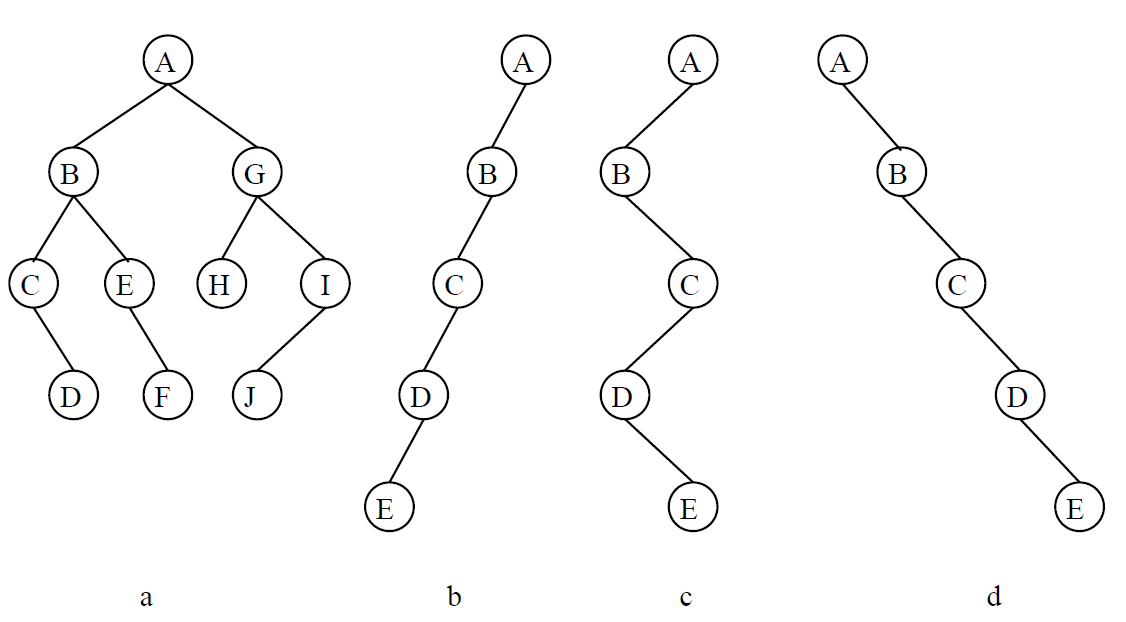
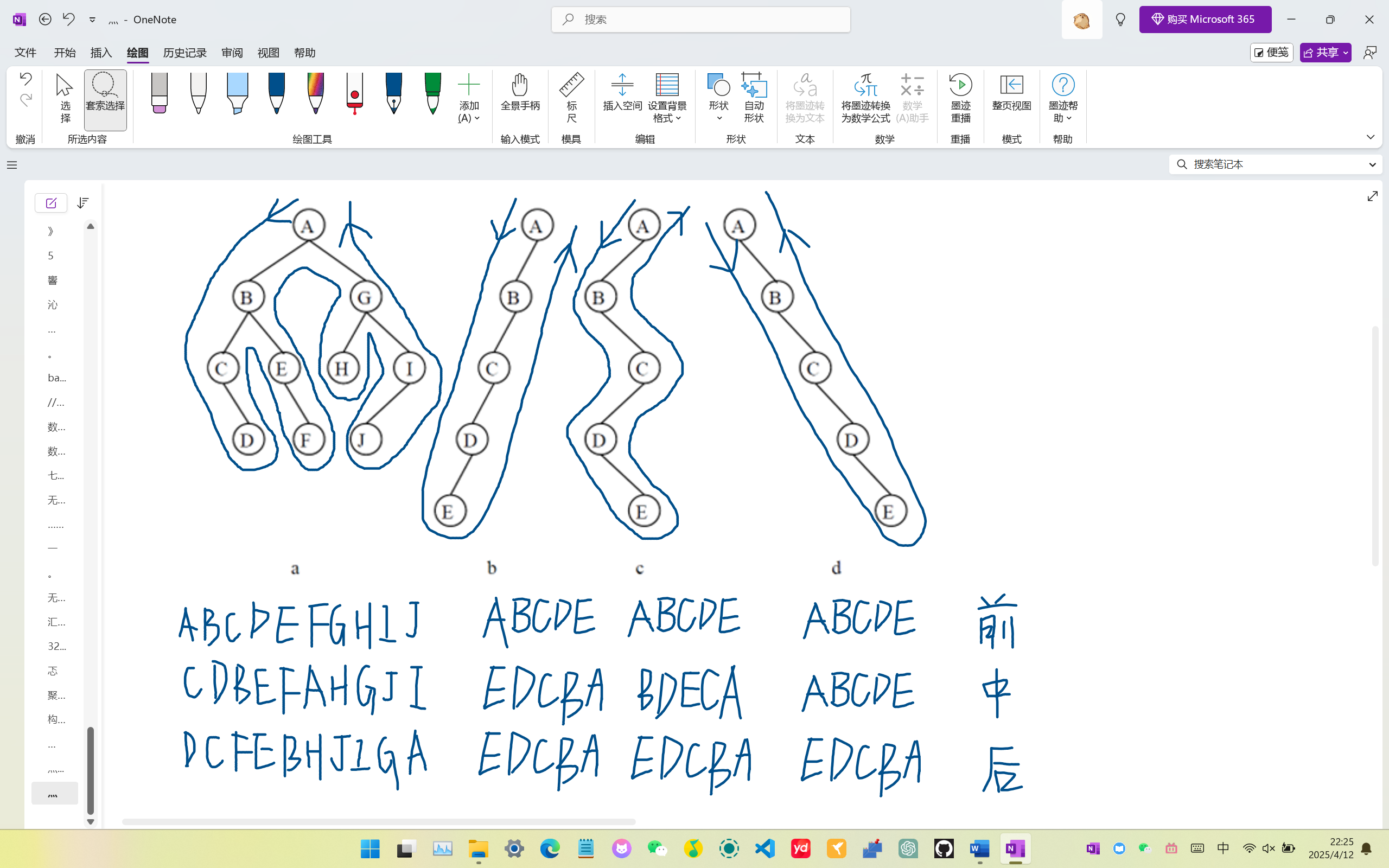
