

## K 中序拜訪

說明：

二元樹的定義：

1. 樹不可以為空集合，亦即至少必須有一個根節點，但二元樹卻可以是空集合。
2. 樹的兄弟節點位置次序並非固定，但二元樹是固定的。也就是下面是相同的樹，但卻不是相同的二元樹。

在二元樹的運用上，常常需要找出所有的節點資料，這個過程稱為樹的拜訪或追蹤。依拜訪追蹤的次序可分成下列三種：前序preorder、中序inorder及後序postorder。

- 前序preorder定義：先訪根節點，若有左子樹，先拜訪其左子樹的所有節點；若有右子樹，再拜訪其右子樹的所有節點。
- 中序inorder定義：拜訪根節點前，若有左子樹，先拜訪其左子樹的所有節點；拜訪根節點；若有右子樹，再拜訪其右子樹的所有節點。
- 後序postorder定義：拜訪根節點前，若有左子樹，先拜訪其左子樹的所有節點；若有右子樹，再拜訪其右子樹的所有節點，最後再拜訪根節點。

二元搜尋樹(Binary Search Tree)定義：

二元搜尋樹是一種二元樹，它可以為空集合，若不為空集合，則必須要滿足以下條件：

1. 若左子樹不為空集合，則左子樹的鍵值均須要小於樹根的鍵值。
2. 若右子樹不為空集合，則右子樹的鍵值均須要大於樹根的鍵值。
3. 左子樹與右子樹必須也要保持二元搜尋樹。

由使用者輸入 $x$ 筆資料，建立一個二元搜尋樹(Binary Search Tree)，輸出二元搜尋樹的中序拜訪的結果。

## 輸入

第一列的數字 $n$ 代表有幾組資料要測試， $1 \leq n \leq 5$ ，第二列起為每組的測試資料，之後每二列為每組的測試資料。每組測試資料的第一列是一個整數 $3 \leq x \leq 20$ ，用來表示這組測試資料有幾個節點；每組測試資料的第二列為這組測試資料各節點編號，以","分隔各節點編號，編號為一整數 $0 \leq N \leq 99$ ，各節點編號不會相同。用測試資料以二元搜尋樹方式建樹。

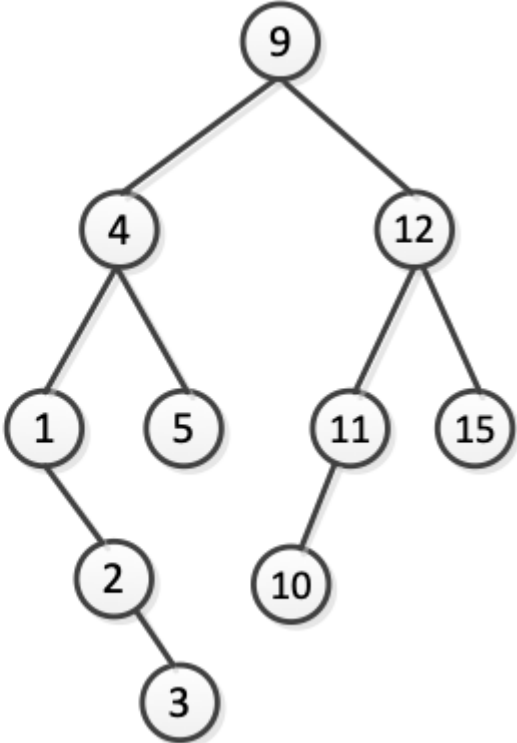
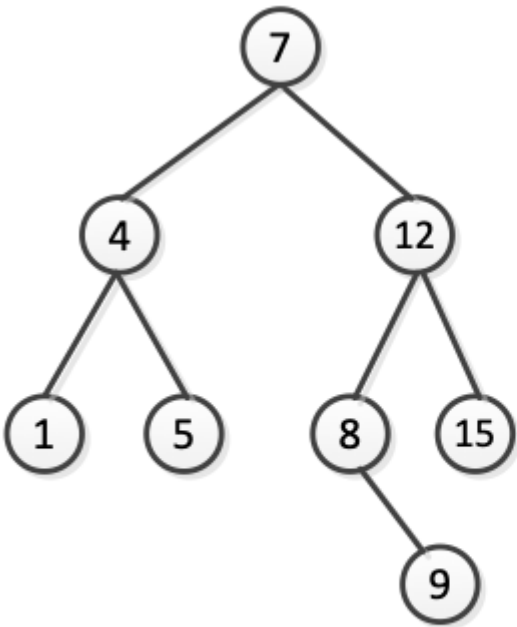
## 輸出

在測試資料中所建二元搜尋樹，輸出二元搜尋樹的中序拜訪的結果，以" "分隔各節點編號。

範例輸入輸出

範例輸入 I

1 7,4,1,5,12,8,9,15  
2 9,4,1,5,12,11,10,15,2,3

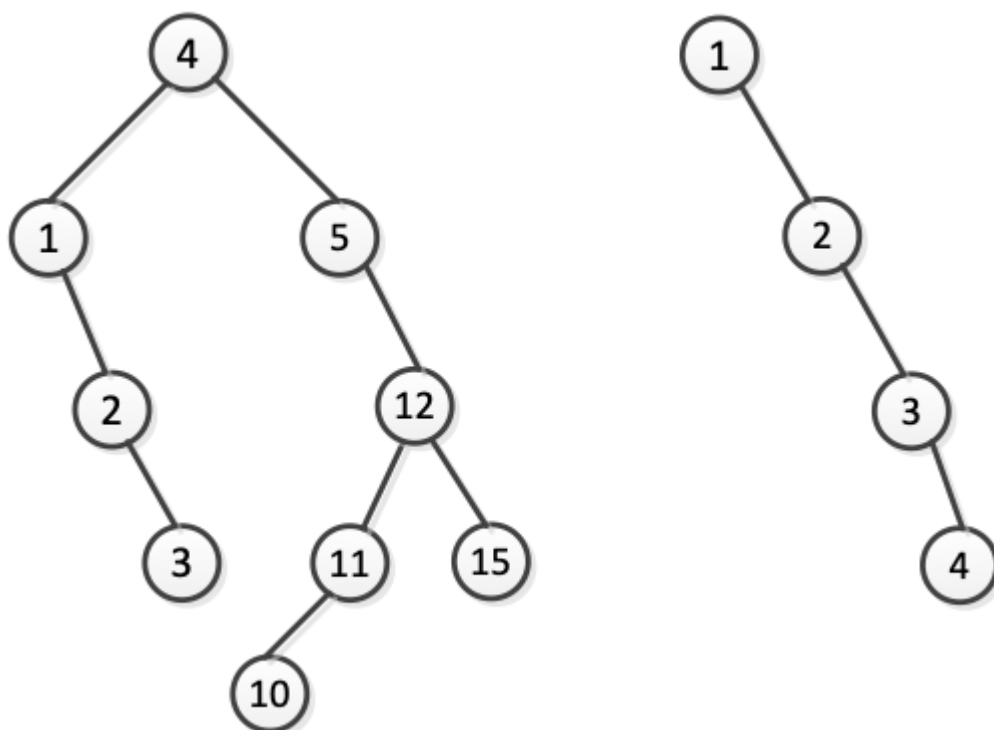


範例輸出 I

1 1 4 5 7 8 9 12 15  
2 1 2 3 4 5 9 10 11 12 15

## 範例輸入 II

```
1 | 4,1,5,12,11,10,15,2,3
2 | 1,2,3,4
```



## 範例輸出 II

```
1 | 1 2 3 4 5 10 11 12 15
2 | 1 2 3 4
```