**一、（35分） 基础知识填空**

1.对于非空满K叉树，其分支结点数目为n，则其叶结点数目为\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_。  
2.高度为h的AVL树上的最少结点个数是\_\_\_\_\_\_\_\_\_\_个，最多结点个数是\_\_\_\_\_\_\_\_\_\_\_\_\_。  
3.有n个顶点的有向连通图最少有\_\_\_\_\_\_\_\_\_条边。  
4.某二叉树中序序列为A，B，C，D，E，F，G，后序序列为B，D，C，A，F，G，E，则前序序列是\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_。  
5.关键码序列（tang,deng ,an ,wan,shi,bai ,**fang** ,liu）按字典顺序排序（”an”&lt;”bai”,空格小于任何其它字符）。采用“右补空格”，使得这些关键码都成为4位字符的等长关键码。按最低位优先法进行基数排序，进行两趟分配和收集后得到的序列为\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_。  
6.将一个A[1…100,1…100]的三对角矩阵，按行优先顺序存储在一维数组B[1…298]中，A中元素A[66][65]在B数组中的位置K为  
注释：三对角矩阵是对角线以及对角线的上下两条与对角线斜率相同的紧邻斜线上的元素不为零，而其它位置元素都是零的矩阵，它的特点是第一行和最后以行各有两个非零元素，去他各行各有三个非零元素  
**二、（30分） 辨析题**  
1.请设计一种“自调整’数据结构。假设检索的关键码是中文词组。 要求：  
（1）定义其ADT（说明其逻辑结构，以及主要的运算）。  
（2）简单的说明它怎样根据所检索过的中文词组的而进行调整，使得最近出现的词组能比较快速的被检索到。  
2．请判断下述计算前缀表达式的算法是否正确。  
（a）如果正确，请给出一个简单实例，演示算法运行步骤；  
(b)如果不正确，请指出错误之处，并给出改正后的算法，如果只是局部修改，可以只具体给出所修改部分 ，其他未修改部分，用其原来的功能编号(1)(2)，a, b.1,b.2 指代。  
[计算前缀表达式的算法]  
// 利用一个栈，从左向右扫描，边扫描边求值。  
（1） 从左向右扫描前缀表达式：  
a : 当遇到的是一个运算符时，则操作数入栈，转到1。  
b．对于操作数，如果栈为空，则将该操作数返回，退出；如果栈不空：  
b.1：当栈顶是一个操作符时，则将操作数压入栈中，转到1；  
b.2：当栈顶也是一个操作数是，两次取栈顶（第二次取得的应为操作符），并按照运算符对两个操作数进行计算（其中从栈中取得的操作数为第一操作数），所的计算结果作为操作数，将新操作数入栈，转到1。  
（2）扫描结束，正确的情况，输出结果应该在b处返回，并退出；如果程序没有在b处返回退出，而是进行到此处，则表明出错，输出错误消息。  
[计算前缀表达式的算法结束]  
注释：前缀表达式与后缀表达式一样，不包含括号，运算符放在两个运算对象的前面，如\*+213。  
3．请判断下面快速模式匹配算法KMP\_FindPat()是否正确。  
（a） 如果正确，请给出一个简单实例，演示算法运行步骤；  
（b）如果不正确，请指出错误之处，并给出改正后的算法，如果只是局部修  
改，只需要具体给出所修改的部分（例如，第x行-第y行代码修改为………）。  
int \*Next(String P){  
int m=P.strlen(); //m为模板P的长度  
**assert**( m&gt;0); // 若m=0,退出  
int \*N = new int[m]; //动态存储区开辟整数数组  
**assert** ( N !=0); //若开辟存储区域失败，退出  
N[0]=0; //分析P的每个位置i  
for (int i=1;i<m;i++){  
</m;i++){  
 int k=N[i-1]; //第（i-1）位置的最长前缀串长度  
//以下while 语句递推决定合适的前缀位置k  
while (k&gt;0 &amp;&amp;p[i]!=p[k])  
K=N[k-1];  
//根据P[i]比较第k位置前缀字符，决定N[i]  
If (p[i]==p[k])  
N[i]=k+1;  
else N[i] = 0;  
