

## 软工23C1 《数据结构与算法》 - 栈和队列

开始时间 1970/01/01 08:00:00 结束时间 1970/01/01 08:00:00 答题时长 0分钟

答卷类型 标准答案

试卷总分 173

## 判断题

23 分

1-1 若一个结点是某二叉树的中序遍历序列的最后一个结点，则它必是该树的前序遍历序列中的最后一个结点。

1分

☐ T☒ F

| 参考答案

答案 F

1-2 若 A 和 B 都是一棵二叉树的叶子结点，则存在这样的二叉树，其前序遍历序列为 ...A...B...，而中序遍历序列为 ...B...A...。

1分

☐ T☒ F

| 参考答案

答案 F

1-3 某二叉树的前序和中序遍历序列正好一样，则该二叉树中的任何结点一定都无左孩子。

1分

☒ T☐ F

| 参考答案

答案 T

1-4 存在一棵总共有2016个结点的二叉树，其中有16个结点只有一个孩子。

1分

☐ T☒ F

| 参考答案

答案 F

1-5 一棵有124个结点的完全二叉树，其叶结点个数是确定的。

1分

☒ T☐ F

| 参考答案

答案 T

1-6 具有10个叶结点的二叉树中，有9个度为2的结点。

1分

☒ T ☐ F

| 参考答案

答案 T

1-7 完全二叉树中，若一个结点没有左孩子，则它必是树叶。

1分

☒ T ☐ F

| 参考答案

答案 T

1-8 已知一棵二叉树的先序遍历结果是ABC，则CAB不可能是中序遍历结果。

1分

☒ T ☐ F

| 参考答案

答案 T

1-9 若一个结点是某二叉树的中序遍历序列的最后一个结点，则它必是该树的前序遍历序列中的最后一个结点。

1分

☐ T ☒ F

| 参考答案

答案 F

1-10 某二叉树的后序和中序遍历序列正好一样，则该二叉树中的任何结点一定都无右孩子。

1分

☒ T ☐ F

| 参考答案

答案 T

1-11 某二叉树的前序和中序遍历序列正好一样，则该二叉树中的任何结点一定都无右孩子。

1分

☐ T ☒ F

| 参考答案

答案 F

1-12 将一棵完全二叉树存于数组中（根结点的下标为1）。则下标为23和24的两个结点是兄弟。

1分

☐ T ☒ F

| 参考答案

答案 F

1-13 完全二叉树中，若一个结点没有左孩子，则它必是树叶。

1分

☒ T ☐ F

| 参考答案

答案 T

1-14 二叉树只能用二叉链表表示。

1分

☐ T ☒ F

| 参考答案

答案 F

1-15 给定一棵树，可以找到唯一的一棵二叉树与之对应。

1分

☒ T ☐ F

| 参考答案

答案 T

1-16 将一棵树转成二叉树，根结点没有左子树。

1分

☐ T ☒ F

| 参考答案

答案 F

1-17 度为二的树就是二叉树。

1分

☐ T ☒ F

| 参考答案

答案 F

1-18 哈夫曼树的结点个数不能是偶数。

1分

☒ T

☐ F

| 参考答案

答案 T

1-19 存在一棵总共有2019个结点的二叉树，其中有16个结点只有一个孩子。

1分

☒ T

☐ F

| 参考答案

答案 T

1-20 堆是完全二叉树，完全二叉树不一定是堆。（）

1分

☒ T

☐ F

| 参考答案

答案 T

1-21 二叉排序树的后序遍历序列必然是递增的。

1分

☐ T

☒ F

| 参考答案

答案 F

1-22 在二叉排序树中插入一个新结点，总是插入到叶子结点下面。

1分

☐ T

☒ F

| 参考答案

答案 F

1-23 每个结点的关键字都比左孩子关键字大，比右孩子关键字小，这样的二叉树一定是二叉排序树。

1分

☐ T

☒ F

| 参考答案

答案 F

单选题

70 分

2-1 一棵有47个结点的树一定有\_\_\_\_条边。

2分

- ☐ A. 47      ☒ B. 46      ☐ C. 92      ☐ D. 23

| 参考答案

答案      B

2-2 若二叉树 T 中有 244 个结点 (叶结点的高度为 1), 则 T 的高度至少是

2分

- ☐ A. 8      ☐ B. 7      ☒ C. 6      ☐ D. 5

| 参考答案

答案      C

2-3 一棵度为4的树T中, 有20个度为4的结点, 10个度为3的结点, 1个度为2的结点, 10个度为1的结点, 则树T的叶结点个数是 ( )

