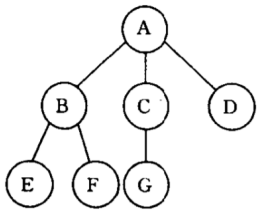
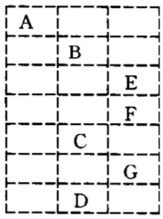
Homework 11.9

和泳毅 PB19010450

```c
//孩子-兄弟链表
typedef struct CSNode{
    char            data;//数据域
    struct CSNode   *fristchild,*nextsibling;//孩子、兄弟
}CSNode,*CSTree;
```

6.71 假设树上每个结点所含的数据元素为一个字母，并且以孩子—兄弟链表为树的存储结构，试写一个按凹入表方式打印一棵树的算法。例如:左下所示树印为右下形状。

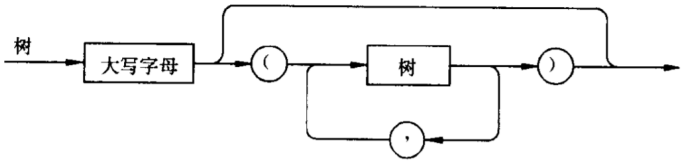




类C描述:

```c
Status PrintCSTree_1(CSTree T,int i){
//以孩子-兄弟链表按凹入表打印树,i初始为0用以控制缩进量,每一层次缩进相同
    if(T){//树存在
        for(n = 1;n <= 2 * i;n++)   printf(" ");//一缩进两空格
        printf("%c\n",T->data);
        PrintCSTree_1(T->firstchild,i + 1);
        PrintCSTree_1(T->nextsibling,i);
    }//if
    return OK;
}//PrintCSTree_1
```

6.74 试写一递归算法,以6.73题给定的树的广义表表示法的字符序列形式输出以孩子—兄弟链表表示的树。



6.71 题中的树可用下列形式的广义表表示：

$$A(B(E,F),C(G),D)$$

类C描述:

```c
Status PrintCSTree_2(CSTree T){
//以孩子-兄弟链表按广义表格式打印树的递归算法
```

```
    if(T){
        printf("%c",T->data);
        if(T->firstchild){
            printf("(");
            for(p = T->firstchild;p;p = p->nextsibling){
                PrintCSTree_2(p);
                if(p->nextsibling)  printf(",");
            }//for
            printf(")");
        }//if_2
    }//if_1
    return OK;
}//PrintCSTree_2
```

完整C实现:

```
#include <stdio.h>
#include <stdlib.h>
#include <malloc.h>
#define OVERFLOW    -1
#define OK  1
#define Status int

typedef struct CSNode{
    char            data;
    struct CSNode   *firstchild,*nextsibling;
}CSNode,*CSTree;

Status CreateCSTree(CSTree &T){
    char n;
    scanf("%c",&n);
    getchar();
    if(n == '#') T = NULL;
    else{
        T = (CSTree)malloc(sizeof(CSNode));
        if (!T){
            printf("建立二叉树时出错。\n");
            exit(OVERFLOW);
        }
        T->data = n;
        printf("输入%c的孩子节点：",n);
        CreateCSTree(T->firstchild);
        printf("输入%c的兄弟节点：",n);
        CreateCSTree(T->nextsibling);
    }
   return OK;
}

Status PrintCSTree_1(CSTree T,int i)
Status PrintCSTree_2(CSTree T)

int main(){
    CSTree T;
    printf("输入树的根节点：");
    CreateCSTree(T);
    printf("\n凹入表方式打印\n");
```

```
41      PrintCSTree_1(T,0);
42      printf("\n广义表方式打印\n");
43      PrintCSTree_2(T);
44      return 0;
45  }
```

结果测试: