4) (10 pts) DSN (Binary Trees)

Write a ***recursive*** function that takes the root of a binary tree (possibly NULL) and returns the sum of all the nodes that are left children in the tree. (See the example below, which returns 15 + 49 = 64, since the only nodes that are left children anywhere in the tree are 15 and 49.)

For this tree, the function should return 15 + 49 = 64:

```
   28
  /  \
15    22
 \   /  \
  1 49  68
```

The node struct and function signature are:

```
typedef struct node {
    int data;
    struct node *left;
    struct node *right;
} node;

int add_all_left_children(node *root) {

    if (root == NULL) return 0;

    int sum = 0;
    if (root->left != NULL)
        sum += root->left->data;

    return sum + add_all_left_children(root->left) +
                 add_all_left_children(root->right);

}
```

Grading: 2 pts base case, 1 pt NULL check left, 2 pts adding left node if it exists, 5 pts for the final return (1 pt for previous sum, 2 pts for each recursive call)