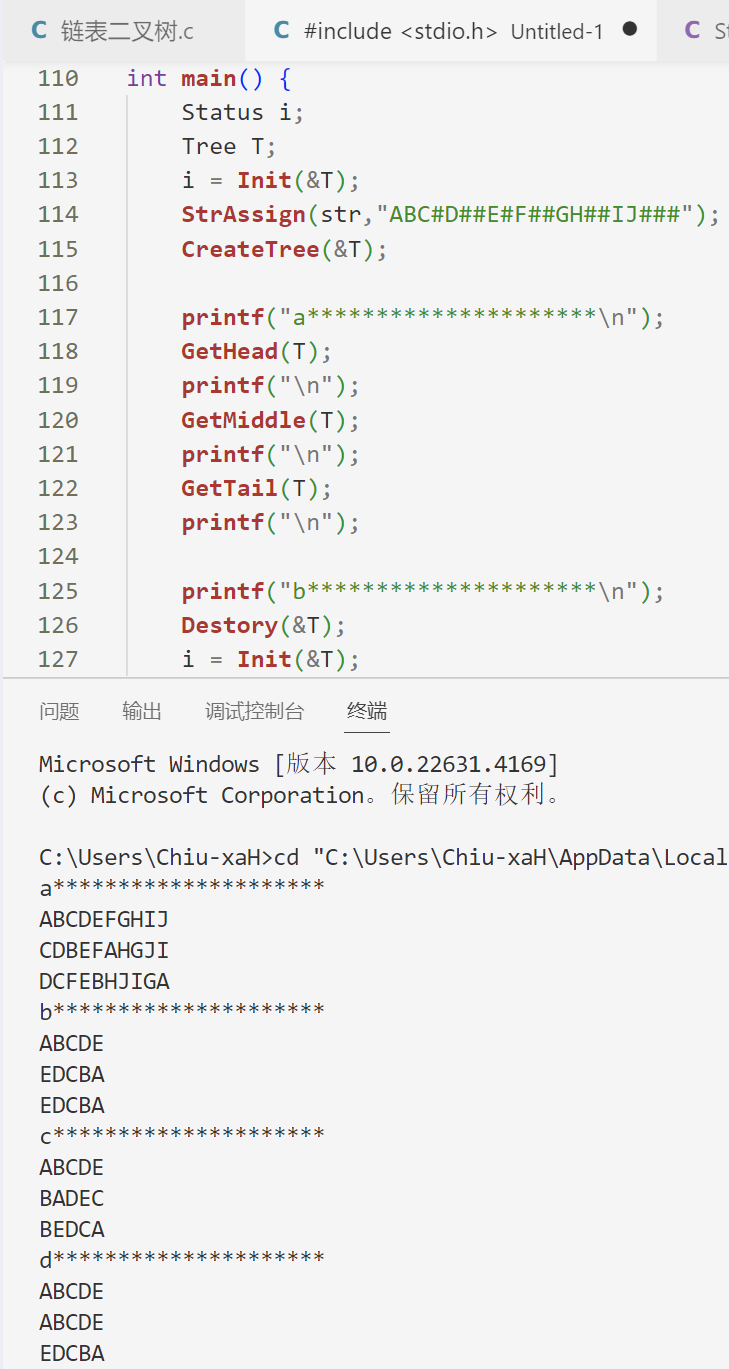
5.8 分别求出下图中二叉树的三种遍历序列。



