/***
* Project Report Template * Project 3 (Map Routing), ECE368
***/
Name: Nausherwan Korai Login: nkorai
/***
* Explain your overall approach to the problem and a short general
summary of your solution and code.
***/
My overall approach was to first parse in the input map file and create the map. Milestone 2 was very helpful because it let me do just that. I then took this map and parsed in the query, which gave me a source and destination. My implementation of the program involved setting the distance of all nodes to infinity (10001). I would then set the source's distance as 0, as the source's distance to itself is 0. The algorithm would then search for the node with the smallest distance that hadn't been visited, calculate the distance between it and all of it's edges, and then set it to visited. It would then repeat this, looking for the next smallest distance that hadn't been visited and calculating the distances to it's edges, all the way until the destination has been set as visited. All along the way, each node would point back to the node that allowed the shortest distance to the source. The destination contained the total distance and by reversing the linked list and printing it we had the route.

/***
* Known bugs / limitations of your program / assumptions made.
***/
Bugs: None that I am aware of.
Assumptions: 1) Distances above 10000 are infinity. 2) All vertices will connect to at least one other vertex. 3) The graph is undirected. 4) There are no "dead ends", i.e. every vertex has >= 2 edges.
Limitations: I implemented a linked list structure to the program, and as such searching through the linked list takes time (search is in linear time) and thus large maps take longer to search through. When searching through usa.txt with usa1.txt query, the program took a few minutes to come back with the answer.

/***
* List whatever help (if any) that you received.
***/
I have a job as Web Developer on campus, and my supervisor helped explain Dijkstra's algorithm to me.

/***
* Describe any serious problems you encountered.
***/
This is the first time I've implemented a linked list of linked lists and it was tough to keep things straight conceptually in my head.

/***
* List any other comments/feedback here (e.g., whether you enjoyed
doing the exercise, it was too easy/tough, etc.).
***/
Please compile with the –lm flag. My entire project is contained in one file; I've commented it as best as I could. By far my most favorite coding project of my time at Purdue. It was amazing to see the theory and then actually implement it and make it work. It was definitely tough but that made figuring it out so much better. Loved it.