# 树和二叉树

**作业无截止时间，但你们懂的哦，越早越好哦，太晚会被点名的哦！小可爱们，加油哟！**

**认真做，一定是自己独立完成的，不懂的可以看书可以问，但不能应付我哦。**

**一、单项选择题**

1. 以下说法错误的是 ( AE )

A．树形结构的特点是一个结点可以有多个直接前趋

B．线性结构中的一个结点至多只有一个直接后继

C．树形结构可以表达(组织)更复杂的数据

D．树(及一切树形结构)是一种"分支层次"结构

E．任何只含一个结点的集合是一棵树

2．下列说法中正确的是 ( D )

A．任何一棵二叉树中至少有一个结点的度为2

B．任何一棵二叉树中每个结点的度都为2

C．任何一棵二叉树中的度肯定等于2

D．任何一棵二叉树中的度可以小于2

3．讨论树、森林和二叉树的关系，目的是为了（ A ）

A．借助二叉树上的运算方法去实现对树的一些运算

B．将树、森林按二叉树的存储方式进行存储

C．将树、森林转换成二叉树

D．体现一种技巧，没有什么实际意义

4．树最适合用来表示 ( C )

A．有序数据元素 B．无序数据元素

C．元素之间具有分支层次关系的数据 D．元素之间无联系的数据

5．若一棵二叉树具有10个度为2的结点，5个度为1的结点，则度为0的结点个数是（ B ）

A．9 B．11 C．15 D．不确定

6．设森林F中有三棵树，第一，第二，第三棵树的结点个数分别为M1，M2和M3。与森林F对应的二叉树根结点的右子树上的结点个数是（ C ）。

A．M1 B．M1+M2 C．M3 D．M2+M3

7．一棵完全二叉树上有1001个结点，其中叶子结点的个数是（ E ）

A． 250 B． 500 C．254 D．505 E．以上答案都不对

8. 设给定权值总数有n 个，其哈夫曼树的结点总数为( D )

A．不确定 B．2n C．2n+1 D．2n-1

9．二叉树的第I层上最多含有结点数为（ A ）

A．2I  B． 2I-1-1 C． 2I-1 D．2I  -1

10．一棵二叉树高度为h,所有结点的度或为0，或为2，则这棵二叉树最少有( B )结点

A．2h B．2h-1 C．2h+1 D．h+1

11. 利用二叉链表存储树，则根结点的右指针是（ C ）。

A．指向最左孩子 B．指向最右孩子 C．空 D．非空

12．已知一棵二叉树的前序遍历结果为ABCDEF,中序遍历结果为CBAEDF,则后序遍历的结果为（ A ）。

A．CBEFDA B． FEDCBA C． CBEDFA D．不定

13．已知某二叉树的后序遍历序列是dabec, 中序遍历序列是debac , 它的前序遍历是（ D ）。

A．acbed B．decab C．deabc D．cedba

14．在二叉树结点的先序序列，中序序列和后序序列中，所有叶子结点的先后顺序（ C ）

A．都不相同 B．完全相同

C．先序和中序相同，而与后序不同 D．中序和后序相同，而与先序不同

15．在完全二叉树中，若一个结点是叶结点，则它没（ C ）。

A．左子结点 B．右子结点

C．左子结点和右子结点 D．左子结点，右子结点和兄弟结点

16．在下列情况中，可称为二叉树的是（ BCD ）

A．每个结点至多有两棵子树的树

B. 哈夫曼树

C．每个结点至多有两棵子树的有序树

D. 每个结点只有一棵右子树

E．以上答案都不对

17．由3 个结点可以构造出多少种不同的二叉树？（ D ）

A．2 B．3 C．4 D．5

18．下面几个符号串编码集合中，不是前缀编码的是（ D ）。

A．{0,10,110,1111} B．{11,10,001,101,0001}

C．{00,010,0110,1000} D．{b,c,aa,ac,aba,abb,abc}

19. 一棵有n个结点的二叉树，按层次从上到下，同一层从左到右顺序存储在一维数组A[1..n]中，则二叉树中第i个结点（i从1开始用上述方法编号）的右孩子在数组A中的位置是（ B ）

A．A[2i](2i<=n) B．A[2i+1](2i+1<=n)

