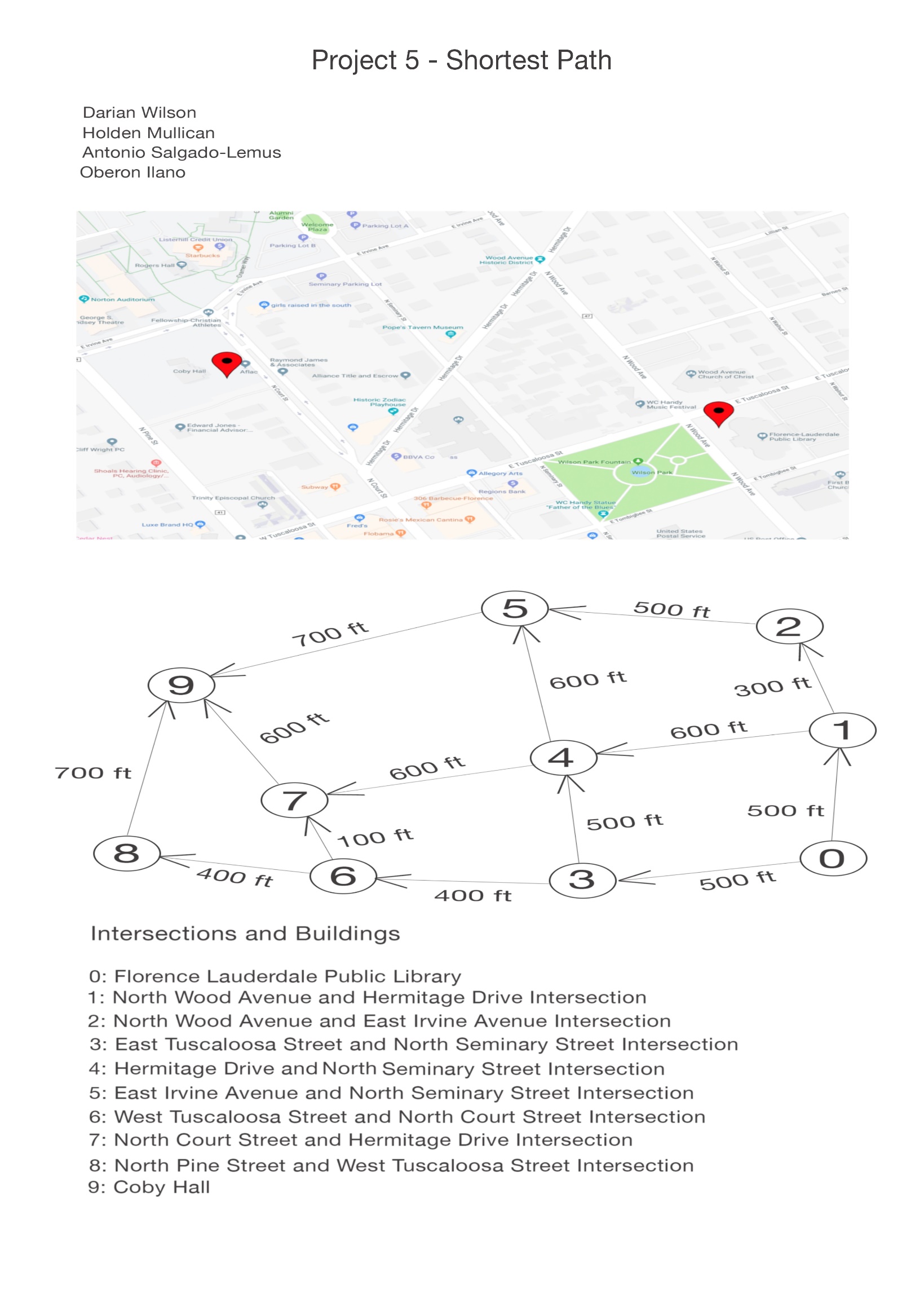
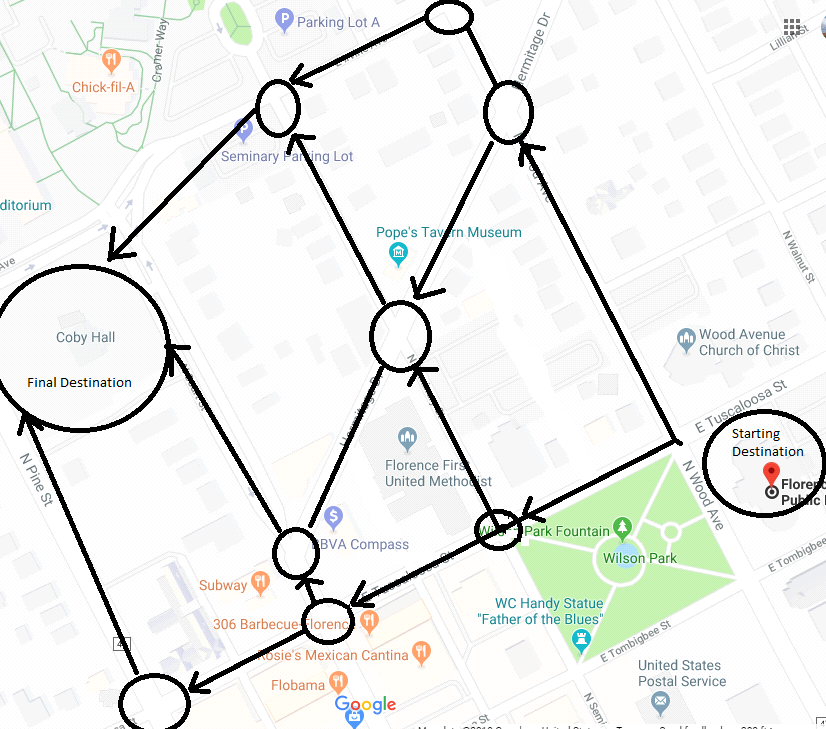