2分

- ☐ A. 41  
☒ B. 82  
☐ C. 113  
☐ D. 122

| 参考答案

答案      B

2-4 一棵满二叉树中127个节点, 其中叶子节点的个数是 ( )

2分

- ☐ A. 63  
☒ B. 64  
☐ C. 65  
☐ D. 不确定

| 参考答案

答案      B

2-5 二叉树的中序遍历也可以循环地完成。给定循环中堆栈的操作序列如下 (其中 **push** 为入栈, **pop** 为出栈) :

2分

```
push(1), push(2), push(3), pop(), push(4), pop(), pop(), push(5), pop(),  
pop(), push(6), pop()
```

以下哪句是对的?

- ☒ A. 3和5是兄弟结点  
☐ B. 1是5的父结点  
☐ C. 6是根结点

☐ D. 以上全不对

| 参考答案

答案      A

2-6 具有1102个结点的完全二叉树一定有\_\_个叶子结点。

2分

☐ A. 79

☒ B. 551

☐ C. 1063

☐ D. 不确定

| 参考答案

答案      B

2-7 任何一棵二叉树的叶结点在先序、中序和后序遍历序列中的相对次序

2分

☐ A. 发生改变

☒ B. 不发生改变

☐ C. 不能确定

☐ D. 以上都不对

| 参考答案

答案      B

2-8 若一棵二叉树的前序遍历序列是{ 4, 2, 1, 3, 6, 5, 7 }, 中序遍历序列是{ 1, 2, 3, 4, 5, 6, 7 }, 则下列哪句是错的?

2分

☐ A. 这是一棵完全二叉树

☒ B. 4是3的父结点

☐ C. 所有的奇数都在叶子结点上

☐ D. 这是一棵二叉搜索树

| 参考答案

答案      B

2-9 已知字符集{ a, b, c, d, e, f, g, h }。若各字符的哈夫曼编码依次是 0100, 10, 0000, 0101, 001, 011, 11, 0001, 则编码序列 0100011001001011110101 的译码结果是:

2分

☐ A. acgabfh

☐ B. adbaggbb

☐ C. afbeagd

☒ D. afeefgd

| 参考答案

答案

D

2-10 具有65个结点的完全二叉树其深度为（根的深度为1）：

2分

- ☐ A. 8
- ☒ B. 7
- ☐ C. 6
- ☐ D. 5

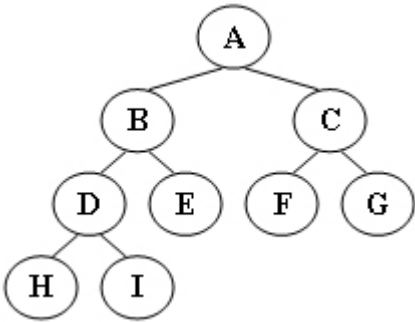
| 参考答案

答案

B

2-11 先序遍历图示二叉树的结果为

2分



- ☐ A. A, B, C, D, H, E, I, F, G
- ☒ B. A, B, D, H, I, E, C, F, G
- ☐ C. H, D, I, B, E, A, F, C, G
- ☐ D. H, I, D, B, E, F, G, A, C