C．A[i-2] D．条件不充分，无法确定

20、以下说法错误的是 ( C )

A．哈夫曼树是带权路径长度最短的树，路径上权值较大的结点离根较近。

B．若一个二叉树的树叶是某子树的中序遍历序列中的第一个结点，则它必是该子树的后序遍历序列中的第一个结点。

C．已知二叉树的前序遍历和后序遍历序列并不能惟一地确定这棵树，因为不知道树的根结点是哪一个。

D．在前序遍历二叉树的序列中，任何结点的子树的所有结点都是直接跟在该结点的之后。

**二、判断题（在各题后填写“√”或“×”）**

1. 完全二叉树一定存在度为1的结点。( **×** )

2．对于有N个结点的二叉树，其高度为log2n。( **×** )

3. 二叉树的遍历只是为了在应用中找到一种线性次序。( **×** )

4. 一棵一般树的结点的前序遍历和后序遍历分别与它相应二叉树的结点前序遍历和后序遍历是一致的。( **√** )

5. 用一维数组存储二叉树时，总是以前序遍历顺序存储结点。( **×** )

6．中序遍历一棵二叉排序树的结点就可得到排好序的结点序列。 ( **×** )

7．完全二叉树中，若一个结点没有左孩子，则它必是树叶。( **×** )

8. 二叉树只能用二叉链表表示。( **×** )

9. 给定一棵树，可以找到唯一的一棵二叉树与之对应。( **×** )

10. 用链表(llink-rlink)存储包含n个结点的二叉树，结点的2n个指针区域中有n-1个空指针。( **×** )

11．树形结构中元素之间存在一个对多个的关系。( **√** )

12．将一棵树转成二叉树，根结点没有左子树。( **×** )

13．度为二的树就是二叉树。( **×** )

14. 二叉树中序线索化后，不存在空指针域。( **×** )

15．霍夫曼树的结点个数不能是偶数。( **√** )

16．哈夫曼树是带权路径长度最短的树，路径上权值较大的结点离根较近。( **√** )

**三、填空题**

1．在二叉树中，指针p所指结点为叶子结点的条件是\_结点无左右孩子\_\_ \_\_\_。

2．深度为k的完全二叉树至少有\_\_\_ k \_\_\_\_个结点，至多有\_\_\_ 2^k-1 \_\_\_\_个结点。

3．高度为8的完全二叉树至少有\_\_\_2^6+1\_\_\_个叶子结点。

4.具有n个结点的二叉树中,一共有\_\_\_\_2n\_\_\_\_个指针域,其中只有\_\_\_\_n-1\_\_\_\_个用来指向结点的左右孩子，其余的\_\_\_n+1\_\_\_\_个指针域为NULL。

5．树的主要遍历方法有\_\_\_\_前序遍历\_\_\_\_、\_\_\_\_中序遍历\_\_\_\_、\_\_\_\_后序遍历\_\_\_\_等三种。

6．一个深度为k的，具有最少结点数的完全二叉树按层次，（同层次从左到右）用自然数依此对结点编号，则编号最小的叶子的序号是\_\_ 2^(k-1) \_；编号是i的结点所在的层次号是\_ [logi]+1 \_\_（根所在的层次号规定为1层）。

7．如果结点A有 3个兄弟，而且B是A的双亲，则B的度是\_\_\_4\_\_\_。

8．二叉树的先序序列和中序序列相同的条件是\_\_\_ 空树||子树只有右结点或无结点的树 \_\_\_。

9．若一个二叉树的叶子结点是某子树的中序遍历序列中的最后一个结点，则它必是该子树的\_\_\_\_ 前序遍历 \_\_序列中的最后一个结点。

10．若以{4，5，6，7，8}作为叶子结点的权值构造哈夫曼树，则其带权路径长度是\_\_\_4\_\_\_。

11．以下程序段采用先根遍历方法求二叉树的叶子数，请在横线处填充适当的语句。

Void countleaf(bitreptr t,int &count)/\*根指针为t，假定叶子数count的初值为0\*/

{

if(t!=NULL)

