列可唯一确定一棵二叉树。因此，若入栈序列为 1,2,3,...,n，相当于前序遍历序列是 1,2,3,...,n，出栈序列就是该前序遍历对应的二叉树的中序序列的数目，而中序遍历的过程实质就是一个结点进栈和出栈的过程。

二叉树的形态确定了结点进栈和出栈的顺序，也就确定了结点的中序序列。当结点入栈序列为{1,2,3}时，出栈序列可能为：{3,2,1}，{2,3,1}，{2,1,3}，{1,3,2}，{1,2,3}，它们对应二叉树如下：



【扩展】进栈出栈操作与二叉树中序遍历的关系：①一个结点进栈后有两种处理方式：要么立刻出栈（没有左孩子）；或者下一个结点进栈（有左孩子）。②一个结点出栈后也有两种处理方式：要么继续出栈（没有右孩子）；或者下一个结点进栈（有右孩子）。

42. 解析：

（1）算法的基本设计思想：

本题要找 p 和 q 的最近父结点 r ，不失一般性，设 p 在 q 的左边。采用后序遍历，后序遍历最后访问根结点，即在递归算法中，根是压在栈底的，当访问到某结点时，栈中所有结点均为该结点的祖先。后序遍历必然先遍历到结点 p ，栈中元素均为 p 的祖先。先将栈复制到另一辅助栈中。继续遍历到结点 q 时，将栈中元素从栈底开始逐个到辅助栈中去匹配，第一个失配的元素前面元素就是结点 p 和 q 的共同父结点。

算法的步骤如下：

①设要匹配结点是 p 和 q ，设置当前工作指针 bt ，初始化为 $root$ ，栈顶指针 top 为 0。

②顺着 bt 的左分支查找元素。如果 bt 与 p 和 q 都不相等，则把 bt 的左分支进栈；如果 bt 的左子树为空，转向③。

③如果栈非空，栈顶元素右子树没有进栈，执行④，如果右子树进栈，那么判断栈顶元素是否与 p 和 q 中一个第一次相等，如果和 p 和 q 都不相等则退栈，重复执行③。如果找到第一个相等，则保存此时的栈中元素和剩余的指针。如果栈顶元素与 p 和 q 中剩余那个指针相等，则从栈底开始逐个到辅助栈中去匹配，返回第一个失配前的元素。结束。

④栈顶元素的右子树进栈， bt 指向栈顶元素的右子结点，执行②。

（2）算法的实现如下：

```
typedef struct{
    BiTree t;
    int tag;           //tag=0表示左子女已被访问，tag=1表示右子女已被访问
}stack;

stack s[],s1[];      //栈，容量足够大

BiTree CommonParent(BiTree ROOT,BiTNode *p,BiTNode *q){ //求二叉树中p和q的最近公共父节点

    int top=top1=0;   //栈顶指针
    BiTree bt=ROOT;   //初始化bt工作指针
    Bool first=false;  //第一次找到p或者q
    BiTree rest=NULL;  //找到p或者q后的剩余的指针
    while(bt!=NULL||top>0){
```

```

while (bt!=NULL&&(bt!=p||bt!=q)) { //结点入栈。
    s[++top].t=bt;
    s[top].tag=0;
    bt=bt->lchild;
} //沿左分支向下
if (bt!=NULL&&bt==p&&bt==q) return top>0? s[top].t:NULL;
//如果p和q相同，直接返回父结点
while (top!=0&&s[top].tag==1) { //如果右孩子树已经遍历过，遍历本结点，实现
后序遍历
    if (!first) {
        if (s[top].t==p) { //假定遇到p时，栈中元素均为p的祖先
            for (i=1; i<top; i++)
                s1[i]=s[i]; //复制到辅助栈
            top1=top-1;
            first=true;
            rest=q; //设置剩余指针
        } //将栈s的元素转入辅助栈s1保存其父节点
        else if (s[top].t==q) { //假定遇到q时，栈中元素均为q的祖先
            for (i=1; i<top; i++)
                s1[i]=s[i]; //复制到辅助栈
            top1=top-1;
            first=true;
            rest=p; //设置剩余指针
        }
    }
    else if (s[top].t==rest) { //找到剩余结点
        int min=top1<top-1? top1:top-1;
        for (int i=1; i<=min; i++) { //将栈中元素的树结点到s1中去匹配
            if (s[i].t!=s1[i].t) return s[i-1];
        }
        if (i>min) return s[min];
    }
    top--; //退栈
} //while(top!=0&&s[top].tag==1)
if (top!=0) { //此时，s[top].tag==0，所以右孩子进栈。
    s[top].tag=1;
    bt=s[top].t->rchild;
} //沿右分支向下遍历

