《数据结构》期中考

一、 填空题（25%）(第 1~10 题每题 2 分，第 11 题 5 分)

1. 一个栈的进栈序列是 A、B、C，写出所有可能的出栈序列_____。

答案：ABC，ACB，BAC，BCA，CBA

评分标准：少写或填错不给分，只有全对才给 2 分。

2. 广义表 GL= ((a,b,c),(d,e,f))，假设求表头操作为 Head，求表尾操作为 Tail，则 f=_____。

答案：head [tail [tail [head [tail [GL]]]]]

3. 具有 4 个节点的不同二叉树共有_____棵。

答案：14 或者是 14 * 4! = 336

4. 已知数组 A[1…10, 1…10]为对称矩阵，期中每个元素占 5 个单元。现将其下三角部分按行优先次序存储在起始地址为 1000 的连续内存单元中，则元素 A[5][6]对应的地址为_____

答案：1095

5. 算术表达式(x+y)/10+a*b/c-(x-y)*(a-b)转为后缀表达式后为__。

答案：xy+10/ab*c/+xy-ab-*-

6. 如果表示静态链表的数组 elem[1].link =3, elem[1].data = 1, elem[2].link = 3, elem[2].data = 2, elem[3].link = 4, elem[3].data = 5, 则在静态链表中 elem[1]指向的下一个元素的值是_____.

答案： 5

7. 设栈 s 和队列 q 的初始状态为空，元素 a, b, c, d, e 和 f 依次通过栈 s，一个元素出栈后即进入队列 q，若 6 个元素出队的序列是 bdcfea，则栈 s 的容量至少应该存放____个元素。

答案：3

8. 树的后序遍历和它通过左孩子右兄弟方法转换的二叉树的_____遍历是等价的。

答案：中序

9. 试判断下面的关键码序列中哪一个是堆_____。

① {94, 31,53, 23, 16, 72} ② {16, 31, 23, 94, 53, 72} ③ {16, 53, 23, 94, 31, 72}

④ {94, 53, 31, 72, 16, 23} ⑤ {16, 72, 31, 23, 94, 53}

答案：2

10. 已知一棵二叉树是以二叉链表的形式存储的，其结点结构说明如下：

```
struct node {
```

```
        int data;
        struct node *left;
        struct node *right;
        }
```

请在下面的_____处进行填空，空成题目要求的功能。注意：每空只能填一个语句，多填为 0 分。

求出以 T 为根的二叉树或子树的结点个数

```
int size (struct node *T){
   if (_____________)
        return 0;
   else
      ____________
}
```

答案： T == NULL， return(1+size(T->left)+size(T->right));

评分标准：每空 1 分。共 2 分。

11. 高度为 h 的满二叉树(，其结点总数 n 可以用 h 表示为_____，而 h 可以用 n 表示为_____，若该树的结点从零开始按中序遍历顺序编号，则树根的编号用 h 表示为_____，树根的左孩子的编号用 h 表示为_____，树根的右孩子的编号用 h 表示为_____。

答案：$2^{h+1}-1$, $\log_2(n+1)-1$, 2^h-1, $2^{h-1}-1$, $2^{h+1}-2^{h-1}-1$

评分标准： 每空 1 分，共 5 分。

二、 问答题（35%，每题 7 分）

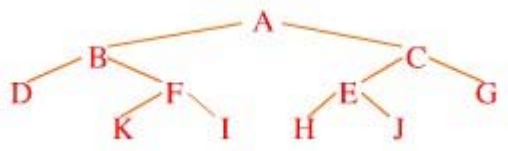
1. 一棵二叉树的前序、中序和后序遍历序列分别如下，其中有一部分未显示出来（每个空白均只有一个节点），请补全下列空白并画出该二叉树。（7 分）

   先序序列： _B_F_ICEH_G;

   中序序列： D_KFIA_EJC_;

   后序序列： _K_FBHJ_G_A;