{

if((t->lchild==NULL)&&(t->rchild==NULL))\_\_\_\*count++\_\_\_\_\_;

countleaf(t->lchild,count);

countleaf(t->rchild,count);\_\_\_\_

}

}

12．以下程序是二叉链表树中序遍历的非递归算法，请填空使之完善。二叉树链表的结点类型的定义如下：

typedef struct node /\*C语言/

{char data; struct node \*lchild,\*rchild;}\*bitree;

void vst(bitree bt) /\*bt为根结点的指针\*/

{ bitree p; p=bt; initstack(s); /\*初始化栈s为空栈\*/

while(p || !empty(s)) /\*栈s不为空\*/

if(p) { push (s,p); (1)\_\_\_ p=p->lchild ; } /\*P入栈\*/

else { p=pop(s); printf(“%c”,p->data);(2)\_\_ p=p->rchild \_\_; }/\*栈顶元素出栈\*/

}

13．二叉树存储结构同上题，以下程序为求二叉树深度的递归算法，请填空完善之。

int depth(bitree bt) /\*bt为根结点的指针\*/

{

int hl,hr;

if (bt==NULL)

return((1)\_ 0 \_\_);

hl=depth(bt->lchild);

hr=depth(bt->rchild);

if((2)\_ hr<hl \_\_)

(3)\_ hr=hl \_\_\_\_；

return(hr+1);

}

14．将二叉树bt中每一个结点的左右子树互换的C语言算法如下，其中ADDQ(Q,bt),DELQ(Q),EMPTY(Q)分别为进队，出队和判别队列是否为空的函数，请填写算法中得空白处，完成其功能。

typedef struct node

{

int data ;

struct node \*lchild, \*rchild;

}btnode;

void EXCHANGE(btnode \*bt)

{

btnode \*p, \*q;

if (bt){ADDQ(Q,bt);

while(!EMPTY(Q))

{p=DELQ(Q); q=(1)\_ p->rchild \_\_; p->rchild=(2)\_ p->lchild \_\_; (3)\_\_p->lchild \_=q;

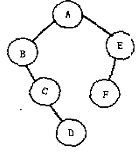
if(p->lchild) (4)\_ EXCHANGE(p->lchild) \_\_; if(p->rchild) (5)\_ \_EXCHANGE(p->rchild)\_;