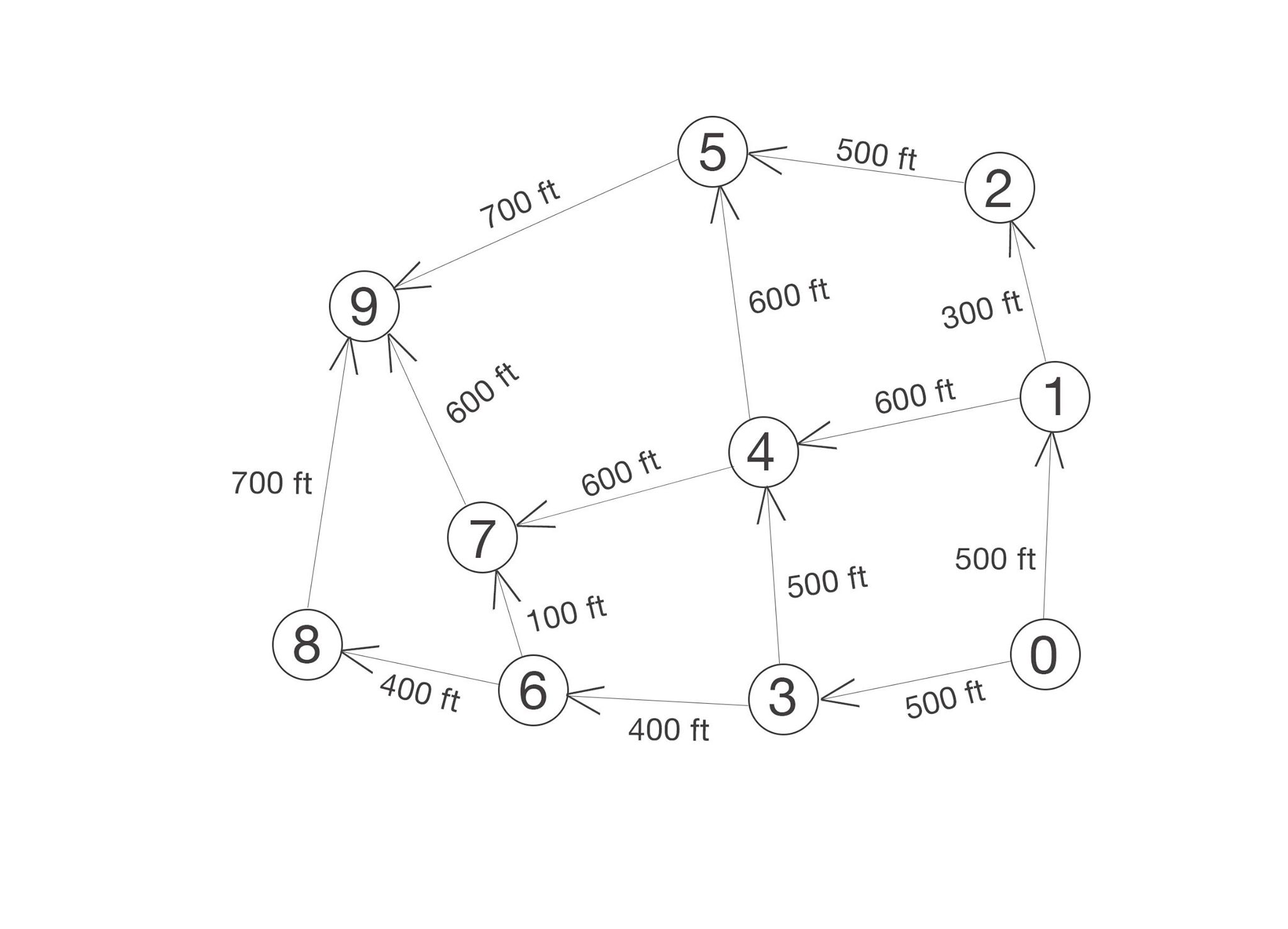
**Initial Idea Description:**