答案：



先序序列：ABDFKICEHJG 中序序列:DBKFIAHEJCG 后序序列：DKIFBHJEGCA

评分标准：

先序 2 分，每一个空格 0.5 分

中序 2 分，只对 1 个给 1 分，对 2 个给 1.5 分，对 3 个给 2 分。

后序 2 分，每一个空格 0.5 分

二叉树 1 分，画错不给分，全对给 1 分。

2. 已知目标串 T=”abcabbabcabaaa”，模式串 P=”abcabaa”。（7 分）

a) 请按照 KMP 算法的要求计算模式串 P 的 next[ ]数组包括 next 数组定义方法和优化方法。

b) 请画出利用 a)生成的 next 数组，KMP 算法进行模式匹配时的每一趟匹配过程。

评分标准：a 部分 4 分，每一空给 0.5 分。全对给 4 分。刘鹏

b 部分 3 分，对 1 个给 1 分，对 2 个给 1.5 分，对 3 个给 2 分，全对给 3 分。

定义

0	1	2	3	4	5	6
A	B	C	A	B	A	A
-1	0	0	0	1	2	1

优化：

0	1	2	3	4	5	6	7
A	B	C	A	B	A	A	
-1	0	0	-1	0	2	1	1

(1)

定义匹配过程:

abcabbabcabaaa

| | | | | X

abcabaa

abcabbabcabaaa

| | X

abcabaa

abcabbabcabaaa

X

abcabaa

abcabbabcabaaa

| | | | | | |

abcabaa

(2)优化匹配过程

abcabbabcabaaa

| | | | | X

abcabaa

abcabbabcabaaa

| | X

abcabaa

abcabbabcabaaa

X

abcabaa

abcabbabcabaaa

|||||||

abcabaa

3. 找出所有满足下列条件的二叉树，假设任意两个节点都不相同：

(1) 它们的前序遍历序列和中序遍历序列相同。

(2) 它们的后序遍历序列和中序遍历序列相同。

(3) 它们的前序遍历序列和后序遍历序列相同。

答案 (1) 空二叉树、仅有根节点的二叉树、全部只有右子树的二叉树

(2) 空二叉树、仅有根节点的二叉树、全部只有左子树的二叉树

(3) 空二叉树、仅有根节点的二叉树

评分标准：刘鹏

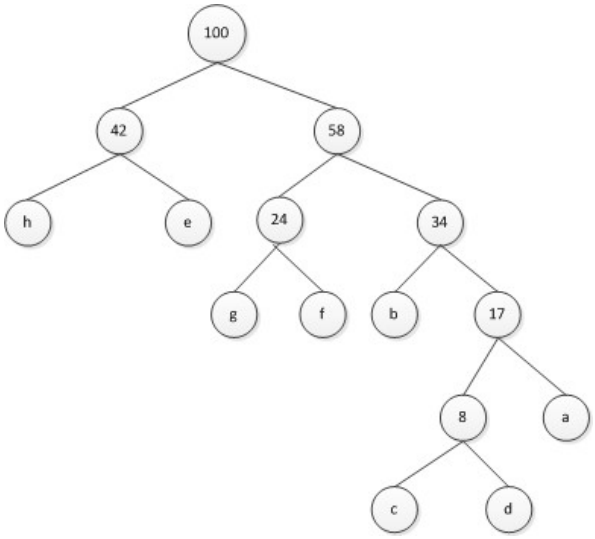
(1) 3 分，对一个给 1 分，一共 3 分

(2) 3 分，对一个给 1 分，一共 3 分。

(3) 1 分，对一个给 0.5 分，一共 1 分。

4. 假设在通信中，只允许出现 8 种字符，分别用 a、b、c、d、e、f、g、h、来代替，假设字符出现的频率依次为 9、17、2、6、22、13、11、20，试建立哈夫曼树。

答案：



评分标准：一共有 7 步，每对一步给 1 分。共 7 分。刘鹏

5. 有如下递归算法：

```
void print( int w) {
    int i;
    if ( w != 0){
        print(w - 1);
        for ( i = 1; i <= w; i++)
            cout<<w;
        cout<<endl;
    }
}
```

调用语句 print(4)的结果是什么？不用栈将该递归算法改为非递归。

答案：print(4)的结果是：

```
1
2  2
3  3  3
4  4  4  4
```

```
void print( int w) {
      int k = 1;
      while(k!=w)
      {
         For(int i = 1; i <=k; i++)
            cout<<k;
         cout<<endl;
         k++;
      }
  }
```

评分标准：1. 输出结果 4 分，每对一行给 1 分。共 4 分。刘鹏

2. 改非递归 3 分。代码基本正确给 3 分，部分有错不影响正常阅读给 2 分，只对部分给 1 分，错误给 0 分）

三、 算法设计题（40%）

1、 将下面 Hanoi 塔问题的递归算法转化为非递归算法。（10 分）

```
void towers(int n, int A, int B, int C)
{
    if(n == 1) cout<<A<<”→”<<C<<endl;
    else
    {
       towers(n-1, A, C, B);
       cout<<A<<”→”<<C<<endl;
       towers(n-1, B, A, C);

    }
}
```