}

} }

**四、应用题**

1.分别给出下图所示二叉树的先根、中根和后根序列。



前序：ABCDEF

中序：BCDAFE

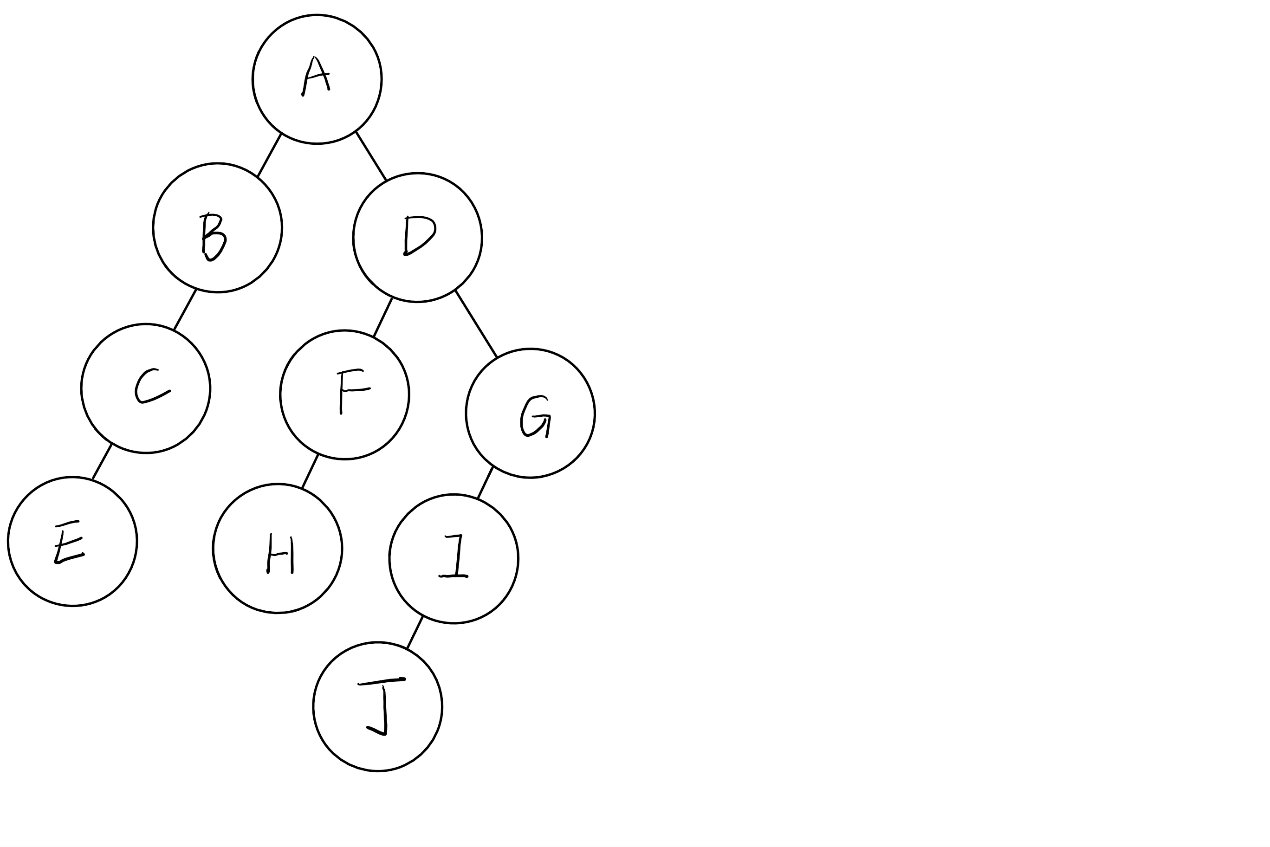
后序：DCBFEA

2.设二叉树BT的存储结构如下:

1 2 3 4 5 6 7 8 9 10

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Lchild | 0 | 0 | 2 | 3 | 7 | 5 | 8 | 0 | 10 | 1 |
