3) (10 pts) DSN (Tries)

Write a function that takes the root of a trie (*root*) and returns the number of strings in that trie that **end** with the letter *q*. The *count* member of the node struct indicates how many occurrences of a particular string have been inserted into the trie. So, a string can be represented in the trie multiple times. If a string ending in *q* occurs multiple times in the trie, all occurrences of that string should be included in the value returned by this function.

The node struct and function signature are given below. You must write your solution in a **single** function. You cannot write any helper functions.

```
typedef struct TrieNode
{
   // Pointers to the child nodes (26 total).
   struct TrieNode *children[26];

   // Indicates how many occurrences of a particular string are contained
   // in this trie. If none, this is set to zero (0).
   int count;
} TrieNode;

int ends_with_q_count(TrieNode *root)
{

      int i;
      int retval = 0;

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

      // Grading: 1 pt loop to 26, 1 pt retval +=, 1 pt rec call
      for (i = 0; i < 26; i++)
            retval += ends_with_q_count(root->children[i]);

      // Grading: 2 pts NULL check, 2 pts retval +=
      // Note: 'q' - 'a' = 16. This can go before the for also.
      if (root->children['q' - 'a'] != NULL)
            retval += root->children['q' - 'a']->count;

      // Grading: 1 pt
      return retval;
}
```