参考答案：

评分标准：思路基本正确(提到用栈来转化成非递归)给 1 分，正确并详细说明酌情给 2~3 分；否则本题 0 分。不写思路不评分。

代码基本正确给 7 分，代码能表达出思路的正确意思但部分书写错误，每错一处扣一分。代码不知所云给 0 分。 （刘雪君）

```
void towers( int n, int A, int B, int C){
   stack s;
   int done = false;

   while(!done){

       while(n>1){
       s_tack(n, A,B,C,&s);
       setVar1(&n, &A, &B, &C);
   }
    Cout<<A<<”→”C<<endl;
   if(!empty(&s))
   {
      Restore(&n, &A, &B, &C, &s);
      Cout<<A<<”→”C<<endl;
      setVar2(&n, &A, &B, &C);
   }
   else done = true;
 }
}
```

s_stack()函数将它的参数放入栈中。setVar1()把 n 设为 n-1,并且交换 C 和 B 的值，setVar2()把 n 设为 n-1,并且交换 A 和 B 的值，restore()负责将局部变量恢复。

2、 已知二叉树以完全二叉树形式存储在数组当中，p、q 为二叉树的两个节点及该二叉树的节点个数 n，设计算法求这两个节点的最近公共祖先节点。**(节点各不相同)**（最近公共祖先节点是 p 和 q 公共祖先当中层数最大的节点）（15 分）

思路：以完全二叉树形式保存二叉树，求某结点(数组编号为 i)的祖先，如果此节点的序号 i 为奇数，则其父亲结点为(i-1)/2，否则为(i-2)/2。则祖先节点可以通过此方法求出。求 p,q 的公共祖先的第一个交集，即为最近公共祖先。

评分标准：（思路正确清晰 3 分，代码正确完整 12 分）。否则本题 0 分。不写思路不评分。

代码基本正确，缺少对 p、q 的根节点判断扣一分，求父节点的思路正确计算错误得 4 分，代码能表达出思路的正确意思但部分书写错误，每错一处扣一分。代码不知所云给 0 分。

参考代码：

```
int nearestAncestor( int *a, int p ,int q){
    if(p == 0 || q == 0) cout << “没有最近公共祖先 <<endl;
    int first[100],firstSize = 0;
    int second[100],secondSize = 0;
    // 获得 p 的所有祖先节点//
    int temp = p;
    while(temp!=0){
     if(temp%2 == 0)
       { temp = (temp-2)/2; first[firstSize++] = temp;}
     else
       { temp = (temp-1)/2; first[firstSize++] = temp;}
  }
    temp = p;
     while(temp!=0){
     if(temp%2 == 0)
       { temp = (temp-2)/2;
       int i = 0;
       for(i = 0; i < firstSize; i++) if(first[i] == temp) break;
       if( i != firstSize) break;
  }
  cout<<temp << “是最近公共祖先<< endl;
}
```

3、 试编写算法，对二叉树做自底向上，自左向右的层次遍历，并按遍历次序输出各结点的值。(15 分)

思路：要达到自底向上，自左向右的层次遍历，则先自顶向下，自右向左使用队列来对二叉树进行层次遍历，每个出队的节点进入一个栈中，注意在遍历的时候是先遍历右孩子节点再遍历左孩子节点。出栈的时候才能达到自底向上，自左向右的层次遍历结果。

评分标准：思路正确给 5 分。思路中没有提到如何使栈出队序列达到自左向右的层次遍历的处理，得 3 分。

代码基本正确得 10 分，代码中如果没有实现自左向右（实现了自右向左），得 5 分。

```
void revTraversing (NODE *head){
  queue<NODE *> q;
  stack<Type> data;
  NODE *p = head,*temp;
  q.push(p);
  while( !q.empty()){
      temp = q.pop();
      data.push(temp->data);
      if(temp->rchild!=null) q.push(temp->rchild);
      if(temp->lchild!=null) q.push(temp->lchild);
  }
  While(!data.isEmpty())
  {
      cout<<data.top();
      data.pop();
  }

}
```

详细评分标准（陈楚南）

1. 得分：
   a) 思路正确：3 分
   b) 实现层次遍历：4 分
   c) 自左向右输出：8 分
2. 扣分：
   a) Root 没有判断是否为 null：扣 1 分
   b) 输出顺序是自右向左：扣 5 分
   c) 输出树节点而不是 data：扣 1 分