1) (10 pts) DSN (Binary Trees)

Michael took CS 1 last semester. During the Winter break he thought that it would be cool to keep track of all of the new words that he learned while reading a novel. He has stored all of his words (all 1-19 lowercase letters only) in alphabetic order in a binary search tree (BST). The nodes of his BST are stored in the following structure:

```
typedef struct bstNode {
   struct bstNode *left, *right;
   char word[20];
} bstNode;
```

Michael wants to count the number of words in his binary search tree that come before a specified word in alphabetical order. Write a **recursive** function `countBefore` which takes in a pointer to the root of a binary search tree storing the words and a string `target` (of 1-19 lowercase letters only) and returns the number of words in the tree that ***come before*** `target`, alphabetically.

```
int countBefore(bstNode* root, char target[]){

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

    if (strcmp(target, root->word) <= 0)     // 2 pts
        return countBefore(root->left);      // 2 pts

    // 1 pt return, 1 pt 1, 1 pt left, 1 pt right
    return 1 + countBefore(root->left) + countBefore(root->right);
}
```