| 参考答案

答案

B

2-12 由分别带权为9、2、5、7的四个叶子结点构成一棵哈夫曼树，该树的带权路径长度为：

2分

- ☐ A. 23
- ☐ B. 37
- ☒ C. 44
- ☐ D. 46

| 参考答案

答案

C

2-13 已知一棵二叉树的先序遍历结果是ABC，则以下哪个序列是不可能的中序遍历结果：

2分

- ☐ A. ABC
- ☐ B. BAC
- ☐ C. CBA
- ☒ D. CAB

| 参考答案

答案	D
----	---

2-14 树最适合于用来表示

2分

- ☐ A. 有序数据元素
- ☐ B. 无序数据元素
- ☐ C. 元素之间无联系的数据
- ☒ D. 元素之间具有分支层次关系的数据

| 参考答案

答案	D
----	---

2-15 设树T的度为4，其中度为1、2、3、4的结点个数分别为4、2、1、1。则T中有多少个叶子结点？

2分

- ☐ A. 4
- ☐ B. 6
- ☒ C. 8
- ☐ D. 10

| 参考答案

答案	C
----	---

2-16 有一个四叉树，度2的结点数为4，度3的结点数为2，度4的结点数为1。问该树的叶结点个数是多少？

2分

- ☐ A. 8
- ☒ B. 12
- ☐ C. 18
- ☐ D. 20

| 参考答案

答案	B
----	---

2-17 对一棵满二叉树，m个树叶，n个结点，深度为h，则（ ）。

2分



- ☐ A.  $n=h+m$
- ☐ B.  $h+m=2n$
- ☐ C.  $m=h-1$
- ☒ D.  $n=2^h-1$

| 参考答案

答案      D

2-18 有关树和二叉树的叙述错误的是（ ）。

2分

- ☐ A. 树中的最大度数没有限制，而二叉树结点的最大度数为2；
- ☐ B. 树的结点无左右之分，而二叉树的结点有左右之分；
- ☒ C. 树的每个结点的孩子数为0到多个，而二叉树每个结点均有两个孩子；
- ☐ D. 树和二叉树均为树形结构

| 参考答案

答案      C

2-19 有关二叉树下列说法正确的是（ ）。

2分

- ☐ A. 二叉树的度为2
- ☒ B. 一棵二叉树的度可以小于2
- ☐ C. 二叉树中至少有一个结点的度为2
- ☐ D. 二叉树中任何一个结点的度都为2

| 参考答案

答案      B

2-20 在一棵度为4的树T中，若有20个度为4的结点，10个度为3的结点，1个度为2的结点，10个度为1的结点，则树T的叶结点个数是：

2分

- ☐ A. 41
- ☒ B. 82
- ☐ C. 113
- ☐ D. 122

| 参考答案

答案      B

2-21 一个具有1025个结点的二叉树的高h为（ ）个。

2分

- ☐ A. 11
- ☐ B. 10

- ☒ C. 11至1025之间
- ☐ D. 10至1024之间

| 参考答案

答案	C
----	---

2-22 已知二叉树的先序遍历序列为ABCDEFGH，中序遍历序列为CBEDFAGH，则该二叉树形态中，父节点的右子节点为（ ）。

2分

- ☐ A. D
- ☐ B. H
- ☒ C. G
- ☐ D. F

| 参考答案

答案	C
----	---

2-23 对  $n$  个互不相同的符号进行哈夫曼编码。若生成的哈夫曼树共有 115 个结点，则  $n$  的值是：

2分

- ☐ A. 56
- ☐ B. 57
- ☒ C. 58
- ☐ D. 60

| 参考答案

答案	C
----	---

2-24 设  $T$  是非空二叉树，若  $T$  的先序遍历和后序遍历序列相同，则  $T$  的形态是 \_\_

2分

- ☒ A. 只有一个根结点
- ☐ B. 没有度为 1 的结点
- ☐ C. 所有结点只有左孩子
- ☐ D. 所有结点只有右孩子

| 参考答案

答案	A
----	---

2-25 设  $T$  是非空二叉树，若  $T$  的后序遍历和中序遍历序列相同，则  $T$  的形态是 \_\_

2分

- ☐ A. 只有一个根结点
- ☐ B. 没有度为 1 的结点
- ☒ C. 所有结点只有左孩子

☐ D. 所有结点只有右孩子

| 参考答案

答案	C
----	---

2-26 已知二叉树的前序遍历序列为 ABDCEFG，中序遍历序列为 DBCAFEG，则后序遍历序列为 \_\_

2分

☐ A. BDACEFG

☒ B. DCBFGEA

☐ C. ABCDEFG

☐ D. GFEDCBA

| 参考答案

答案	B
----	---

2-27 在一棵度为 3 的树中，度为 2 的结点个数是 1，度为 0 的结点个数是 6，则度为 3 的结点个数是 \_\_

2分

☒ A. 2

☐ B. 3

☐ C. 4

☐ D. 无法确定

| 参考答案

答案	A
----	---

2-28 对于任意一棵高度为 5 且有 10 个结点的二叉树，若采用顺序存储结构保存，每个结点占 1 个存储单元（仅存放结点的数据信息），则存放该二叉树需要的存储单元的数量至少是：

2分

☒ A. 31

☐ B. 16

☐ C. 15

☐ D. 10

| 参考答案

答案	A
----	---

2-29 由3 个结点可以构造出多少种不同的二叉树( )

