以下为提高要求中的代码，后缀表达式变前缀表达式

/\*

表达式的三种形式对应的是一颗二叉树的三种遍历方式

\*/

#include <iostream>

#include <fstream>

using namespace std;

typedef struct node

{

char data;

struct node \*lchild, \*rchild;

}\*BiT,BitNode;

/\*

后序遍历+中序遍历建立二叉树

\*/

BiT crtBT2(char \*bk, int &n, char \*mi, int u, int v)

{

BiT bt;

int k;

if(u > v)

return NULL;

bt = new BitNode;

bt->data = bk[n--];

for(k = u; k <= v; k++)

if(mi[k] == bt->data)

break;

bt->rchild = crtBT2(bk, n, mi, k+1, v);

bt->lchild = crtBT2(bk, n, mi, u, k-1);

return bt;

}

void preOrderTraverse(BiT bt)

{

if(bt)

{

cout << bt->data;

preOrderTraverse(bt->lchild);

preOrderTraverse(bt->rchild);

}

}

int main()

{

char middle[100] = "a+b\*c-d-e/f";

char bk[100] = "abcd-\*+ef/-";//后缀表达式

int i = 0;

int n = 10;

BiT root;

root = crtBT2(bk, n, middle, 0, 10);

preOrderTraverse(root);

return 0;

}