```

```
    } //while (bt!=NULL||top>0)
}
```

【思路（2）】从根结点开始，判断以当前结点为根的左右子树中是不是包含我们要找的两个结点。如果两个结点都出现在它的左子树中，那么最低的共同父结点也出现在它的左子树中。如果两个结点都出现在它的右子树中，那么最低的共同父结点也出现在它的右子树中。如果两个结点一个出现在左子树中，一个出现在右子树中，那么当前的结点就是最低的共同父结点。可以采用树的后序遍历。

【思路（3）】本题可以求出从根结点到 p 和 q 的路径（看成是单向链表），然后设想每个结点都有一个指向其父结点的指针（建 2 个辅助链表，也即将单向链表逆置），于是可以从任何一个结点出发，得到一个到达根结点的单向链表，因此这个问题转换为求两个单链表的第一个公共结点。

43 . 解析：

本题考查指令的寻址方式。前两小题涉及数据寻址，其最终目的是寻址操作数，第 3 小题涉及指令寻址，其目的是寻址下一条将要执行的指令地址。下表列出了基本的寻址方式，其中偏移寻址包括变址寻址、基址寻址和相对寻址三种方式。

寻址方式	规则	主要优点	主要缺点
立即寻址	操作数=A	无需访问存储器	操作数范围受限
寄存器寻址	EA=R	无需访问存储器	寻址空间受限
直接寻址	EA=A	简单	寻址空间受限
间接寻址	EA=(A)	寻址空间大	多次访问主存
寄存器间接寻址	EA=(R)	寻址空间大	多访问一次主存
偏移寻址	EA=(R)+A	灵活	复杂

特别注意相对寻址方式中的 PC 值更新的问题：根据历年统考真题，通常在取出当前指令后立即将 PC 的内容加 1（或加增量），使之变成下条指令的地址。

（1）变址寻址时，操作数 $S=((R_x)+A)=(23A0H+001AH)=(23BAH)=1748H$ 。

（2）间接寻址时，操作数 $S=((A))=((001AH))=(23A0H)=2600H$ 。

（3）转移指令使用相对寻址，因为指令字长等于存储字长，PC 每取出一条指令后自动加 1，因此转移地址 $=(PC)+1+A=1F05H+1+001AH=1F20H$ 。若希望转移到 23A0H，则指令的地址码部分应为 $23A0H-(PC)-1=23A0H-1F05H-1=049AH$ 。

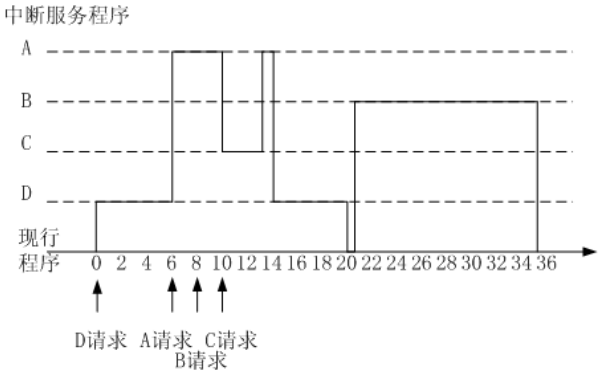
44 . 解析：

本题考查中断处理次序。中断响应次序和中断处理次序是两个不同的概念。中断响应次序也称为硬件排队次序，它是不可改变的。在不改变硬件排队电路的前提下，可以通过改变中断屏蔽字来改变中断处理的优先级，使原来级别较低的中断源变成较高的级别。