代码验证



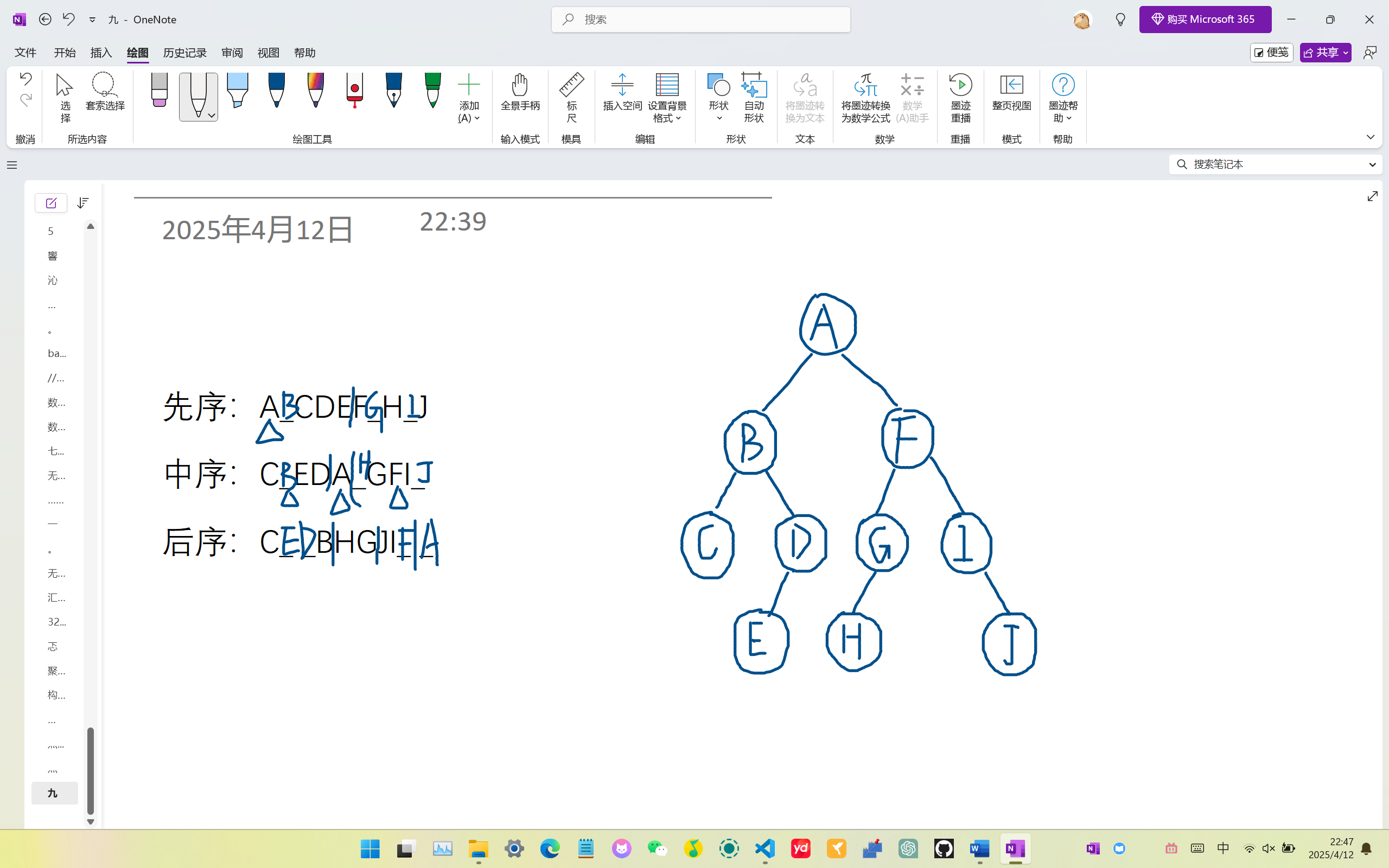
结果一致

5.12 已知一棵二叉树的先序、中序和后序序列如下，其中各有一部分未给出其值，请构造出该二叉树。

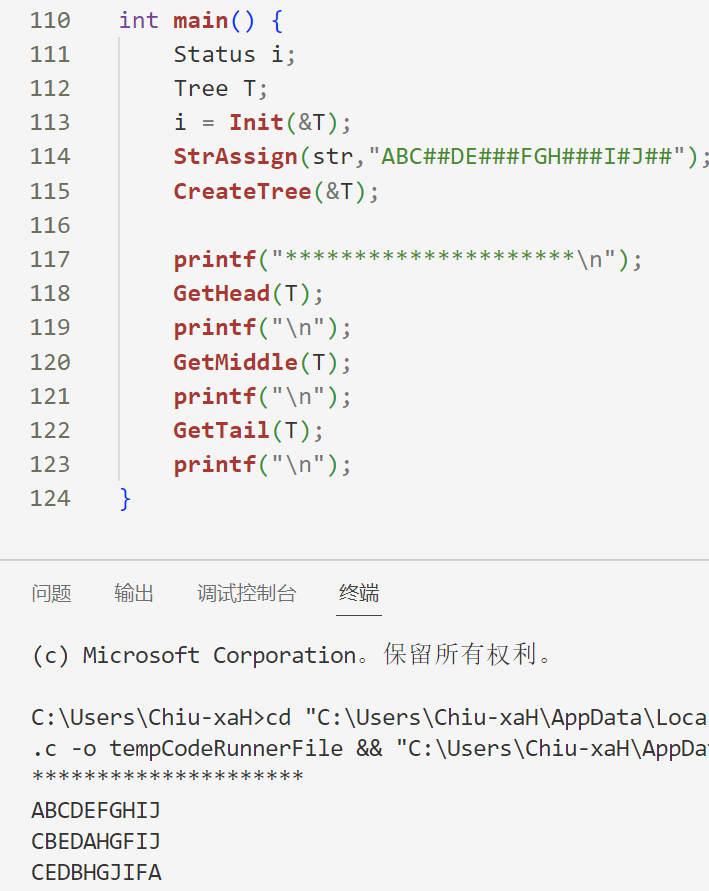
先序：A\_CDEF\_H\_J

中序：C\_EDA\_GFI\_

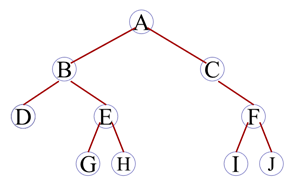
后序：C\_ \_BHGJI\_ \_



代码验证：



结果一致

5.15 设计算法按中序次序依次输出各结点的值及其对应的序号。例如，下图中的二叉树的输出结果是(D,1) (B,2) (G,3) (E,4) (H,5) (A,6) (C,7) (I,8) (F,9) (J,10)。

