Computer Science I – Fall 2010
Recitation #7: Binary Search Trees

1. Draw the binary search tree that results from inserting the following values into an initially empty binary search tree in the following order: 50, 27, 16, 88, 34, 65, 52, 77, 93, 4, 12, 29, 44, 92

2. What are the outputs of a pre-order and post-order traversal of the final binary search tree drawn in question 1?

Pre-order: ___ , ___ , ___ , ___ , ___ , ___ , ___ , ___ , ___ , ___ , ___ , ___ , ___ , ___

Post-order: ___ , ___ , ___ , ___ , ___ , ___ , ___ , ___ , ___ , ___ , ___ , ___ , ___ , ___

3. If a search was conducted for the value 37 in the final binary search tree from question #1, which nodes would get visited? (List them in the order they get visited.)

4. Write a function which returns the smallest value stored in a *non-empty* binary search tree. The prototype is below:

```
int minVal(struct treenode* root) {




}
```

5. Write a function which returns the number of leaf nodes in a binary search tree. The prototype is below:

```
int numLeafNodes(struct treenode* root) {









}
```

6. What does the following function do?

```
struct treenode* q6(struct treenode* root, int x) {

  if (root == NULL)
    return NULL;

  if (root->data > x) {

    struct treenode* tmp = q6(root->left, x);
    if (tmp == NULL)
      return root;
    else
      return tmp;
  }
  else
    return q6(root->right, x);
}
```