```
/*****************************************************************
 * Project Report Template
 * Project 3 (Map Routing), ECE368
 *****************************************************************/

Name:       Ranjan Behl
Login:      rbehl

/*****************************************************************
 *  Explain your overall approach to the problem and a short
 *  general summary of your solution and code.
 *****************************************************************/

This project asked us to create an implementation of Dijkstra's algorithm
which is one of the most common shortest path algorithms. To implement the
algorithm, I broke the project up into multiple steps and the first step was
creating an effective graph structure. I used the adjacency list
representation for my graph because first it was required for milestone 2,
second because it is the most memory efficient way. After this point the
algorithm recreated the graph from the input files, for this part I had to be
somewhat creative in how I assigned the x and y values and weight for each
adjacency node. The edge list(connection) of each vertex was a priority queue
which stored the list in ascending order based on the vertex number. Then I
read in the query and created a minheap for all the vertices with the
distance from source (specified by the query file input) to each vertex. The
source is set to 0 and then the standard Dijkstra's algorithm is used. For
each query given I had two arrays to hold the path one to trace it forward,
the other to reverse the traced path so it could be printed.

Compile the Code: gcc -std=c99 -Werror -Wall -lm Pathfinder.c -o shortestpath

/*****************************************************************
 *  Known bugs / limitations of your program / assumptions made.
 *****************************************************************/
I assumed that only integers values were given for the x and y coordinates
and that all the vertices are whole numbers.
The limitations are that the max vertex number is 100k and the max value of x
and y is 10k.

/*****************************************************************
 *  List whatever help (if any) that you received.
 *****************************************************************/
I looked back to my project 2 code for the help with my some of the code
I also referenced the textbook and the Wikipedia page to gain a better
understanding of Dijkstra's algorithm.


/*****************************************************************
 *  Describe any serious problems you encountered.
 *****************************************************************/
Once I did my initial implementation, I found out that the way I had written
my structures was ineffective and caused me multiple bugs. To fix this I
changed the structures and decided to implement a min heap approach.  I did
this because the min heap reduced the runtime down to O(VlogE), where E is
the edge number.
```

```
/*****************************************************************
 *  List any other comments/feedback here (e.g., whether you
 *  enjoyed doing the exercise, it was too easy/tough, etc.).
 *****************************************************************/
```
Overall, I found this project to be interesting and useful, as shortest path is one of the most important things to know how to implement. However, I found this project much easier than project 2 which was the hardest project to data in my opinion.