| Data | J | H | F | D | B | A | C | E | G | I |
| Rchild | 0 | 0 | 0 | 9 | 4 | 0 | 0 | 0 | 0 | 0 |

其中BT为树根结点的指针，其值为6,Lchild,Rchild分别为结点的左、右孩子指针域,data为结点的数据域。试完成下列各题:

（l）画出二叉树BT的逻辑结构;

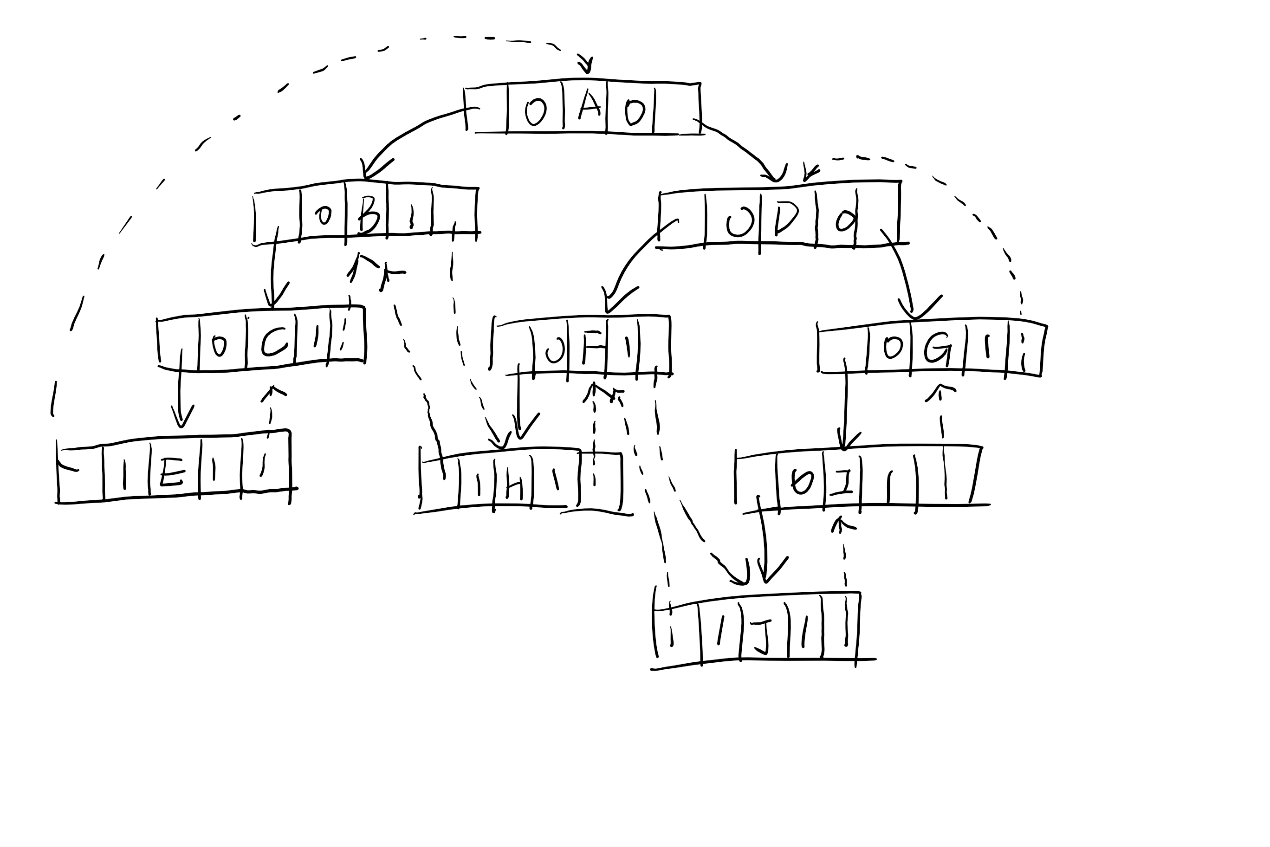
（2）写出按前序、中序、后序遍历该二叉树所得到的结点序列;

