**CS 531: Fundamentals of Systems Programming**

**Homework # 4 Rubric**

Homework # 4 builds upon Homework # 3. Recall that in HW#3, our program reads the contents of datafile “CS531\_Inet.txt”. Each line in “CS531\_Inet.txt” contains an IPv4 address and alias pair as shown below:

**111.22.3.44 platte**

**131.250.95.21 jet**

**172.66.7.88 wabash**

**111.22.5.66 green**

**131.250.47.63 baker**

As with HW#3, for this assignment, you will create your own test data

files using the sample format. Grading will involve using different

data files with the same format.

In HW#3, you were asked to construct a linked list of **address\_t** structures, each of which contained the four integers of an IPv4 addressalong with a fifth component in which to store an associated alias of upto 10 characters. In this project, ***you will replace the linked list with a Binary Search Tree (BST).***

Again, you will create a structure type called **address\_t** similar to that used HW#3. In addition, **address\_t** will also contain the following two fields:

Struct address\_t \*leftChild, \*rightChild;

As CS531\_Inet.txt is being read in, the data will be stored in a BST composed of **address\_t** structures containing the address/alias pairs read in from the file.

Once the BST has been created, the user will receive the following menu

options:

1) Add address

2) Look up address

3) Update address

4) Delete address

5) Display list

6) Display aliases for location

7) Save to file

8) Quit

**Program structure and design:**

* ***Display list and Display aliases for location shall be based on Inorder Traversal***
* Delete address, Look up address, and Display aliases for location; will display an error message if the address (or location) entered is not listed. Following the error message, the menu will be redisplayed.
* A separate UDF (user defined function) will be defined for each menu option.
* No duplicate aliases or address are allowed. If attempted, display an appropriate error message followed by the menu.
* For this exercise, all aliases will be entered in lower case.

Based on the sample data discussed above, an example run of the program may look like:

1) Add address

2) Look up address

3) Update address

4) Delete address

5) Display list

6) Display aliases for location

7) Save to file

8) Quit

Enter menu option: **5**

**baker 131.250.47.63**

**green 111.22.5.66**

**jet 131.250.95.21**

**platte 111.22.3.44**

**wabash 172.66.7.88**

(**note that option 5 displays the list in alphabetical order, due to the inorder traversal**)

Enter menu option: **6**

Enter Locality: **131.250**

**Location: 131.250**

**baker**

**jet**

Enter menu option: **2**

Enter alias: **platte**

**platte: 111.22.3.44**

Enter menu option: **1**

**Enter IPv4 address: 131.250.42.18**

**Enter alias: barbara**

Enter menu option: **5**

**baker 131.250.47.63**

**barbara 131.250.42.18**

**green 111.22.5.66**

**jet 131.250.95.21**

**platte 111.22.3.44**

**wabash 172.66.7.88**

Enter menu option: **6**

Enter Locality: **131.250**

**Location: 131.250**

**baker**

**barbara**

**jet**

Enter menu option: **8**

**Goodbye!**

**Rubric 10 points**:

* Is the source code well documented and formatted using clearly readable indentation and white space (while viewed within **vi**)? **1 point**
* Is the BST and associated recursion properly implemented? **3 points**
* Does each menu option map to its own UDF, and is each UDF properly implemented? **6 points**
* **Note:** Your program must compile using gcc/unix on Mason in order to receive credit. Submit via Blackboard by the deadline. 1 point deduction per day late.