2分

☐ A. 2

☐ B. 3

☐ C. 4

☒ D. 5

| 参考答案

答案 D

2-30 下面几个符号串编码集合中，不是前缀编码的是（ ）。

2分

☐ A. {0,10,110,1111}

☒ B. {11,10,001,101,0001}

☐ C. {00,010,0110,1000}

☐ D. {b,c,aa,ac,aba,abb,abc}

| 参考答案

答案 B

2-31 以下说法错误的是（ ）。

2分

☐ 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個。

☐

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。

| 参考答案

答案 C

2-32 在下述结论中，正确的是（ ）。

2分

①只有一个结点的二叉树的度为0; ②二叉树的度为2; ③二叉树的左右子树可任意交换; ④深度为K的完全二叉树的结点个数小于或等于深度相同的满二叉树。

☐ A. ①②③

☐ B. ②③④

☐ C. ②④

☒ D. ①④

| 参考答案

答案 D

2-33 设森林T中有4棵树，第一、二、三、四棵树的结点个数分别是 $n_1, n_2, n_3, n_4$ ,那么当把森林T转换成一棵二叉树后，且根结点的右子树上有( )个结点。

2分

- ☐ A.  $n_1-1$
- ☐ B.  $n_1$
- ☐ C.  $n_1+n_2+n_3$
- ☒ D.  $n_2+n_3+n_4$

| 参考答案

答案

D

2-34 把一棵树转换为二叉树后，这棵二叉树的形态是（ ）。

2分

- ☐ A. 唯一的，且根结点一定没有左孩子
- ☒ B. 唯一的，且根结点一定没有右孩子
- ☐ C. 有多种，但根结点都没有左孩子
- ☐ D. 有多种，但根结点都没有右孩子

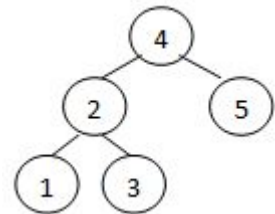
| 参考答案

答案

B

2-35 下列给定的关键字输入序列中，不能生成如下二叉排序树的是：

2分



- ☐ A. 4, 5, 2, 1, 3
- ☒ B. 4, 5, 1, 2, 3
- ☐ C. 4, 2, 5, 3, 1
- ☐ D. 4, 2, 1, 3, 5

| 参考答案

答案

B

函数题

50 分

6-1 本题要求输出二叉树的后序遍历，输出格式见样例。

10分

函数接口定义：

```
#后序遍历
def postOrder(T)
```

其中二叉树类的定义如下：

```
#二叉树的存储-二叉链表
class BinaryTree:
    #1.构造方法
    def __init__(self,newValue):
        self.key = newValue      #树根
        self.left = None        #左子树初始化为空
        self.right = None       #右子树初始化为空
    #2.访问左子树
    def getLeft(self):
        return self.left
    #3.访问右子树
    def getRight(self):
        return self.right
    #4.修改树根的值
    def setRoot(self,newValue):
        self.key = newValue
    #5.访问树根的值
    def getRoot(self):
        return self.key
```

裁判测试程序样例：

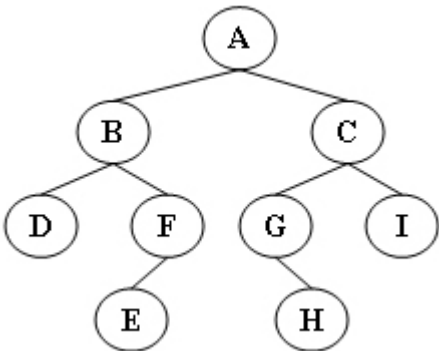
```
T = createBT()    #创建二叉树，实现细节不表
print("PostOrder:",end = "")
postOrder(T)      #输出后序遍历

#你的代码将被嵌在这里
```

输入样例：

```
ABDFECGHI
DBEFAGHCI
```

输出样例（对于图中给出的树）：



```
PostOrder:D E F B H G I C A
```

| 参考答案

答案            编译器: PYTHON3

6-2 本题要求输出二叉树的高度（树根在第1层）。

10分

### 函数接口定义：

```
#求二叉树T的高度
def getHeight(T)
```

其中二叉树类的定义如下：

```
#二叉树的存储-二叉链表
class BinaryTree:
    #1.构造方法
    def __init__(self,newValue):
        self.key = newValue      #树根
        self.left = None        #左子树初始化为空
        self.right = None       #右子树初始化为空
    #2.访问左子树
    def getLeft(self):
        return self.left
    #3.访问右子树
    def getRight(self):
        return self.right
    #4.修改树根的值
    def setRoot(self,newValue):
        self.key = newValue
    #5.访问树根的值
    def getRoot(self):
        return self.key
```