思路：改造现有的中序遍历算法，加入num递增递归传递

代码：

// 递归子函数

Status MiddleWithNum(Tree T, int \*num) {

if (T != NULL) {

MiddleWithNum(T->leftChild, num);

printf("(%c,%d) ", T->data, (\*num)++);

MiddleWithNum(T->rightChild, num);

return OK;

} else return ERROR;

}

// 起始点

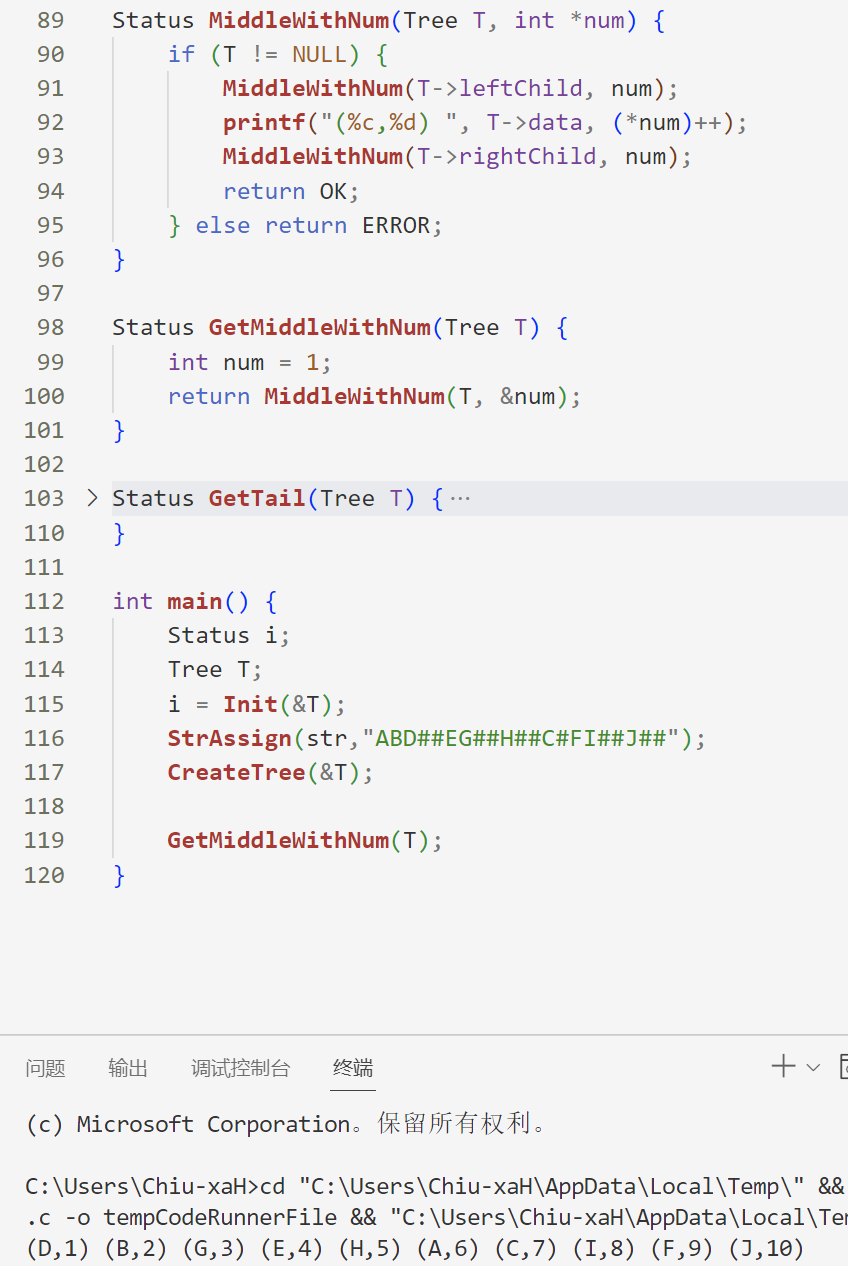
Status GetMiddleWithNum(Tree T) {

int num = 1;

return MiddleWithNum(T, &num);

}

运行测试：



结果一致