}  
return N;  
}  
int KMP\_FindPat(String S,String P, int \*N,int startindex){ /\* 第1行\*/  
//假定事先已经计算出P的特征数组N，并作为输入参数 /\* 第2行\*/  
int LastIndex=S.strlen()-P.strlen(); /\* 第3行\*/  
if ((LastrIndex-startindex)&lt;0) /\* 第4行\*/  
Return(-1); // 匹配失败 /\* 第5行\*/  
int i;// i是指向S 内部字符的游标 /\* 第6行\*/  
int j=0; // j是指向P内部字符的游标 /\* 第7行\*/  
for (i=startindex;i&lt;LastIndex;i++){ //I 是S游标 /\* 第8行\*/  
//若当前位置的字符不同，则用N循环求当前的j /\* 第9行\*/  
while ( P[j]!=S[i]) /\* 第10行\*/  
j=N[j-1]; //寻找j将p 的恰当位置与S 的i位置对准 /\* 第11行\*/  
if (P[j]==S[i]) //P[j]==S[i]位相同时，则继续下一步循环 /\* 第12行\*/  
j++; /\* 第13行\*/   
//若当前位置的字符不同，j将由下一步循环的N循环决定 /\* 第14行\*/  
if (j==P.strlen()) /\* 第15行\*//  
return(i-j); //匹配成功，返回该S子串的开始位置 /\* 第16行\*/  
} /\* 第17行\*/  
return (-1); /\* 第18行\*/  
} /\* 第19行\*/  
**三、（15分）算法填空**  
下面是一个拓扑排序算法，对于带回路的图，该算法将调用 “此图有环！”，并逐步退出到调用它的主调函数。请填充算法中空缺的部分，使其成为完整的算法。  
//深度优先搜索方式实现拓扑排序  
**void** Do\_topsort\_Circle(Graph &amp; G,int V, int \*result,int &amp; tag)  
{  
if(G.Mark[v]==VISITED){  
空缺1   
}  
G.Mark[v]=VISITED;  
for (Edge e=G.FirstEdge(v);G.IsEdge(e);e.=G.NextEdge(e)){  
If (空缺2 ）  
Do\_topsort\_Circle(G,G.ToVertex(e),result,tag);  
}  
result[tag++]=V;  
空缺3   
}  
**四、（40分） 问答题**  
1. 实现进程通信的方式有几种？请分别简要描述这些通信方式。  
2. 在实现虚拟页式存储管理方案时，页表表项是由什么决定的？通常页表设置那些表项？每一表项的作用是什么？  
3. 文件系统的性能对整体系统的性能影响很大，请说明在实现文件系统时可以从哪些方面提高文件系统的性能，简要给出这些手段的具体解决思路。  
**五、（20分）综合应用题**  
1. 系统中有5个进程，每个进程的运行时间（单位：ms）、优先级和到达时刻如下表所示  
进 程： P1 P2 P3 P4 P5  
运行时间： 10 1 2 1 5  
优 先 级： 4 6 2 3 6  
到达时刻： 0 1 2 3 4  
请给出当系统跟别采用时间片轮转算法（时间片为1ms）、不可抢占优先级调度算法和抢占式优先级调度算法时，各进程的执行情况。  
2. 假设有如下访盘请求，请计算出对于这些请求的服务次序，使得平均访问时间最短。设当前磁头的位置是6号柱面。，  
请求顺序 柱面号 磁头号 扇区号  
① 3 2 1  
② 5 1 5  
③ 3 2 5  
④ 3 4 1  
⑤ 9 2 1  
⑥ 9 1 5  
⑦ 5 2 5  
⑧ 5 4 8  
**六、（10分） P，V 操作题**  
某系统如此定义P，V操作：  
P （S）：  
S=S-1 ；  
若S&lt;0,本进程进入s信号量等待队列的末尾；否则，继续执行。  
V（S）  
S=S+1；  
若S≤0，释放等待队列中末尾的进程，否则继续运行。  
现有四个进程 ， ， ， 竞争使用某一个互斥资源（每个进程可能反复使用多次），试用上面定义的P，V操作正确解决 ， ， ， 对该互斥资源的使用问题。