前序：ABCEDFHGIJ

中序：ECBAHFDJIG

后序：ECBHFJIGDA

（3）画出二叉树的后序线索树。



3.设有正文AADBAACACCDACACAAD,字符集为A,B,C,D,设计一套二进制编码，使得上述正文的编码最短。（要求左孩子权值小于等于右孩子）

A:1

B:000

C:01

D:001

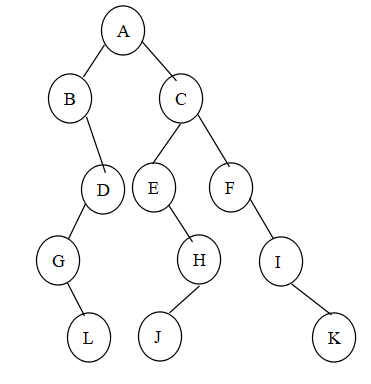
**五、编程题**

**1.建立二叉树并实现各种遍历**

**要求：**

**预期效果及实际效果**

**读入的二叉树图为：**



**简化为：A(B(,D(G(,L),)),C(E(,H(J,)),F(,I(,K))))**

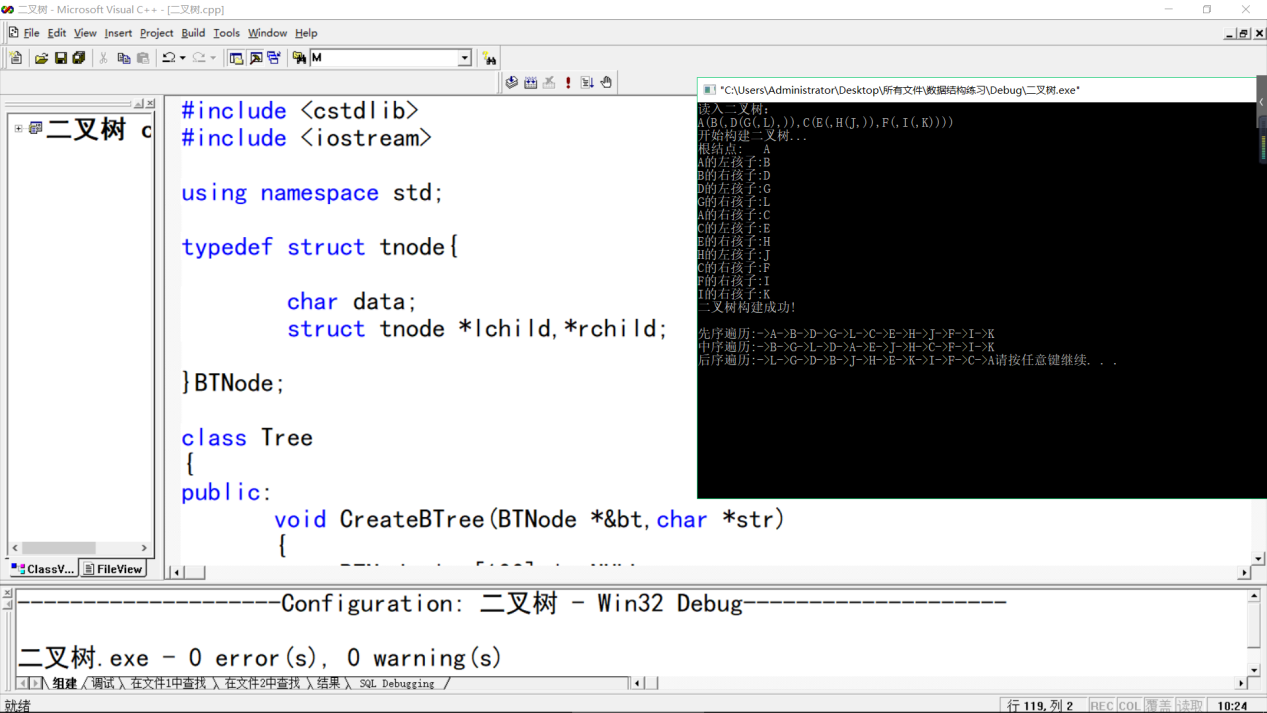
**预期效果：**

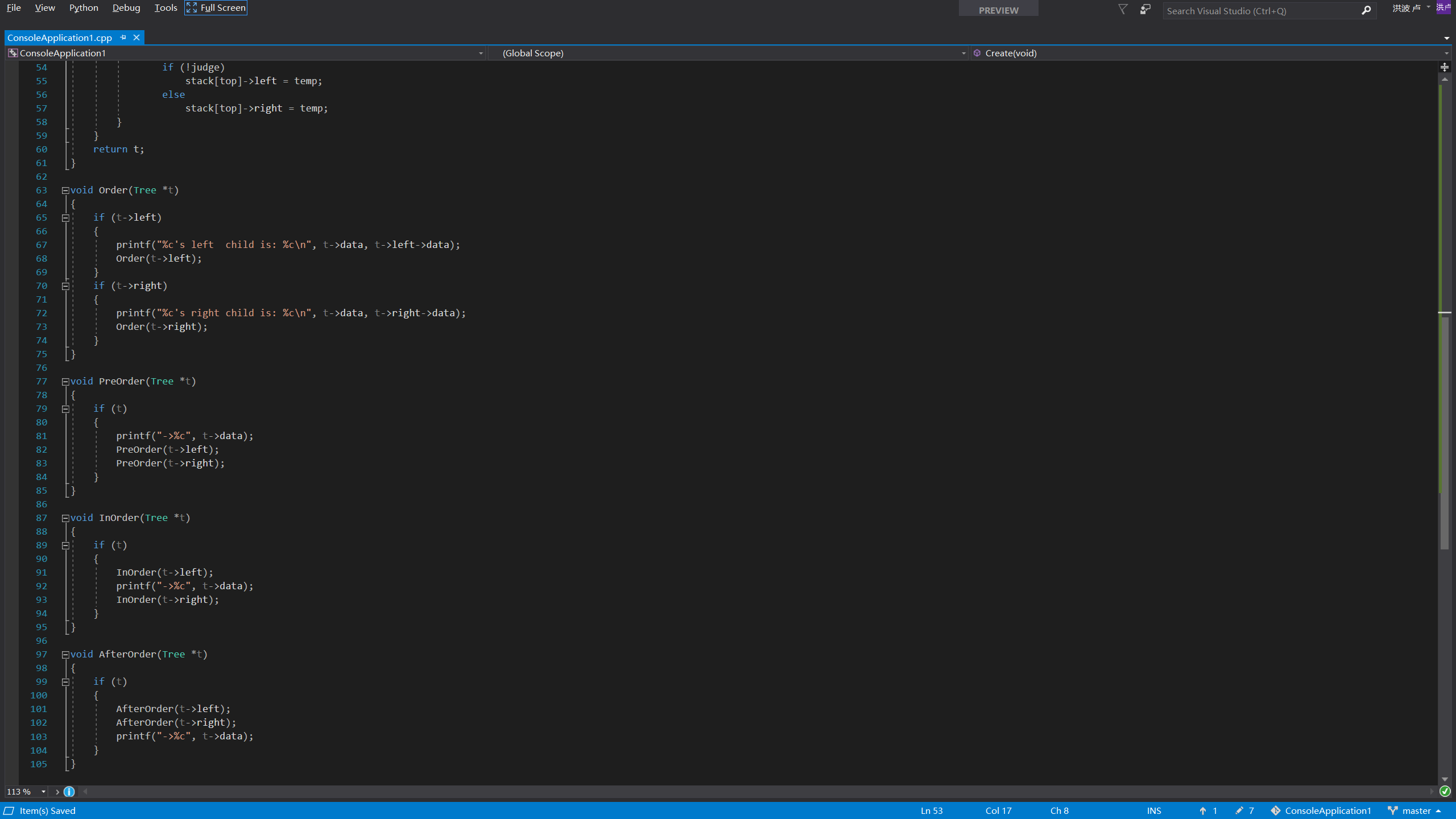
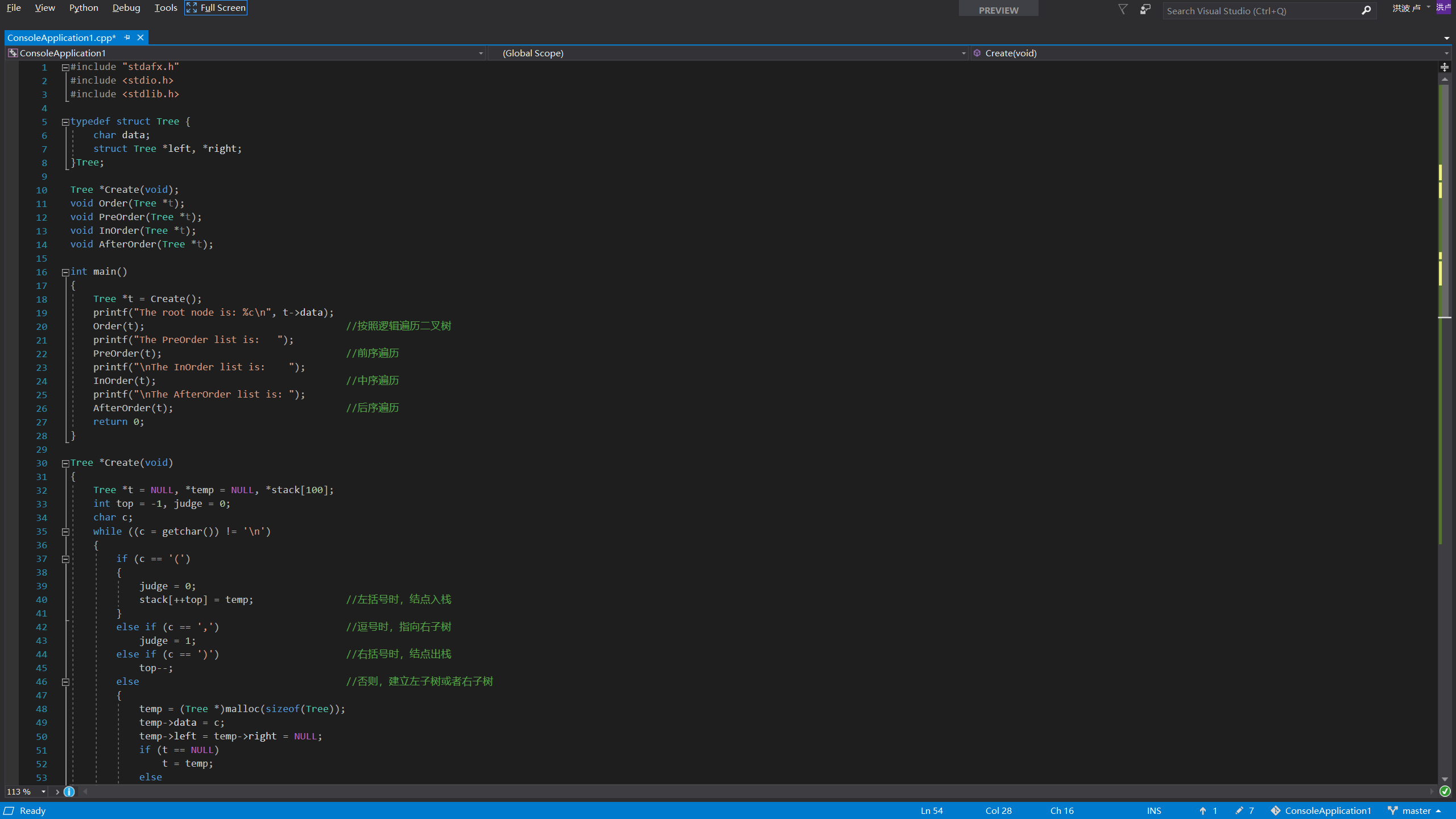
**先序遍历：A->B->D->G->L->C->E->H->J->F->I->K**

