

C 语言选择题要会区分 `strlen()` 和 `sizeof()` 的区别

要知道 `static` 这个关键字, 考了个 `stdin` 和 `stdout` 流的区别

C 语言分析题:

(6 分) 第一题: (插入排序)

```
void F(int i,int A[],int n){  
    int tmp=A[i];  
    for(int j = i+1; j > 0 && A[j-1] > tmp ; j--) A[j] = A[j-1];  
    A[j] = tmp;  
}  
  
int main(){  
    int A[] = {6,-1,7,2,5, 8,9,4};  
    int n = (sizeof(A)/sizeof(A[0]));  
    for(int i = 1; i < n/2; i++) F(i,A,n);  
    for(int i = 0; i < n; i++) printf("%d",A[i]);  
    return 0;  
}
```

求输出的结果: -1 2 6 7 5 8 9 4

(6 分)第二题:

```
#define D(x)      x<=9 && x>=0
#define C(x)      x<=z && x>=a

int main(){
    char A[] = "-12-bBb-";
    int n = strlen(A);
    int num = 0,flag=0;
    for(int i = 0; i < n; i++){
        if(D(A[i])) num = 16*num+A[i] - '0';
        else if(C(A[i])) num = 16*num+10+A[i]-'a';
        else if(A[i] == '-') flag = -flag;
        else break;
    }
    printf("%d",num);
    return 0;
}
```

求输出的结果： 299

(6 分)第三题:

```
#define F(a,b)    (a^=b^=a^=b)
```

```
int Fnc(int A[],int n){  
    int count = 0;  
    for(int i = 0;i < n;i++)  
        for(int j = i+1; j < n;j++)  
            if(A[i] > A[j]) count++;  
    return count;  
}
```

```
void main(){  
    int A[]={5,1,7,4,3,6,2};  
    printf("%d",Fnc(A,7));  
    F(A[1],A[2]);  
    printf("%d",Fnc(A,7));  
}
```

求输出的结果：12 13

（6分）第四题：

定义了一个

```
char format[] = "%d%c";
```

```
printf(format, sizeof(format), "abcdef"[2]+1);
```

求输出的结果：5d

（6分）第五题：填空

删除链表中的逆序元素使链表有序如 1->5->2->3->6->4->7 删除后为  
1->5->6->7

```
void Delete(ListNode* L){  
    ListNode* p = L, t;  
    while(p){  
        t = p->next;  
        if( t == NULL ) break;  
        else if( t->data > p->data ) p = p->next;  
        else{  
            p->next = t->next;  
            free(t);  
        }  
    }  
}
```

```
}  
  
}  
  
}
```

数据结构选择题挺简单的，用王道就没问题了，还有几题是原题。878 考的都是很基础的，比 408 简单多了，看完王道上考场你会发现自己看多了，不过没关系。

## 数据结构

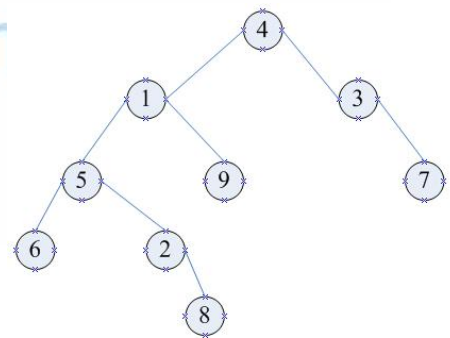
(18 分) 第一题:

已知二叉树的中序遍历: 6 5 1 2 8 9 4 3 7

后序遍历: 6 5 8 2 9 1 7 3 4

1. 画出二叉树

2. 写出二叉树的先序遍历: 4 1 5 6 9 2 8 3 7



(12 分) 第二题:

将关键字序列 (2,11,18,20,23,7,27,42,33,19) 散列存储到散列表中，散列表的存储空间是下标从 0 开始的长度为 11 的一维数组，散列函数为  $H(\text{key}) = \text{key} \text{MOD} 11$ ，处理冲突采用线性探测再散列法。

1. 请画出所构造的散列表

0	1	2	3	4	5	6	7	8	9	10
11	23	2	33	19	27		18	7	20	42

2. 当发生第一次冲突时, 求此时散列表的填装因子 $\alpha$

$$\alpha = 5 / 11$$

(12 分) 第三题:

用 Flyod 算法求解最短路径的问题: 填空

```
for (k=1; k<=n; k++)  
  
    for (i=1; i<=n; i++)  
  
        for (j=1; j<=n; j++) {  
  
            if (d[i][k] + d[k][j] < d[i][j] ) {  
  
                d[i][j] = d[i][k] + d[k][j];  
  
                path[i][j] = path[i][k];  
  
            }  
  
        }  
  
    }
```

(8 分) 算法设计题

设计程序解决二叉排序树的两个节点的最近公共祖先问题。

先序遍历最后访问根节点, 即在递归算法中, 根是压在栈底的, 要找一个节点的祖先只需要在找到节点后把剩余的节点全部按顺序出栈就

行了。程序如下：

```
int Ancestor(Tree T, int n, int length[]) { //返回节点深度，不存在则返回
                                           0，路径保存在 length 中
    TreeNode* p = T;
    InitStack(S);
    push(S, p);
    while(!IsEmpty(S)) {
        push(S, p);
        if(p->data == n) {
            int i = 0;
            while(!IsEmpty(S)) {
                Push(S, p);
                length[i++] = p->data;
            }
            return i-1;
        }
        if(p->lchild) push(s, p->lchild);
        if(p->rchild) push(s, p->rchild);
    }
    return 0;
}

int LCA(Tree T, int u, int v) {
    int length1[Maxsize], length2[Maxsize];
    int deep1=0, deep2=0;
    if(T == NULL) {
        printf("error"); return false;
    }
    deep1 = Ancestor(T, u, length1);
    deep2 = Ancestor(T, v, length2);
    if(!deep1 || !deep2) {
        printf("error"); return false;
    }
}
```

```
for(int i = 0; i<= deep1; i++)  
    for(int j = 0; j < deep2; j++)  
        if(length1[i] == length2[j])  
            return length1[i];  
return false;  
}
```