The initial idea for this program is to find the shortest path from Florence-Lauderdale Public Library to Coby Hall. There are 8 road intersections on the route from the Library to Coby Hall. There are 10 vertices that are in the graph/map. These vertices represent all the intersections, the public library, and the Coby hall in the real world. The edges represent the roads between the intersections and the two buildings. The weight of the edges will be measured by feet. The program will find the shortest path from the public library to every intersection and to Coby Hall. The data structure is a graph that is represented by using dijkstra’s algorithm, weighted graph, priority queue, and adjacency list.





Graph:



**Updated Functional Requirements:**

* To calculate the minimum distance and keep track of the vertices that have been visited:

int getMinDist(int hArray[], bool sptSet[])

* To display the distance/edges cost/weight from the source:

void DisplayShortPath(int\* distance, int n)

* To calculate and determine the shortest path in the graph:

void ShortestPath(int source, int destination)

* To add edges to the source and to the destination:

void addEdge(int source, int destination, int weight)

**Updated Non-Functional Requirements:**

Readability

Source documentations and comments.

Usability

* Input data from text files:

void customGraph(BinaryHeap &graph)

* Menu Option

int Menu();

**Updated Mock-Up of the User Interface:**

List of Locations

-----------------------

1.) Intersection 1 (N Wood Ave & Hermitage Dr)

2.) Intersection 2 (N Wood Ave & E Irvine Ave)

3.) Intersection 3 (E Tuscaloosa St & N Seminary St)

4.) Intersection 4 (Hermitage Dr & N Seminary St)

5.) Intersection 5 (E Irvine Ave & N Seminary St)

6.) Intersection 6 (W Tuscaloosa St & N Court St)

7.) Intersection 7 (N Court St & Hermitage Dr)

8.) Intersection 8 (N Pine St & W Tuscaloosa St)

9.) Coby Hall

-----------------------

+ - Find and output shortest path from Florence Library to (+Numbered Location)

S - Show map (S)

Q - Quit (Q)

**Updated Back End Data Structures Description:**

AdjList - A constructor will initialize all of the member variables of the heap

Incoming Data: source(int), destination(int) and weight (int)

Outgoing Data: source(int), destination(int) and weight (int)

Priority Queue - to store vertices/nodes that are being observed

addEdge - This method will add an edge in the graph

Incoming Data:

- Parameter 1: integer source

- Parameter 2: integer destination

Outgoing Data: N/A

ShortestPath - This method will implement shortest path algorithm upon the source that is passed in.

