

3) (10 pts) DSN (Tries)

Write an **iterative, non-recursive** function that takes the root of a trie (*root*) and a string (*str*) and returns the number of new nodes we would have to add to our trie in order to insert that string. You may assume that *str* is non-NULL, non-empty, and contains lowercase alphabetic characters only (i.e., it won't contain uppercase letters or non-alphabetic characters). However, you must handle the case where *root* is NULL.

Special Restrictions:

- Please do not use pointer arithmetic (e.g., `str + 1`).
- Do not modify or corrupt the trie or the string. (Do not add nodes to the trie!)
- Do not call `strlen()` repeatedly, as it is an $O(k)$ function (where k is the length of the string). If you need to call `strlen()`, find a way to do it only once for the given string.

The trie node struct and function signature are as follows. Do NOT write any helper functions.

```
#include <string.h>
typedef struct TrieNode {
    struct TrieNode *children[26];
    int flag; // 1 if the string is in our trie, 0 otherwise
} TrieNode;

int newNodeCount(TrieNode* root, char* str) {
```

```
}
```

Computer Science Foundation Exam

January 15, 2022

Section C

ALGORITHMS ANALYSIS

**NO books, notes, or calculators may be used,
and you must work entirely on your own.**

Name: _____

UCFID: _____

NID: _____

Question #	Max Pts	Category	Score
1	5	ANL	
2	10	ANL	
3	10	ANL	
TOTAL	25		

You must do all 3 problems in this section of the exam.

Problems will be graded based on the completeness of the solution steps and not graded based on the answer alone. Credit cannot be given unless all work is shown and is readable. Be complete, yet concise, and above all be neat. For each coding question, assume that all of the necessary includes (stdlib.h, stdio.h, math.h, string.h) for that particular question have been made.