4) (10 pts) DSN (Binary Trees)

We define the ***offcenter value*** for each node in a binary tree as being the absolute value of the difference between the height of its left subtree and the height of its right subtree. For example, for an AVL tree, each node has an offcenter value of 0 or 1. Also, note that we define the offcenter value of a null node to be 0. Write a function, `maxOffCenterValue` that computes the maximum offcenter value of any node in a tree pointed to by root. To make your task easier, assume that the height of each node is stored in the corresponding struct for that node in the component `height`.

Using the struct definition given below, complete the function in the space provided.

```
#include <math.h>

typedef struct treenode {
    int value;
    int height;
    struct treenode *left;
    struct treenode *right;
} treenode;

int max(int a, int b) {
    if (a > b) return a;
    return b;
}

int maxOffCenterValue(treenode* root) {

    if (root == NULL) return 0;
    if (root->left == NULL && root->right == NULL) return 0;
    if (root->left == NULL || root->right == NULL)
        return root->height;

    int lVal = maxOffCenterValue(root->left);
    int rVal = maxOffCenterValue(root->right);
    int res = max(lVal, rVal);

    int cur = abs(root->left->height - root->right->height);
    return max(res, cur);
}
```

Grading: 1 pt NULL case, 1 pt 1 node case, 2 pts root has one child, 1 pt for left rec call, 1 pt for right rec call, 2 pts for current node calculation, 2 pts for getting max of all three.