\*\*README\*\*

The project wasn't completed. I got to the point all the neighbors get saved into the individual nodes, I can do the first pass of the periodic update (but when I tried putting it in a loop, it would just go infinite (used a sentinel that flags when a change doesn't happen to get out of the loop). Spent the whole weekend, and a few hours today on it, kept getting stuck, and other homework and projects are coming up, so decided to call it where it is on this one.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**Design specifications:**

Class for the Routers, which has an id, table for neighbors, and its Routing Table, getID, setId, it can add to the RT table (though for whatever reason it doesn't add more to it from dv packet updates, it only changes existing values), display the RT, add neighbors, display neighbors, initializeRT() from the neighbors table, and receiving dv packets. I did initialize the Router nodes to an id of -1 though so that way when I'm looping through the array of nodes I can set a check that looks if the id is >=0 and if so then it'll do the rest otherwise it'll do nothing. I have 4 structs which include; routingtable, neighbor, dvpacket, and "pairs" which is used inside the dvpacket struct. I have an event struct I made, but haven't used in any of my code or calls so it does nothing for my overall program at the point it is at. Didn't implement the linked list for the event queue yet either. I also have 2 functions which are displayingRoutingTable which displays all the RT for all the nodes, and periodicUpdate which is where the dvpackets are created and sent to all the other nodes.

Thinking about it now, it may have been cleaner and easier to either nest the Router class inside a network class or create a network class and have an array of the Router class inside of it so that way I can access all the values easier as well as keep all the functions inside the classes more organized, but if I nest them then they will be tightly coupled which can be considered bad.

No planned events listed have been added to be handled, because I set it up where nodes can only send to their neighbors and no matter what if they receive a dv packet they will compare it to its RT and make any adjustments needed, so I didn't take into account if a packet goes to the wrong node (it doesn't check the intended destination) and could end up causing all the RT to go bad if one did in fact go to the wrong node.