**Assignment 2**

**Level 1:  
A. Implement in JavaScript the function findMiddle(L) than we did in class. The  
DLinkedList class is included from the DLinkedList.js file.**

"use strict";

const List = require("./DLinkedList.js");

const findMiddle = function(L) {

    if(L.isEmpty)    // return null if L is empty

    return "No elements";

    let f = L.first();

    let l = L.last();

    while (f!==l && L.after(f)!== l){

        f=L.after(f);

        l=L.before(l);

        return L.element;

    }

};

**B. Describe, in pseudo-code, how to implement the stack ADT using a DLinkedList.  
What is the running time of the push() and pop() methods in this case?  
Implement a new Stack class in JavaScript based on (using) the DLinkedList class  
like done in A above.**

For List L

Algorithm push(e)

If L.isEmpty() then

L.first() = e

L.Last() = e

Else

L.after(L.last()) = e

L.last() = L.after(e)

Size = size +1

The running time is O(1)

Algorithm pop()

If L.is Empty() then

Return error

Data = L.last()

L.last() = L.before(L.last())

Size = size -1

Return data

The running time is O(1)

**C. Describe, in pseudo-code, how to implement the queue ADT using a  
DLinkedList. What is the running time of the enqueue() and dequeue() methods  
in this case? Implement a new Queue class in JavaScript using on the  
DLinkedList class.**

For List L

Algorithm enqueue(e)

If L.isEmpty() then

L.first() = e

L.Last() = e

Else

L.after(L.last()) = e

L.last() = L.after(e)

Size = size +1

The running time is O(1)

Algorithm dequeue()

If L.is Empty() then

Return error

Data = L.first()

L.first() = L.after(L.first())

Size = size -1

Return data

The running time is O(1).

**Level2:  
C-2.2 Describe, in pseudo-code, how to implement the queue ADT using two stacks.  
What is the running time of the enqueue() and dequeue() methods in this case?**