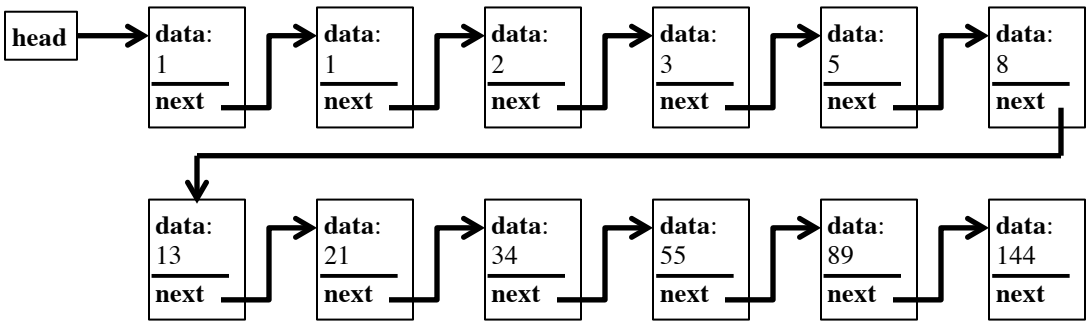
CMPS 12A – Fall 17
Lab 6 - Practice with Linked Lists
Due: Monday November 20 @ 8am

In this lab you will implement a queue using a linked list. A linked list is a simple data structure that organizes a list of data using a Node class; the Node class has *data* (in this assignment, an integer), and a *next* field which points to the next Node in the list. The Node class is defined recursively. Recall, a queue is a structure that organizes a list such that the first item in is the first item out.

The queue will have the following visual representation after the running the following java program:

```
public class LinkedListTest{
   public static void main(String[] args){
      // the first item in the linked list
      LinkedList fibNums = new LinkedList();
      fibNums.push(1);
      fibNums.push(1);
      fibNums.push(2);
      fibNums.push(3);
      fibNums.push(5);
      fibNums.push(8);
      fibNums.push(13);
      fibNums.push(21);
      fibNums.push(34);
      fibNums.push(55);
      fibNums.push(89);
      fibNums.push(144);
   }
}
```



Executing these following statements would give the accompanying outcomes:

```
// This statement prints 13
System.out.println(fibNums.find(7, fibNums.head));

// This statement prints 1 and removes first item in list
System.out.println(fibNums.pop());

// This statement prints 1 and removes first item in list
System.out.println(fibNums.pop());

// This statement prints 2 and removes first item in list
System.out.println(fibNums.pop());
```

head is a pointer of type Node that points to the beginning of the linked list. Each Node object has fields **data** (an integer) and **next** (a Node pointer pointing to the next Node in the list). Your Node class should be implemented the following way:

```
public class Node{
   int data;
   Node next;
   Node(int d, Node n){
      data = d;
      next = n;
   }
}
```

This program will be very similar to the stack example as shown in class (11/13), although the `insert()` and `find()` methods will be different. Your assignment is to implement the LinkedList class. Your LinkedList class will simply have two instance variables: **head**, of type Node (representing the front of the list), and **size**, of type `int` (representing the number of elements in the list). The LinkedList class will have the following instance methods:

`push()` – accepts an `int` parameter, adds a Node object with value passed as argument to the end of a linked list. Instance variable **size** should reflect this modification to the linked list. This method should account for the first time push called (i.e. initially, when the list is empty).

`pop()` – accepts no parameters, removes the first Node in the linked list, and returns the *data* in the Node. Instance variable **size** should reflect this modification to the linked list.

`insert()` – accepts two integer parameters, *n* and *d,* and a Node parameter *a*. A Node object with *data d* will be created and inserted in the linked list at position *n*. Use recursion with this method.

`find()` – recursive method `find()` that accepts an integer *n* as a parameter and a Node *a*, and returns the data as an integer at position *n* from a linked list. The first item in the linked list is at n=1. If the data is not found, `-1` is returned.

Once your LinkedList class has been completed, submit your LinkedList class to Canvas. Please note that all classes and method names should be identical to those shown in this document, otherwise points will be deducted.

Rubric

Program Compiles and Runs	3 points
Indentation/Comments	1 point
LinkedList class with instance variables	1 point
push()/pop() methods correctly implemented	1 point
insert() correctly implemented	2 points
find() correctly implemented	2 points