CSc 225 Assignment 3:
Linked Lists, Stacks, Queues, and Matching

The submission deadline is 11:55pm on Thursday, June 18th, 2020.

Submit your **`assignment3.pdf`** and **`ArrayMatch.java`** files through the Assignment 3 link on the CSC225 conneX page.

IMPORTANT: the files submitted **must** have **.pdf** and **.java** extensions.

1. In pseudocode, describe a recursive algorithm that reverses the elements in a singly linked list.

   Assumption: that the recursive algorithm is originally called with the **head** node in a linked list.

Algorithm reverse(n)
Input: The first node in a sequence of elements forming a singly linked list
Output: A reverse linked list (n ends up as the last node in the sequence).

```
if(n.next==null OR n==null) then
        return n
Node r ← reverse(n.next)
n.next.next ← n
n.next ← null
return r
```

2. Consider how the stack ADT could be implemented using two queues, **Q1** and **Q2**.

When a user **pushes** a number of elements to the stack and then **pops** an element, the last element (most recent) inserted should be returned and removed (LIFO).

a) Describe how the push and pop operations are implemented.

Push: if Q1 empty, enqueue Q2, else enqueue Q1

Pop: if Q1 isn't empty, transfer n-1 elements from Q1 to Q2, dequeue the n^{th} element and return it. Same for Q2 if Q1 is empty

b) What are the running times of the push() and pop() methods for this implementation?

push: O(1)
reason: literally just enqueueing elements

Pop: O(n)
reason: must copy elements from one queue to an other, for a total of n-1 elements $\in O(n)$