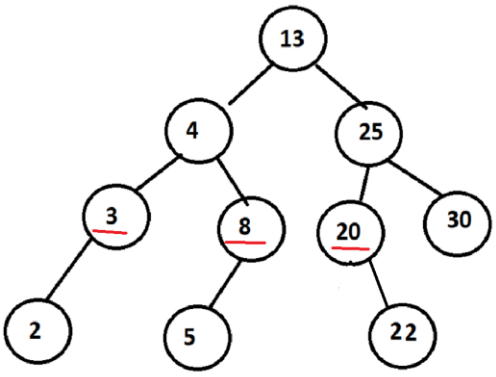
1) (10 pts) DSN (Binary Trees)

Write a function named *sumSingleParents*() that takes a pointer to the root of a binary tree (*root*) and returns the sum of all the values in the nodes with a single child.

For example, if you pass the root of the following binary tree, the function should return 31 (=3+8+20) as the nodes containing 3, 8, and 20 have only one child:



You must write your solution in a **single** function. You cannot write any helper functions.

The function signature and node struct are given below.

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

int sumSingleParents(node *root) {

    if(root == NULL)                                  // 1 pt
        return 0;

    int sum = 0;                                      // 1 pt

    if((root->left == NULL && root->right != NULL) ||  // 2 pts
       (root->left != NULL && root->right == NULL))
        sum+= root->data;                             // 1 pt

    sum +=  sumSingleParents(root->left);             // 2 pts
    sum +=  sumSingleParents(root->right);            // 2 pts
    return sum;                                       // 1 pt
}
```

Note: Many ways to do this. Map points accordingly. Watch out for NULL ptr errors.