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

A modified BST node that stores the `sum` of the `data` values in its sub-tree. **Complete** writing the insert function shown below ***recursively***, so that it takes in a pointer to the root of a binary search tree, *root*, and an integer, *value*, inserts a node storing value in it into the tree and returns a pointer to the root of the resulting tree. Notice that this task is more difficult than a usual binary tree insert since the sum values in several nodes must be updated as well. The struct used to store a node is shown below.

```
typedef struct bstNode {
   struct bstNode * left, * right;
   int data;
   int sum;
} bstNode;

bstNode* insert(bstNode * root, int value){


    if (root == NULL) {
        bstNode* res = malloc(sizeof(bstNode));

        res->data = value;   // 1 pt

        res->sum = value;    // 1 pt

        res->left = NULL;    // 1 pt

        res->right = NULL;   // 1 pt
        return res;
    }

    if (value <= root->data)

        root->left = insert(root->left, value) ;     // 2 pts
    else
        root->right = insert(root->right, value)  ; // 2 pts

    root->sum += value ;                              // 2 pts

    return root;
}
```