```
/*********************************************************************
 * Project Report Template
 * Project 3 (Map Routing), ECE368
 **********************************************************************/
       Name: Asheem Chhetri
       Login: achhetri@purdue.edu
/*********************************************************************
 * Explain your overall approach to the problem and a short
 * general summary of your solution and code.
 **********************************************************************/
```

1. Read the map file first and populate an array of 'Vertice' points with the information provided in the map file.
2. The adjacency list for each Vertice is populated using edge information.
3. For indexing/locating the Vertices in the Vertice array, an alternate integer array is used to store location of each ID value.
4. The query file is then read and each query processed.
5. The source element is located in O(1) time using the addressing integer array and is placed at the top of the Vertice array.
6. The Vertice array is used to maintain both the min-heap and the visited list.
7. At the start of each iteration, the first element of min-heap is swapped with the last element in the min-heap and then the new first element is downward heapified.
8. After this, each element in adjacency list of above swapped element is updated and upward heapified if need, provided not present in the completed list. (Check completed list using binary search)
9. Loop repeats until either min-heap is empty or destination element has been popped from the min-heap.
10. The path is calculated by following predecessors from destination.
11. Vertice array is then reset (distance and predecessor reset).
12. New query is processed until no more queries remain.

To run the code, add " -lm" to the gcc command (because I included math.h library in the code)

```
gcc -Werror -Wall shortestpath.c -lm -o myShortestpath
```

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

Only assumption I made was that, the map file will have vertex data with ID number in continuous ascending order starting from the index of 0.

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

None taken, except the slides provided by instructor.

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

Initial idea was to use Linked List, but it made execution time beyond 15 minutes, finally decided to use heapify method.

```
/*********************************************************************
 *  List any other comments/feedback here (e.g., whether you
 *  enjoyed doing the exercise, it was too easy/tough, etc.).
 **********************************************************************/
```

The exercise was enjoyable and challenging although not hard. It was not too time consuming either.