CS212 – Lab 8 – List Insertion and Removal

In this lab, you will be adding code to, and modifying, your pointer-based 2-way queue program. These modifications will give it more attributes of a general-purpose 2-way list.

A. If your 2-way-list does not have the current-node pointer in global memory, you may move it there if you need to do so to simplify your program.

B. Implement the following new functions for manipulating the list:

   a. InsertAfter(node* node_ptr, *value*) – inserts a new node following the node pointed to by node_ptr. The new node will have *value* inserted as its data. The type for *value* will be consistent with your previous 2-way list lab.

   b. DeleteNode(node* node_ptr) – removes the node pointed to by node_ptr from the list. BE SURE TO FREE THE NODE!!! Be careful if the anchor is the 1st node!! Do not delete that one if it is not dynamic!

   c. FindValue(*value*) – locates the **first occurrence** of *value* in the list and returns a pointer to that node and sets a global integer that tells which node number it is. Do NOT count the anchor as a node unless your design has the anchor containing data like any other node. Use this as input to DeleteNode.

   d. You may implement an "isEmpty" function if it simplifies your program.

   e. You may need to add code to, or modify, your original Push code.

C. Using the above functions, modify your main program as follows:

   a. Read a sequence of at least 10 characters and *remember* the 3rd character (not including *linend* or *space* characters) in the list for later.

   b. Push the characters into the list (1 character per node) – DUPLICATES ARE KEPT. Do NOT allow *linend* or *space* characters to be in the list, because they do not print.

   c. Do the following after the last symbol is reached:

      i. print the list in forward order

      ii. search the list for the *remembered character* and do these actions:

         1. display the position # of that character in the list
         2. insert the symbol, "*", after the remembered character
         3. delete that remembered character from the list
         4. print the list in forward order again, so we can see it was deleted
         5. delete all remaining copies of the remembered character (if any).
         6. print the list in forward order again

D. you may use any of the functions you have already implemented in your 2-way-list lab.

Notes:

1. The rules for data and processing the list from the previous labs still apply.