CSCI 112
Friday, July 23, 2010

Homework

LinkedList.java

```
class LinkedList{
	private class Node{
		public int data;
		public Node next;

		public Node(){
			data = 0;
			next = null;
		}
		public Node(int data){
			this.data = data;
			next = null;
		}
	}
	private Node head;
	private int size;

	public LinkedList(){
		head = null;
		size = 0;
	}
	public boolean isEmpty(){
		return size == 0;
	}
	public int size(){
		return size;
	}
	public int get(int pos){
		if( pos < 0 || pos >= size ){
			System.out.println( "Error: Out of Bounds: " + pos );
			System.exit(1);
		}
		Node thisNode = head;
		for( int i = 0; i < pos; i++ ){
			thisNode = thisNode.next;
		}
		return thisNode.data;
	}
	public void set(int pos, int data){
```

```
        if( pos < 0 || pos >= size ){
                System.out.println( "Error: Out of Bounds: " + pos );
                System.exit(1);
        }
        Node thisNode = head;
        for( int i = 0; i < pos; i++ ){
                thisNode = thisNode.next;
        }
        thisNode.data = data;
}
public int remove(int pos){
        if( pos < 0 || pos >= size ){
                System.out.println( "Error: Out of Bounds: " + pos );
                System.exit(1);
        }
        int value; //stores data being removed
        if(pos == 0){
                value = head.data;
                head = head.next;
        }
        else{
                Node thisNode = head;
                for( int i = 0; i < pos-1; i++ ){
                        thisNode = thisNode.next;
                }
                value = thisNode.next.data;
                thisNode.next = thisNode.next.next;
        }
        size--;
        return vlue;
}
public void insert(int pos, int value){
        if( pos < 0 || pos > size ){
                System.out.println( "Error: out of bounds: " + pos );
                System.exit(1);
        }
        Node myNode = new Node(value);
        if( pos == 0 ){
                myNode.next = head;
                head = myNode;
        }
        else{
                Node thisNode = head;
                for( int i = 0; i < pos-1; i++ ){
                        thisNode = thisNode.next;
                }
                myNode.next = thisNode.next;
                thisNode.next = myNode;
        }
```

```
                size++;
        }
}
```

Applications of Queues

Cryptography

Logic Tables

“And” Logic:

A	B	And(A, B)
0	0	0
0	1	0
1	0	0
1	1	1

“Or” Logic:

A	B	Or(A, B)
0	0	0
0	1	1
1	0	1
1	1	1

“Not” Logic

A	Not(A)
0	1
1	0

“XOR” Logic:

A	B	XOR(A, B)
0	0	0
0	1	1
1	0	1
1	1	0

Code:

```
int a = 2;
int b = 5;
int c = a ^ b; //a XOR b
System.out.println(c); //6
```

How Does this Relate to Queues?

Code:

```
Queue<Character> q = new Queue<Character>();

String message = "Attack at dawn.";
String key = "key";

for( int i = 0; i < key.length(); i++ ){
        q.enqueue(ket.charAt(i));
}
for( int i = 0; i < message.length(); i++ ){
        int a = (int)q.dequeue();
        int b = (int)message.charAt(i);
        System.out.println( (char)(a^b) );
        q.enqueue((char)a);
}
```