**Programming Assignment 3**

**50 points**

Your task is to develop a system that routes packets from source to destination among a series of routers on a network. You are provided code that contains a routing table, and generates packets with a source and a destination. Each router on the network must forward each packet to the next node on the way to its destination, recording that the packet traversed that router. The routing table for each router provides the appropriate next node.

Three classes are provided to you. The Routing class contains the main method, which performs set up, and generates a number of packets to traverse the network. Each packet generated is an instance of the Packet class. The Packet class holds the source and destination of the packet, as well as a list of the Routers the packet has traversed. The Routing class contains a static variable indicating the number of Packets currently traversing the network.

Each Router is represented by an instance of a Runnable router, and a single corresponding thread. The Routing class contains the routing table for the entire network. When it is constructed, each router receives its own portion of the routing table. The routing table for each Router is a two-dimensional integer array in which the index is the destination router, and the value at that index is the next router that should receive the packet on the way to its destination.

Your task is to implement the Router class. This is the class that forwards packets to the next router on the way to their destination. As you can see, Router implements Runnable. The main() method calls the addWork() method in Router, once it has created the Packet. The addWork() method enqueues the work to that particular thread. The end() method indicates to Router that, once there are no more packets in the network, it should return. The run() method (from Runnable) does the following: waits until the queue has some contents, then records the number of the current router in the packet. If the router is not the packet’s destination, it forwards that packet to the appropriate next router in the routing table by calling the addWork() method for that instance of Router, with the packet as the argument. Since there is one instance of Router for each router in the network, addWork() must add to the queue that corresponds to the correct router. Under no circumstances should any router process a packet more than once. If this is the destination router for the network, the router should record itself, but should not forward the packet any further. Router should use the monitors we discussed in class to manage its synchronization.

This program should use no explicit locks. In addition, it should use none of the Java concurrent data structures. Instead, it should rely on the “synchronized” keyword in Router to ensure synchronization between threads. Note that data structures such as ArrayList may be used concurrently, in a read-only manner. Be advised that you must be careful not to synchronize any more than necessary; in particular, you should not be synchronized when you call addWork() on another node, as this is likely to result in circular locking.

No modifications to any of the code, except the implementation of Router are necessary.

**Objectives:**

* Process all packets, and route them correctly to their destination.
* Correctly print packet routes.
* Exit correctly and cleanly.
* Use no timeouts or timers.
* Use monitors correctly and properly.
* Use no other synchronization.
* Properly use threads and queues.

Improperly laid out code, or code that doesn't meet the description of the assignment will be penalized, in addition to the above categories.

**Program Submission**

Zip the source files and submit to the dropbox. Gross violations such as missing files or not submitting a zip file will incur penalties. If the program has syntax errors, the grade will be 0: no exceptions.