

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

Complete writing function shown below **recursively**, so that it takes in a pointer to the root of a binary search tree, *root*, and an integer, *value*, and returns the number of nodes in the tree that are divisible by *value*. The struct used to store a node is shown below.

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

int countDiv(bstNode *root, int value){

    if (root == NULL) return 0;                //2 pts

    // 4 pts, 2 pts for each recursive call.
    int res = countDiv(root->left, value) +
               countDiv(root->right, value);

    // 2 pts for checking divisibility, 1 pt for adding 1
    if (root->data % value == 0)
        res++;

    // 1 pt for returning.
    return res;
}
```