/***
 * Project 3 (Map Routing), ECE368
 ***/

Name: Dan Suciu
Login: suciu
Compile: gcc -Werror -Wall -O3 -lm shortestpath.c -o shortestpath

/***
 * Approach to the problem / short general summary of solution & code.
 ***/

My approach was to build a codebook in which the column index is the vertex and every non-negative integer in the successive rows is the index of a vertex which is a neighbor to that vertex. There is also a matrix of coordinates which hold the x and y coordinates of each vertex represented by the index. The start and end vertices are scanned in from the query and sent to the Dijkstra algorithm with the neighbor codebook and the coordinate codebook one query at a time. In the Dijkstra algorithm, we loop unitl we set the visited flag of the end vertex, meaning that at some point, the path from any vertex to the end vertex was the shortest distance existing in the distance array, all of which are initialized to INT_MAX axcept for the start vertex which has a distance of 0. Every iteration of the outer loop we find the shortest distance, and visit that vertex. We then calculate the distances to all of that vertex's neighbors and replace their distances if it is smaller than the current one for that neighbor and we haven't visited that vertex yet. When we replace a distance, we also set that neighbors previous value to the vertex we were visiting, keeping track of the path taken. When this exits we have the distance to the end vertex and we can backtrack through previous to find the path taken from the start vertex.

/***
 * Known bugs / limitations of your program / assumptions made.
 ***/

The nature of my algorithm requires a neighbor matrix which is the size of the number of vertices by the number of verices, as in theory every vertex could be a neighbor of every vertex. This matrix must be initialized to -1 to control when we have no more valid neighbors to look at. Initializing this matrix is very slow however, and for the usa.txt file this initialization takes around 20 seconds. Other than this there are no limitations or bugs in the program.

/***
 * List whatever help (if any) that you received.
 ***/

Mitch Bouma and I often work together to discuss algorithm design and help eachother debug code, however we would never plagarize off of eachother.

/***
 * Describe any serious problems you encountered.
 ***/

When writing the Djikstra Algorithm I often would find myself in infinite loops where my code would not move past the vertex it was currently looking at after a certain point. The fix for this was an overall structure change in the algorithm to consider all distances to vertices as possible vertices to visit next, whereas before I was only considering the neighbors of the current vertex as possible vertices to visit next.

/***
 * Comments / Feedback.
 ***/

This was an easier project than the second project, and harder than the first. I did enjoy doing this project, but in the future I would consider teaching graphs using structures and linked lists. If I had more

time and a fresh start I feel I would be able to remove the initialization bug described above using a linked list for my neighbors.