# COMP 2401 -- Tutorial #7

Linked Lists, Continued

## Learning Objectives

After this tutorial, you will be able to:

- insert and delete nodes from any position in a singly linked list

## Tutorial

In this tutorial, your functions should now return an integer representing the status of the function. There are defined status constants `C_OK` and `C_NOK` that you should use.

1. Download the file `T07.tar` from the tutorial page in *cuLearn*. Untar and read through the tutorial files.

2. Write a function `insertStudentAlpha(StudentList *stuList, StudentType *stu)` which takes as input a singly linked list (assumed to be sorted alphabetically by last name) and adds a student to `stuList` so that it remains in sorted order.

3. Write a function `deleteStudent(StudentList *stuList, char *fname, char *lname)` which deletes the student with matching first and last names from the list. Assume that `stuList` is sorted alphabetically by last name. Use this assumption to minimize the number of comparisons needed in the case where a student is not in the list. This function should free memory for both the node and the data. If a student cannot be found, your function should return `C_NOK`.

4. Write a function `cleanupList(StudentList *stuList)` which frees all memory associated with `stuList`.

## Exercises

1. Write a function `sortList(StudentList *stuList)` which takes an unsorted singly linked list and returns the list in alphabetically sorted order.

2. Write a function `gpaRange(StudentList *stuList, StudentList **result, int minGPA, int maxGPA)` which returns (via `result`) a new linked list which contains all the students in `stuList` having GPAs greater than or equal to `minGPA` and less than or equal to `maxGPA`.