### 裁判测试程序样例：

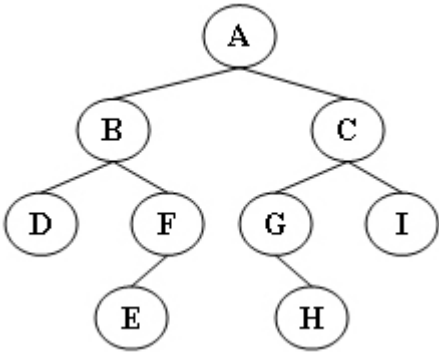
```
T = createBT()    #创建二叉树，实现细节不表
print(getHeight(T)) #输出二叉树的高度
```

### 输入样例：

```
ABDFECGHI
DBEFAGHCI
```

### 输出样例（对于图中给出的树）：





4

| 参考答案

答案 编译器: PYTHON3

6-3 本题要求输出二叉树的层次遍历，输出格式见样例。

10分

函数接口定义：

```
#层次遍历
def layerOrder(T)
```

其中二叉树类和队列类的定义如下：

```
#二叉树的存储-二叉链表
class BinaryTree:
    #1.构造方法
    def __init__(self,newValue):
        self.key = newValue      #树根
        self.left = None        #左子树初始化为空
        self.right = None       #右子树初始化为空
    #2.访问左子树
    def getLeft(self):
        return self.left
    #3.访问右子树
    def getRight(self):
        return self.right
    #4.修改树根的值
    def setRoot(self,newValue):
        self.key = newValue
    #5.访问树根的值
    def getRoot(self):
        return self.key

#定义抽象类型队列Queue, FIFO (First In,First Out)
class Queue:
    #1.构造方法,定义一个空的列表
    def __init__(self):
        self.items = []
    #2.入队,队尾(列表尾部) 入队
    def push(self,item):
        self.items.append(item)
```



```
#3.出队，队首（列表头部）出队
def pop(self):
    return self.items.pop(0)
#4.判断队列是否为空
def isEmpty(self):
    return self.items == []
#5.取队首
def getFront(self):
    return self.items[0]
#6.求队列大小
def getSize(self):
    return len(self.items)
```

裁判测试程序样例：

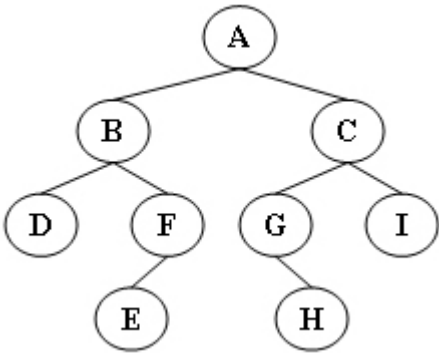
```
T = createBT()    #创建二叉树，实现细节不表
layerOrder(T)     #输出层次遍历

#你的代码将被嵌在这里
```

输入样例：

```
ABDFECGHI
DBEFAGHCI
```

输出样例（对于图中给出的树）：



```
A B C D F G I E H
```

| 参考答案

答案

编译器: PYTHON3

6-4 本题要求统计二叉树结点个数。

10分

函数接口定义：

```
#求二叉树T的高度
def nodeCount(T)
```

其中二叉树类的定义如下：

```
#二叉树的存储-二叉链表
class BinaryTree:
    #1.构造方法
    def __init__(self,newValue):
        self.key = newValue      #树根
        self.left = None        #左子树初始化为空
        self.right = None       #右子树初始化为空
    #2.访问左子树
    def getLeft(self):
        return self.left
    #3.访问右子树
    def getRight(self):
        return self.right
    #4.修改树根的值
    def setRoot(self,newValue):
        self.key = newValue
    #5.访问树根的值
    def getRoot(self):
        return self.key
```