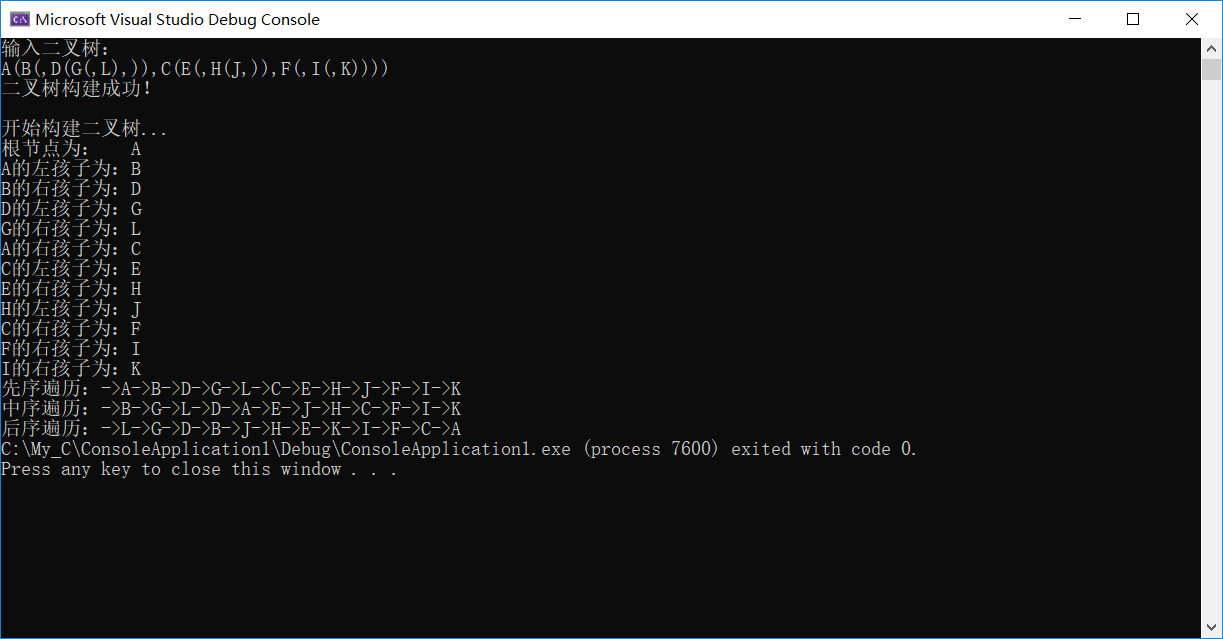
**中序遍历：B->G->L->D->A->E->J->H->C->F->I->K**

**后序遍历：L->G->D->B->J->H->E->K->I->F->C->A**

**实际效果：**







1. 利用二叉树结构实现赫夫曼编译码器

**测试运行**

若字符集个数为8个，每个字符出现概率为：5%,29%,7%,8%,14%,23%,3%,11%。请运行程序输出每个字符的赫夫曼编码

**实际效果：**

