#include<stdio.h>

#define MAXSIZE 100

typedef struct {

char element[MAXSIZE + 1];//静态数组存储元素

int maxNum;//二叉树中结点的最大编号

}BiTree;

void initBiTree(BiTree\*T){//初始化

T->maxNum=0;

for(int i=0;i<=MAXSIZE;i++){

T->element[i]='\0';

}

}

void clearBiTree(BiTree\*T){//适配器设计

initBiTree (T);

}

void createBiTree(BiTree\*T){

char elem;

int num;

int mn=0;

//a1b2c3

scanf("%c",&elem);

scanf("%d",&num);

while(num!=0){

T->element[num]=elem;

if(num>mn){

mn=num;

}

scanf("%c",&elem);

scanf("%d",&num);

}

T->maxNum=mn;

}

void outputBiTree(BiTree\*T){

int i=1;

int level=1;

int len=1;

int k;

while(i<T->maxNum){

//(1)控制输出起始的位置

for(k=0;k<20-2\*level;k++){

printf(" ");

}

//(2)按照这一行的个数输出这一层次的结点

for(k=0;k<len;k++){

printf("%4c",T->element[i]);

i++;

if(i>T->maxNum){

break;

}

}

//(3)换行 ,下一行元素个数计算

len \*=2;

printf("\n");

}

}

void preOrder(BiTree\*T,int rnum){

//(1)if是空 递归终结条件

if(rnum > T->maxNum || T->element[rnum]=='\0'){

return;

}

//(2)if非空 递归调用

int lenum=rnum\*2;

int rinum=rnum\*2+1;

printf("%4c",T->element[rnum]);

preOrder(T,lenum);

preOrder(T,rinum);

}

void inOrder(BiTree\*T,int rnum){

if(rnum > T->maxNum || T->element[rnum]=='\0'){

return ;

}

int lenum =2\*rnum;

int rinum =2\*rnum;

inOrder(T,lenum);

inOrder(T,rinum);

printf("%4c",T->element[rnum]);

}

int leaf(BiTree\* T,int rnum){

//条件1

if(rnum>T->maxNum || T->element[rnum]=='\0'){

return 0;

}

int lenum=rnum\*2;

int rinum =rnum\*2+1;

//条件2

if((lenum>T->maxNum || T->element[lenum])=='\0'&&(rinum>T->maxNum||T->element[rinum]=='\0'))

return 1;

}

int main(){

BiTree tree;

initBiTree(&tree);

createBiTree(&tree);

outputBiTree(&tree);

printf("preOrder: ");

preOrder(&tree,1);

printf("\n");

printf("preOrder: ");

inOrder(&tree,1);

printf("\n");

preOrder(&tree,1);

printf("preOrder: ");

preOrder(&tree,1);

printf("\n");

return 1;

}