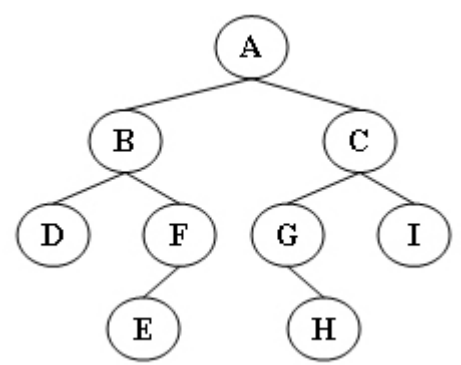
裁判测试程序样例：

```
T = createBT()    #创建二叉树，实现细节不表
print(nodeCount(T)) #输出二叉树的结点数
```

输入样例：

```
ABDFECGHI
DBEFAGHCI
```

输出样例（对于图中给出的树）：



9

| 参考答案

答案 编译器: PYTHON3

6-5 本题要求计算二叉树中有多少片树叶，输出格式见样例。

10分

函数接口定义：

```
#树叶统计
def leafCount(T)
```

其中二叉树类的定义如下：

```
#二叉树的存储-二叉链表
class BinaryTree:
    #1.构造方法
    def __init__(self,newValue):
        self.key = newValue      #树根
        self.left = None        #左子树初始化为空
        self.right = None       #右子树初始化为空
    #2.访问左子树
    def getLeft(self):
        return self.left
    #3.访问右子树
    def getRight(self):
        return self.right
    #4.修改树根的值
    def setRoot(self,newValue):
        self.key = newValue
    #5.访问树根的值
    def getRoot(self):
        return self.key
```

裁判测试程序样例：

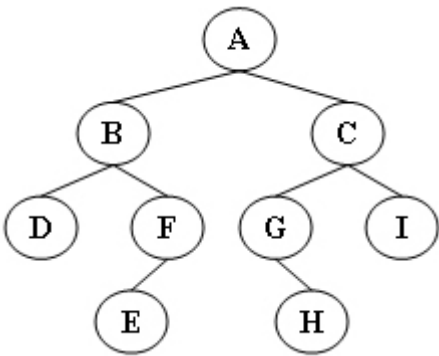
```
T = createBT()    #创建二叉树，实现细节不表
print(leafCount(T))

#你的代码将被嵌在这里
```

输入样例：

```
ABDFECGHI
DBEFAGHCI
```

输出样例（对于图中给出的树）：



4

| 参考答案

答案

编译器: PYTHON3

编程题

30 分

7-1 给定一棵二叉树的前序遍历序列和中序遍历序列，要求计算该二叉树的高度。

10分

输入格式:

输入首先给出正整数  $n$  ( $\leq 50$ )，为树中结点总数。随后 2 行先后给出前序和中序遍历序列，均是长度为  $n$  的不包含重复英文字母（区别大小写）的字符串。

输出格式:

输出为一个整数，即该二叉树的高度。

输入样例:

9  
ABDFGHIEC  
FDHGIBEAC

输出样例:

5

| 参考答案

答案

编译器: NO\_COMPILER

7-2 本题要求根据给定的一棵二叉树的后序遍历和中序遍历结果，输出该树的前序遍历结果。

10分

输入格式:

第一行给出正整数  $n$  ( $\leq 30$ )，是树中结点的个数。随后两行，每行给出  $n$  个整数，分别对应后序遍历和中序遍历结果，数字间以空格分隔。题目保证输入正确对应一棵二叉树。

输出格式:

在一行中输出 Preorder: 以及该树的前序遍历结果。数字间有1个空格，行末不得有多余空格。



输入样例:

```
7
2 3 1 5 7 6 4
1 2 3 4 5 6 7
```

输出样例:

```
Preorder: 4 1 3 2 6 5 7
```

| 参考答案

答案

编译器: NO\_COMPILER

7-3 对于给定的二叉树，本题要求你按从上到下、从左到右的顺序输出其所有叶结点。

10分

输入格式:

首先第一行给出一个正整数  $n$  ( $\leq 10$ )，为树中结点总数。树中的结点从 0 到  $n - 1$  编号。随后  $n$  行，每行给出一个对应结点左右孩子的编号。如果某个孩子不存在，则在对应位置给出 "-"。编号间以 1 个空格分隔。

输出格式:

在一行中按规定顺序输出叶结点的编号。编号间以 1 个空格分隔，行首尾不得有多余空格。

输入样例:

```
8
1 -
- -
0 -
2 7
- -
- -
5 -
4 6
```

输出样例:

```
4 1 5
```

| 参考答案

答案

编译器: NO\_COMPILER

