## 2、

程序代码：

#include <stdio.h>

#include <stdlib.h>

typedef struct bn

{

char d;

struct bn \*lc, \*rc;

}BN;

BN \*create()

{

char ch;

BN \*bt;

ch=getchar();

if(ch=='#')return(NULL);

else

{

bt=(BN\*)malloc(sizeof(BN));

bt->d=ch;

bt->lc=create();

bt->rc=create();

return (bt);

}

}

void inorder(BN \*t)

{

if(t)

{

inorder(t->lc);

printf("%c",t->d);

inorder(t->rc);

}

}

void postorder(BN \*t)

{

if(t)

{

postorder(t->lc);

postorder(t->rc);

printf("%c",t->d);

}

}

int main()

{

BN \*bn;

bn=create();

inorder(bn);

printf("\n");

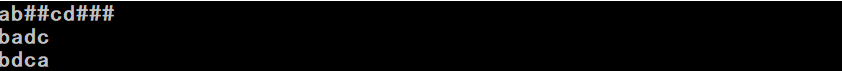
postorder(bn);

printf("\n");

return 0;

}

运行结果截图：



## 3、

程序代码：

#include <stdio.h>

#include <stdlib.h>

#define m 1024

typedef struct bn

{

char d;

struct bn \*lc, \*rc;

}BN;

typedef BN \*QE;

typedef struct qn

{

QE data[m];

int f, r;

}QN;

void Init(QN \*q)

{

q->f=0;

q->r=0;

}

int Empty(QN \*q)

{

if(q->r==q->f)

return 1;

return 0;

}

QE Head(QN \*q)

{

if(q->f==q->r)

return NULL;

else

return (q->data[(q->f+1)%m]);

}

void EQ(QN \*q, QE x)

{

if((q->r+1)%m==q->f)

printf("Overflow\n");

else

{

q->r=(q->r+1)%m;

q->data[q->r]=x;

}

}

QE DQ(QN \*q)

{

if(q->f==q->r)

return NULL;

else

{

q->f=(q->f+1)%m;

return (q->data[q->f]);

}

}

BN \*create()

{

char ch;

BN \*bt;

ch=getchar();

if(ch=='#')return(NULL);

else

{

bt=(BN\*)malloc(sizeof(BN));

bt->d=ch;

bt->lc=create();

bt->rc=create();

return (bt);

}

}

void inorder(BN \*t)

{

if(t)

{

inorder(t->lc);

printf("%c",t->d);

inorder(t->rc);

}

}

void postorder(BN \*t)

{

if(t)

{

postorder(t->lc);

postorder(t->rc);

printf("%c",t->d);

}

}

void LT(BN \*t)

{

QN \* q;

BN \* p;

q=(QN\*)malloc(sizeof(QN));;

Init(q);

p=t;

if(p==NULL)

printf("树为空！\n") ;

EQ(q,p) ;

while(!Empty(q))

{

p = NULL ;

p=DQ(q);

if(p->lc!=NULL)

EQ(q,p->lc) ;

if(p->rc!=NULL)

EQ(q,p->rc) ;

printf("%c", p->d) ;

}

}

int main()

{

BN \*bn;

bn=create();

inorder(bn);

printf("\n");

postorder(bn);

printf("\n");

LT(bn);

printf("\n");

return 0;

}

运行结果及截图：

