3) (10 pts) DSN (Tries)

We are maintaining a Trie for predicting the next letter for a given string. The trie node struct and its properties are discussed below.

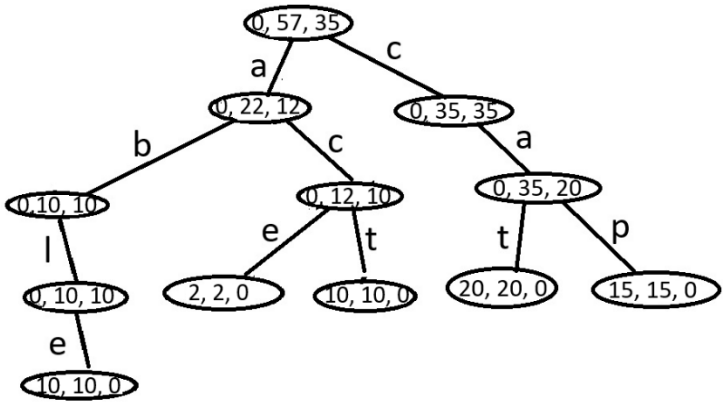
```
typedef struct trienode {
    int freq;
    int sum_prefix_freq;
    int cur_max_freq;
    struct trienode* next[26];
} trienode;
```

- **`freq`:** The frequency of the word represented by this node (i.e., how many times this specific word has been added to the dictionary). If this value is 0, it means that there is no word in the trie that ends at that node.
- **`sum_prefix_freq`**: The total frequency of all words in the dictionary that have this string as a prefix, including the string itself.
- **`cur_max_freq`**: The highest sum frequency among all child nodes of the current node.
- **`next[26]`:** These are typical child pointers of a trie node. This is an array of 26 pointers, each representing one of the possible next letters ('a' to 'z'). A pointer should be NULL if no words in the dictionary continue along that path. Typically, only a subset of these pointers will be active.

As an example, the following trie is constructed after inserting the following list of words and their frequency. The numbers inside a node represent its `freq, sum_prefix_freq, cur_max_freq,` respectively.

List of words and their frequency added to the trie.

`cap 15, cat 20, act 10, able 10, ace 2,`



Your goal is to **complete** the recursive function on the next page that receives the root of a trie, **t**, a string, **str**, and an integer, **k**, the current position in the string, and returns the most likely letter that follows the input string. **You may assume a unique next letter appears the most number of times** (cur_max_freq). If there is no string in the trie that **str** is a proper prefix for, then return the question mark character, '?'.

For example, if the string passed to the function is:

- predict(root, “a”, 0) should make a recursive call to predict(root->next[0], “a”, 1), which should then return ‘c’ because ‘c’ is the most likely letter to follow “a” for the sample trie.
- predict(root, “ab”, 0) will eventually return ‘l’ (lower case L) after two recursive calls.
- predict(root,“ac”, 0) will eventually return ‘t’ after two recursive calls.
- predict(root,“ace”, 0) will eventually return ‘?’ after three recursive calls because “ace” is not a proper prefix of any word in the trie.
- predict(root, “ap”, 0) will make one recursive call and then from there return ‘?’, since “ap” isn’t a prefix of any word in the trie. (In the code, this case is slightly different from the previous one.)

```
char predict(trienode *t, char *str, int k) {

    if (t == NULL) return '?';

    if (k == strlen(str)) {

        // Checks if there is no string with this prefix.

        if ( ____________________________________ )
            return '?';

        // Looks through all possible next letters until it finds the one
        // has the most words that start with that prefix.
        // Note: part before the && is for short-circuiting to avoid null ptr.
        for (int i=0; i<26; i++) {

            if ( _________________________________ &&

                 ____________________________________________________________ )
                    return (char)('a'+i);
        }
    }

    // We require a recursive call in this case.

    return ____________________________________________________________ ;

}
```

Computer Science Foundation Exam

May 17, 2025

Section C

ALGORITHM ANALYSIS

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

PLEASE USE CAPITAL LETTERS IN WRITING YOUR NAME

Last Name: _____

First Name: _____

UCFID: _____

Question #	Max Pts	Category	Score
1	10	ANL	
2	5	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, stdio, math, string) for that particular question have been made.