（1）由题意，可知中断处理的次序为 $C>A>D>B$ 。屏蔽码中的“1”表示屏蔽该中断源的中断请求，“0”表示没有屏蔽，各中断服务程序的屏蔽码如下表所示。

中断源	中断屏蔽码			
	A	B	C	D
A	1	1	0	1
B	0	1	0	0
C	1	1	1	1
D	0	1	0	1

（2）各级中断发出的中断请求信号的的时刻，画出 CPU 执行中断服务程序的序列，如下图所示。第 0us 时，D 请求到来，由于没有其他的中断请求，所以开始执行中断服务程序 D。第 6us 时，A 请求到来，A 的优先级高于 D，转去执行中断服务程序 A。第 8us 时，B 请求到来，由于 B 的优先级低于 A，所以不响应 B 请求，继续执行中断服务程序 A。第 10us 时，C 请求到来，C 的优先级最高，虽然此时中断服务程序 A 还没结束，也必须暂停转去执行中断服务程序 C。中断服务程序 C 所需时间为 3us，当第 13us 时，中断服务程序 C 执行完毕，返回执行中断服务程序 A。第 14us 时，中断服务程序 A 执行完毕（共执行 5us），返回执行中断服务程序 D。第 20us 时，中断服务程序 D 执行完毕（共执行 12us），返回现行程序。因为 B 请求还存在，所以此时开始执行中断服务程序 B，直至 35us 结束（共执行 15us）。



（3）在 35us 时间内，完成了 4 级中断的处理，所以平均执行时间为 $35/4=8.75\mu s$ 。

45 . 解析：

（1）采用首次适应算法，空间从低位开始分配；

空块起始地址	大小
150K	20KB
280K	20KB
400K	112KB

（2）采用 BF 算法，则总是分配能满足需要的最小空闲块。

空块起始地址	大小
430K	72KB
210K	80KB

或

空块起始地址	大小
430K	30KB

460K	42KB
210K	90KB

（3）最佳适应算法将产生许多小碎片，性能并非最佳。因此对（1）可以满足申请，但对（2）没有连续的大区，则不能满足申请。

46 . 解析：

（1）当虚页 4 发生缺页时，使用 FIFO 管理策略，则应置换 1 号页帧中的 1 号虚页，因为它是最先进入存储器的。

（2）当虚页 4 发生缺页时，使用 LRU 管理策略，则应置换 1 号页帧中的 1 号虚页，因为它是最久未被访问和修改过，又是最先进入存储器的。

（3）当虚页 4 发生缺页时，使用 Clock 管理策略，则应置换 1 号页帧中的 1 号虚页，因为它在本周期内既未被访问过，又没有修改过。

（4）

页访问串	当前状态	4	0	0	0	2	4	2	1	0	3	2
标记		*							*		*	
M1	2	2	2	2	2	2	2	2	2	2	2	2
M2	1	4	4	4	4	4	4	4	4	4	3	3
M3	0	0	0	0	0	0	0	0	0	0	0	0
M4	3	3	3	3	3	3	3	3	1	1	1	1

采用 LRU 算法，缺页次数为 3 次。

47 . 解析：

 本题考查TCP首部的序号和确认号字段。TCP首部的序号字段是用来保证数据能有序提交给应用层，序号是建立在传送的字节流之上；确认号字段是期望收到对方的下一个报文段的数据的第一个字节的序号。

（1）第1个报文段的序号是90，说明其传送的数据从字节90开始，第2个报文段的序号是120，说明其传送的数据从字节120开始，即第1个报文段的数据为第90~119号字节，共30字节。同理，可得出第2个报文段的数据为30个字节。

（2）主机B收到第2个报文段后，期望收到A发送的第3个报文段，第3个报文段的序号字段为150，故发回的确认中的确认号为150。

（3）主机B收到第3个报文段后发回的确认中的确认号为200，则说明已收到第199号字节，故第3个报文段的数据为第150~199号字节，共50字节。

（4）TCP默认使用累计确认，即TCP只确认数据流中至第一个丢失（或未收到）字节为止的字节。题中，第2个报文段丢失，故主机B应发送第2个报文段的序号120。