CSCI 460: Operating Systems—**Assignment 2 (8 marks)**

This assignment is on process/thread synchronization. You will implement a variation of the Producer/Consumer problem. Assume the following:

a. The buffer is a doubly linked list (for this assignment, let us assume that it contains at most 20 nodes). Each node has a random integer value (which should be less than 40). Initially the linked list contains 3 nodes.

b. Producer #1 will generate a node and add it at the end of the linked list, the value of the new node is a random odd integer less than 40. Producer #2 will generate a node and add it at the end of the linked list, the value of the new node is a random even integer less than 40. When the buffer is full, both should generate a message and wait.

c. Consumer #1 will delete, from the head of the list, the first node whose value is odd (if the first node has an even value, then wait). Consumer #2 will delete, from the head of the list, the first node whose value is even (if the first node has an odd value, then wait). When the buffer is empty, both should generate a message and wait.

Each of the 4 processes/threads prints the contents of the linked list before and after it gets access to the linked list. Your output should contain the running result from every process/thread. **You must call <pthread.h> to use pthread_create to create a thread, etc, and you must implement your own protected linked list. Using an existing library/language which supports protected linked list operations, e.g. JAVA or C#, is forbidden.** (You might need to search the Internet to learn more about *pthread*, like using `mutex`. Due to the concurrency, if your program got stuck, analyze the reason from the output you have got.)

Date Due: **11:30pm on Monday, October 23, 2017 (on or before 11:30pm, Oct 23, 2017).** Load your source code and output as two separate files on D2L in the folder **Assignment 2**, preferably in the form of **family_name − 2.c** and **family_name − 2.output** (assuming that you are using C).