

```

#include<stdio.h>
#include<stdlib.h>
#define MAXNODE 50
typedef struct Bitree {
    char a;
    Bitree *lchild;
    Bitree* rchild;
}Bitree;

//创建二叉树
void creatBitree(Bitree** T) {
    char ch;
    scanf_s("%c", &ch);
    if (ch == '@') (*T) = NULL;
    else {
        (*T)= (Bitree*)malloc(sizeof(Bitree));
        (*T)->a = ch;
        creatBitree(&(*T)->lchild);
        creatBitree(&(*T)->rchild);
    }
}

//访问结点
void Visit(Bitree* T) {
    printf("%c", T->a);
}

//前序遍历
void PreorderTraverse(Bitree* T) {
    if (T != NULL) {
        Visit(T);
        PreorderTraverse(T->lchild);
        PreorderTraverse(T->rchild);
    }
}

//中序遍历
void InorderTraverse(Bitree *T) {
    if (T != NULL) {
        InorderTraverse(T->lchild);
        Visit(T);
        InorderTraverse(T->rchild);
    }
}

//后序遍历
void PostorderTraverse(Bitree* T) {
    if (T != NULL) {
        PostorderTraverse(T->lchild);
        PostorderTraverse(T->rchild);
        Visit(T);
    }
}

//层序遍历
void LevelorderTraverse(Bitree* T) {
    Bitree* Queue[MAXNODE], * p = T;
    int front = 0, rear = 0;

```

```

        if (p != NULL) {
            Queue[++rear] = p;
            while (front < rear) {
                p = Queue[++front];
                Visit(p);
                if (p->lchild != NULL) Queue[++rear] = p->lchild;
                if (p->rchild != NULL) Queue[++rear] = p->rchild;
            }
        }
    }
}

int main() {
    Bitree* T;
    T = (Bitree*)malloc(sizeof(Bitree));
    printf("输入\n");
    creatBitree(&T);
    printf("前序遍历\n");
    PreorderTraverse(T);
    printf("\n中序遍历\n");
    InorderTraverse(T);
    printf("\n后序遍历\n");
    PostorderTraverse(T);
    printf("\n层序遍历\n");
    LevelorderTraverse(T);
    return 0;
}

```