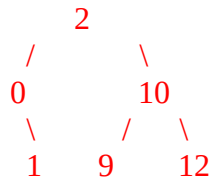
1) (10 pts) DSN (Binary Search Trees)

(a) (3 pts) Given the following traversals, draw the Binary Search Tree they represent.
Pre-Order: 2, 0, 1, 10, 9, 12
Post-Order: 1, 0, 9, 12, 10, 2
In-Order: 0, 1, 2, 9, 10, 12



// 1 pt placing 2 at the root, 1 pt left subtree, 1 pt right subtree

(b) (5 pts) If the nodes of the BST have the following structure, construct a recursive function to count the number of nodes in the tree.

```
typedef struct bstNode {
   struct bstNode *left, *right;
   char word[20];
} bstNode;

int count(bstNode *root){

    if (root == NULL) return 0;              // 2 pts

    //1 pt 1, 1 pt left, 1 pt right
    return 1 + count(root->left) + count(root->right);
}
```

(c) (2 pts) Write a single line of code calling the count function that assigns the number of nodes in the left subtree of the tree pointed to by a pointer `myTreePtr` to the integer variable `leftCount`. You may assume that `myTreePtr` is not pointing to NULL and points to an actual `bstNode`.

```
int leftCount = count(myTreePtr->left);          // 2 pts
```