## **Lab 10 (30 Points)**

**Objectives:**

* Learn how to create and use a circular linked list

**Description:**

In this lab 10, all functions need to be implemented based on a circular linked list.

**Note: you could decide whether to use a dummy node or not but using a dummy node will make implementation way easier.**

Below are functions that you need for this lab:

1. Node \* **createCircularSortedLinkedList** ( int numOfNodes )
   * Create a linked list using **createNode** ( data ) where 1 ≤ **data** ≤ 10, using rand( ).
   * Insert the newly created Node in descending order using **insertOneNewNode** ( ).
   * Return:the head of the linked list.
2. Node \* **createNode** (int num)
   * This function will create a Node struct using malloc( ) and initialize the Node **data** with **num** variable and Node **next** with **NULL**.

**typedef struct l\_node {**

**int** data;

**struct l\_node \*** next;

**} Node;**

* + Return: the newly created Node.

1. Node \* **insertOneNewNode** ( Node \* head, Node \* newNode )
   * This function inserts a new Node in descending order.
   * Return: the head of the updated linked list.
2. Node \* **removeAllKeyNodes** ( Node \* head, int \* searchKey )
   * Remove all the **serachKey** nodes.
   * Return: the head of the updated linked list.
3. Node \* **findNode**(int searchKey, Node \* head)
   * Input:an integer search key, and first node of the linked list.
   * Search the Node that contains the **searchKey**.
   * Return:Node pointer to the first found key node or **NULL** if not found.
4. void **displayLinkedList** (Node \*head)
   * Input:first node of the linked list.
   * Display the keys in the linked list.
5. void **freeLinkedList** (Node \*head)
   * Input: first node of the linked list.
   * Free all the Nodes in the linked list.
   * Print a message, “The linked list has been freed.”
6. **Main** function
   * Create a linked list with 10 Nodes using **createSortedLinkedList( ).**
   * Display your linked list with the **displayLinkedList( ).**
   * Display a message, “Enter an integer search key from 1 to 10 to delete all nodes:”.
   * Delete all the nodes using **removeAllKeyNodes( ).**
   * Display your deleted node linked list with the **displayLinkedList( ).**
   * Free up the linked list using **freeLinkedList( ).**

**Every user-defined function must have a comment describing:**

* What function does;
* What parameter values are;
* What value it returns.

**Example from the terminal window:**

**$ ./a.out**

My linked list’s keys: **10, 6, 5, 5, 4, 3, 2, 1, 1, 1**

Enter an integer search key from 1 to 10 to delete the nodes: **1**

My updated linked list’s keys: **10, 6, 5, 5, 4, 3, 2**

The linked list has been freed.

**$ ./a.out**

My linked list’s keys: **9, 6, 5, 5, 4, 3, 2, 1, 1, 1**

Enter an integer search key from 1 to 10 to delete the nodes: **10**

The key **10** is not found!

The linked list has been freed.

**Grading Criteria:**

* Main program: 3 points
* createSortedLinkedListfunction: 9 points
* insertOneNewNodefunction: 9 points
* removeAllKeyNodes function: 9 points

**Note:**

* If your code does not compile with **–Wall** and **–Werror**, you will receive a **zero** for this assignment.
* You need to finish at least **three** peer reviews within three days of this lab. Otherwise, you will get a 20% penalty.
* You will lose points if you don’t have enough comments.