第 1 题部分代码参考如下：

```
#include <stdio.h>
#include <stdlib.h>
#define TRUE 1
#define FALSE 0
#define OK 1
#define ERROR 0
#define OVERFLOW -2
#define EQ(a, b) (a==b)
#define LT(a, b) (a<b)
typedef int Status;
typedef char TElemType;
typedef int KeyType;
typedef struct {
    KeyType key;                    //关键字域
}ElemType;
typedef struct BiTNode {
    ElemType data;
    struct BiTNode *lchild, *rchild;     //左右孩子指针
}BiTNode, *BiTree;


void main()
{
    BiTree T=NULL, s;
    int n, i;
    ElemType e;
    printf("输入二叉排序树的结点数：");
    scanf("%d", &n);
    printf("输入%d 个结点的值：\n", n);
    for(i=1; i<=n; ++i) {
        scanf("%d", &e.key);
        InsertBST(T, e);
    }
    printf("先序遍历二叉排序树，结果是：\n");
    PreOrderTraverse(T, PrintElement);
    printf("\n");
    printf("中序遍历二叉排序树，结果是：\n");
    InOrderTraverse(T, PrintElement);
    printf("\n");
    printf("后序遍历二叉排序树，结果是：\n");
    PostOrderTraverse(T, PrintElement);
    printf("\n");
    printf("输入查找结点的值：\n");
    scanf("%d", &e.key);
    s = SearchBSTa(T, e.key);
    if (s) printf("已找到！结点的八进制地址为：%o\n", s);
    else printf("没找到！\n");
    printf("输入插入结点的值：\n", n);
    scanf("%d", &e.key);
```

```
    InsertBST(T, e);
    printf("先序遍历二叉排序树，结果是：\n");
    PreOrderTraverse(T, PrintElement);
    printf("\n");
    printf("中序遍历二叉排序树，结果是：\n");
    InOrderTraverse(T, PrintElement);
    printf("\n");
    printf("后序遍历二叉排序树，结果是：\n");
    PostOrderTraverse(T, PrintElement);
    printf("\n");
    printf("输入删除结点的值：\n", n);
    scanf("%d", &e.key);
    DeleteBST(T, e.key);
    printf("先序遍历二叉排序树，结果是：\n");
    PreOrderTraverse(T, PrintElement);
    printf("\n");
    printf("中序遍历二叉排序树，结果是：\n");
    InOrderTraverse(T, PrintElement);
    printf("\n");
    printf("后序遍历二叉排序树，结果是：\n");
    PostOrderTraverse(T, PrintElement);
    printf("\n");
}
```