实验五 线索二叉树

一、实验环境：

Windows 10 64位 MinGW GCC 4.8.1

IDEA：CLoin

二、实验内容：

编写程序实现一棵二叉树，并将其线索化。

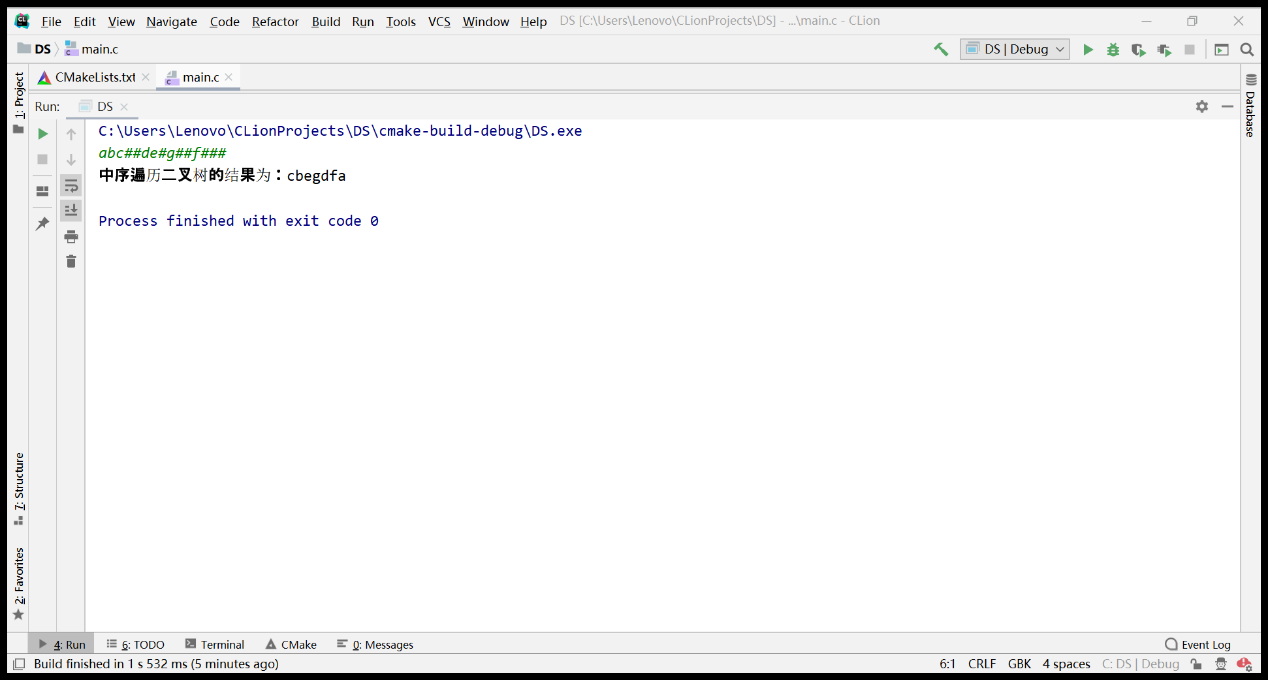
三、算法流程：

通过输入一颗二叉树的中序序列来构建一颗二叉树（#代表空节点，非#元素代表非空节点），并将其线索化，以便于使用迭代的方法遍历这颗二叉树。

四、程序代码：

#include **<stdio.h>**#include **<stdlib.h>**#define **TRUE** 1  
#define **FALSE** 0  
#define **OK** 1  
#define **ERROR** 0  
#define **INFEASIBLE** -1  
#define **OVERFLOW** -2  
**typedef int** Status;  
**typedef char** TElemType;  
**typedef enum** { ***Link***, ***Thread*** } PointerTag;  
**typedef struct** BiThrNode {  
 TElemType data;  
 **struct** BiThrNode \*lchild, \*rchild;  
 PointerTag LTag;  
 PointerTag RTag;  
}BiThrNode, \*BiThrTree;  
Status CreateBiThrNode(BiThrTree \* B) {  
 **char** ch;  
 scanf(**"%c"**, &ch);  
 **if**(ch==**'#'**)   
 \*B = **NULL**;  
 **else**{  
 **if**(!((\*B) = (BiThrNode \*)malloc(**sizeof**(BiThrNode))))   
 exit(**OVERFLOW**);  
 (\*B)->data = ch;  
 (\*B)->LTag = ***Link***;  
 (\*B)->RTag = ***Link***;  
 CreateBiThrNode(&(\*B)->lchild);  
 CreateBiThrNode(&(\*B)->rchild);  
 }  
 **return OK**;  
}  
**void** InThreading(BiThrTree B,BiThrTree \*pre) {  
 **if**(!B)   
 **return**;  
 InThreading(B->lchild,pre);  
 **if**(!B->lchild){  
 B->LTag = ***Thread***;  
 B->lchild = \*pre;  
 }  
 **if**(!(\*pre)->rchild){  
 (\*pre)->RTag = ***Thread***;  
 (\*pre)->rchild = B;  
 }  
 \*pre = B;  
 InThreading(B->rchild,pre);  
}  
Status InOrderThreading(BiThrTree \*Thrt,BiThrTree T){  
 **if**(!(\*Thrt = (BiThrTree)malloc(**sizeof**(BiThrNode))))   
 exit(**OVERFLOW**);  
 (\*Thrt)->LTag = ***Link***;  
 (\*Thrt)->RTag = ***Thread***;  
 (\*Thrt)->rchild = (\*Thrt);  
 **if**(!T) {  
 (\*Thrt)->lchild = (\*Thrt);  
 **return OK**;   
 }  
 BiThrTree pre;  
 pre = (\*Thrt);  
 (\*Thrt)->lchild = T;  
 InThreading(T,&pre);  
 pre->rchild = \*Thrt;  
 pre->RTag = ***Thread***;  
 (\*Thrt)->rchild = pre;  
 **return OK**;  
}  
Status InOrderTraverse(BiThrTree T) {  
 BiThrNode \*p = T->lchild;  
 **while**(p!=T){  
 **while**(p->LTag==***Link***)   
 p = p->lchild;   
 printf(**"%c"**,p->data );  
 **while**(p->RTag==***Thread*** && p->rchild!=T) {   
 p = p->rchild;  
 printf(**"%c"**,p->data );  
 }  
 p = p->rchild;  
 }  
 **return OK**;  
}  
**int** main() {  
 BiThrTree B,T;  
 CreateBiThrNode(&B);  
 InOrderThreading(&T,B);  
 printf(**"中序遍历二叉树的结果为："**);  
 InOrderTraverse(T);  
 printf(**"\n"**);  
 **return** 0;  
}

五、代码运行结果：



六、注意事项：

在构建线索二叉树的时候为充分利用空链域将左空链域设定为该节点的前驱，右链域空设定为该节点的后继，为保证不混淆正常子节点和线索化的节点应该做个标记。