Check Children Sum

```
/*Check for Children Sum Property in a Binary Tree*/
#include <stdio.h>
#include <stdlib.h>

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

struct node *newNode(int data)
{
        struct node *temp = (struct node *)malloc(sizeof(struct node));
        temp->data = data;
        temp->left = temp->right = NULL;
        return temp;
}

int checkChildrenSum(struct node *root)
{
        int lsum = 0, rsum = 0;
        if(!root || (!(root->left) && !(root->right)))
                return 1;
        if(root->left)
                lsum = root->left->data;
        if(root->right)
                rsum = root->right->data;
        return ((root->data == (lsum + rsum))
                && checkChildrenSum(root->left)
                && checkChildrenSum(root->right))? 1: 0;
}


int main()
{
        struct node *root = newNode(10);
        root->left = newNode(8);
        root->right = newNode(2);
        root->left->left = newNode(3);
        root->left->right = newNode(5);
        root->right->right = newNode(2);
        checkChildrenSum(root)? printf("Given tree satisfies children sum property"):
        printf("Given tree not satisfies children sum property");
        return 0;
}
```

Time complexity: O(n)
Space complexity: O(n)