CS211 – Programming Practicum Fall 2013

Programming Project 12

Due: Thursday, 11/14/13 at 11:59 pm

Linked Lists and Recursion

For this lab you are to write a C program that will perform the following operations on a linked list.

- insert
- delete
- contains
- empty
- list items
- list items in the list in reverse order

The linked list is to contain one integer value per node. The values are to be stored in increasing order. The node at the head of the list is to contain the smallest value. The node at the end of the list is to contain the largest value. The same value can be stored multiple times in the list.

The input for the operations will come from standard input. The commands are to follow the descriptions given below. Note: that the form <int> could be any integer number and it will NOT be enclosed in angle brackets. <int> is just a notation to specify and integer value. The integer value is to be input on the same line as the command character. There will just be a single integer value per command. If the first character on the line is not one of the following characters, print an error message and ignore the rest of the information on that line.

q - **quit** the program

i <int> - **insert** the integer value into the linked list. The items in the list are to be stored in increasing order with the smallest values toward the front of the list.

d <int> - **delete** the integer value from the linked list. Be sure to properly deallocate the nodes in the linked list.

c <int> - display a message stating whether the given integer value is **contained** in the linked list.

e - **empty** all values from the linked list. Be sure to properly deallocate the nodes in the linked list.

l - **list** the items contained in the linked list.

r - list the items contained in the linked list in **reverse** order.

? - list the valid input commands

h - list the valid input commands

Recursion

Each of the above operations can be done using a recursive algorithm. You are required to write these operations recursively.

Command Line Argument: Debug Mode

Your program is to be able to take one optional command line argument, the -d flag. When this flag is given, your program is to run in "debug" mode.

When in debug mode, your program is to have each recursive function print out two messages. These messages will help show how the recursive functions work and how many recursive calls are actually made.

1. Print out a message as the first line of every recursive function that shows the nme of the function.
2. Print out a message right before the return that shows the name of the function.

When the flag is not given, this debugging information should not be displayed. One simple way to set up a "debugging" mode is to use a boolean variable which is set to true when debugging mode is turned on but false otherwise. Then using a simple if statement controls whether information should be output.

```
        if ( debugMode == TRUE )
                printf (" Debugging Information \n");
```

Program Hints

Recursion really occurs when traversing through list.   It is suggested that you first use a non-recursive insert function to create a list. The write the recursive code in the following order:

1. list
2. list in reverse order
3. contains
4. insert
5. empty
6. delete

Note that the first three operations do not modify the list, so the node pointer parameter can be pass-by-value. The last three operations do modify the list, so the node pointer parameter for these needs to be pass-by-reference.

Program Submission

You are to submit the programs for this lab via the Assignments Page in Blackboard.

To help the TA, name your file with your net-id and the assignment name, like:

- ptroy1LabX.c