CSCI 2170 Fall 2006
Review for Test 3 (Tuesday, April 4th)

- **Linked list**
  - Know how to retrieve/insert/delete item to/from the beginning/middle/end of a **SORTED list**
  - Recursion with linked list (list insertion, list traversal)
  - Variations on linked list, including circular linked list, doubly linked list, circular doubly linked list (definition), and the advantage of using a dummy head in linked lists
  - Know how to perform insert/delete operations on a circular doubly linked list with a dummy head
  - **Adjacency list--**Understand how to build the adjacency list

- **Stack**
  - Main characteristic
  - Basic stack operations
  - Array-based, pointer-based implementation
  - Client program using stack ADT

- **Queue**
  - Main characteristic
  - Basic Queue operations
  - Array-based implementation
  - Why do we use circular array, instead of regular array?
  - Client program using queue ADT
  - Client program that using queue and stack to solve problems

Practice Questions:

1. What is the characteristic of a stack? A queue?
2. For each of the following situations, which of the ADTs would be most appropriate: (1) Queue, (2) Stack, (3) list (4) none of these ?
   a. The customers at a deli counter who take numbers to mark their turn
   b. An alphabetic list of names
   c. Integers that need to be sorted
   d. The boxes in a box trace of a recursive function
   e. A grocery list ordered by the occurrence of the items in the store
   f. The items on a cash register tape
   g. A word processor that allows you to correct typing errors by using the backspace key
   h. A program that uses backtracking
   i. A list of ideas in chronological order
   j. Airplanes that stack above a busy airport, waiting to land
   k. People who are put on hold when they call an airline to make reservations
   l. An employer who fires the most recently hired person
3. Show the copy constructor of a pointer based implementation of the ADT Stack
4. Show the implementation of the Stack ADT member function "Pop" with array implementation.
5. Show C++ function that creates a sorted list that is a combination of two existing sorted lists.
6. What does a doubly linked list that has 1 node look like? Draw a graph to illustrate the list.

<more on the back>

What does a circular linked list with 2 nodes look like? Draw a graph to illustrate the list.
What does a circular doubly linked list with dummy head and 2 (or 0) nodes look like? Draw a graph to illustrate the list.

7. Given an existing circular doubly linked sorted list with dummy head, show the code to :
    a. insert another node into the middle (or end) of this list.
    b. Delete the first node from the list
    c. Delete the node with key value "keyToDel" from the list.
8. Given a directed graph, show the adjacency list built for this graph
9. Show the value of **front**, **back, count,** and the content of the circular array implementation of a queue, after the following statements are executed (assume MAXQUEUE_SIZE = 8)

```
int i, j;
Queue   Q;

for (i=0; i<5; i++)
     Q.EnQueue(i, success);
Q.DeQueue(j, success);
Q.DeQueue(2*j, success);
Q.DeQueue(3*j, success);
```

10. Write a recursive function that prints the content of a linked list in reverse order.
11. Write a recursive listClass member function that maybe called by the destructor of listClass to free all dynamically allocated memory.