CMSC21 Lab Exercise 10 – Self-referential types and pointers

Using the TreeNode defined in class, design the following recursive functions.

1. Create a function that visits every node in a tree, printing the name of every node (each on it's own line).

2. Create a function that accepts the root of a tree and a name (a character), and determines if the name exists in the tree.

A linked list another self-referential type. It is either:

a) empty, or

b) a node with a pointer to a linked list

Define a ListNode (Hint: similar to a TreeNode, except has only one pointer), and design the following recursive functions.

1. Create a function that visits every node in a linked list, printing the name of every node (each on it's own line).

2. Create a function that returns the “max” name in a linked list. The “max” name is the name with the highest ASCII value.

3. Create a function that finds the average of integer values in a linked-list. You’ll need another type of ListNode, containing an integer instead of a character. (Hint: there are three steps in finding the average value.)

4. **OPTIONAL** Create a function that accepts the "head" of a list and a pointer to a ListNode, and inserts the ListNode at the "tail" of the list. The function should return the head of the list.