Incoming Data:

- Parameter 1: integer source

- Parameter 2: integer destination

Outgoing Data: N/A

DisplayShortPath - This method will print the solution to our shortest path algorithm

Incoming Data:

- Parameter 1: integer array of distances

- Parameter 2: The source

Outgoing Data: distance value (int), source and destination (string)

**Inputs & Outputs Descriptions:**

List of Locations

-----------------------

1.) Intersection 1 (N Wood Ave & Hermitage Dr)

2.) Intersection 2 (N Wood Ave & E Irvine Ave)

3.) Intersection 3 (E Tuscaloosa St & N Seminary St)

4.) Intersection 4 (Hermitage Dr & N Seminary St)

5.) Intersection 5 (E Irvine Ave & N Seminary St)

6.) Intersection 6 (W Tuscaloosa St & N Court St)

7.) Intersection 7 (N Court St & Hermitage Dr)

8.) Intersection 8 (N Pine St & W Tuscaloosa St)

9.) Coby Hall

-----------------------

+ - Find and output shortest path from Florence Library to (+Numbered Location)

* This option will show the user the shortest path from the available locations

S - Show map (S)

* This option will show the user a map of the location

Q - Quit (Q)

**Detailed, Predicted Team Schedule:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Name** | **Time** | **Worked On** |
| April 2, 2019 | Oberon | 2:00 PM | Real world ideas. |
| April 2, 2019 | Darian | 2:00 PM | Real world ideas. |
| April 2, 2019 | Antonio | 2:00 PM | Real world ideas. |
| April 2, 2019 | Holden | 2:00 PM | Real world ideas, submitted real world proposal. |
| April 3, 2019 | Oberon | 3:00 PM | Design document topics including initial idea description, predicted team schedule, functional requirements. |
| April 3, 2019 | Antonio | 8:00 PM | Design documents topics including non-functional requirements, and algorithm. |
| April 4, 2019 | Darian | 4:00 PM | Source code implementation, including binary heap header file declarations, and definition using adjacency matrix. |
| April 4, 2019 | Holden | 4:15 PM | Created Work Log template. |
| April 5, 2019 | Oberon | 10:00 AM | Source code implementation using adjacency matrix. |
| April 8, 2019 | Darian | 10:30 AM | Source code implementation for binary heap driver. |
| April 9 | Darian | 11:00 AM | Source code implementation, modified adjacency matrix to adjacency list. Created binary heap Class, main driver, menu function and shortest path method. |
| April 9 | Oberon, Antonio | 11:05 AM | Created mutator (addEdge function) declaration and definition |
| April 9 | Antonio | 11:06 AM | Created constructor declaration and definition. |
| April 9 | Darian, Antonio | 11:30 AM | Created display function declaration and definition. |
| April 9 | Holden | 12:00 PM | Source code implementation; created input file function. |
| April 9 | Antonio | 1:21 PM | Updated design documentation. |
| April 9 | Oberon | 1:25 PM | Updated team work log. |
| April 10 | Oberon | 3:30 PM | Modified graph picture and created graph pictures handouts. |
| April 10 | Oberon | 3:35 PM | Modified source code implementation for input, display and menu function. |
| April 10 | Oberon | 8:00 PM | Added more photos for front end. |
| April 11 | Oberon | 9:00 PM | Modified map’s scale, rounded distances between intersections, and add path directions to pictures for readability. |
| April 15 | Holden | 2:00 PM | Created slide show for presentation. |
| April 15 | Oberon | 8:45 PM | Updated design document including detailed team scheduled, team work log, photos, handouts for presentation, and test plans. |

**Updated Test Plan:**

Case #1:

Main Menu Selection:

+ - Find and output shortest path from Florence Library to (+Numbered Location)

S - Show map (S)

Q - Quit (Q)

Input:

+0

Output:

Invalid entry

Case #2:

Main Menu Selection:

+ - Find and output shortest path from Florence Library to (+Numbered Location)

S - Show map (S)

Q - Quit (Q)

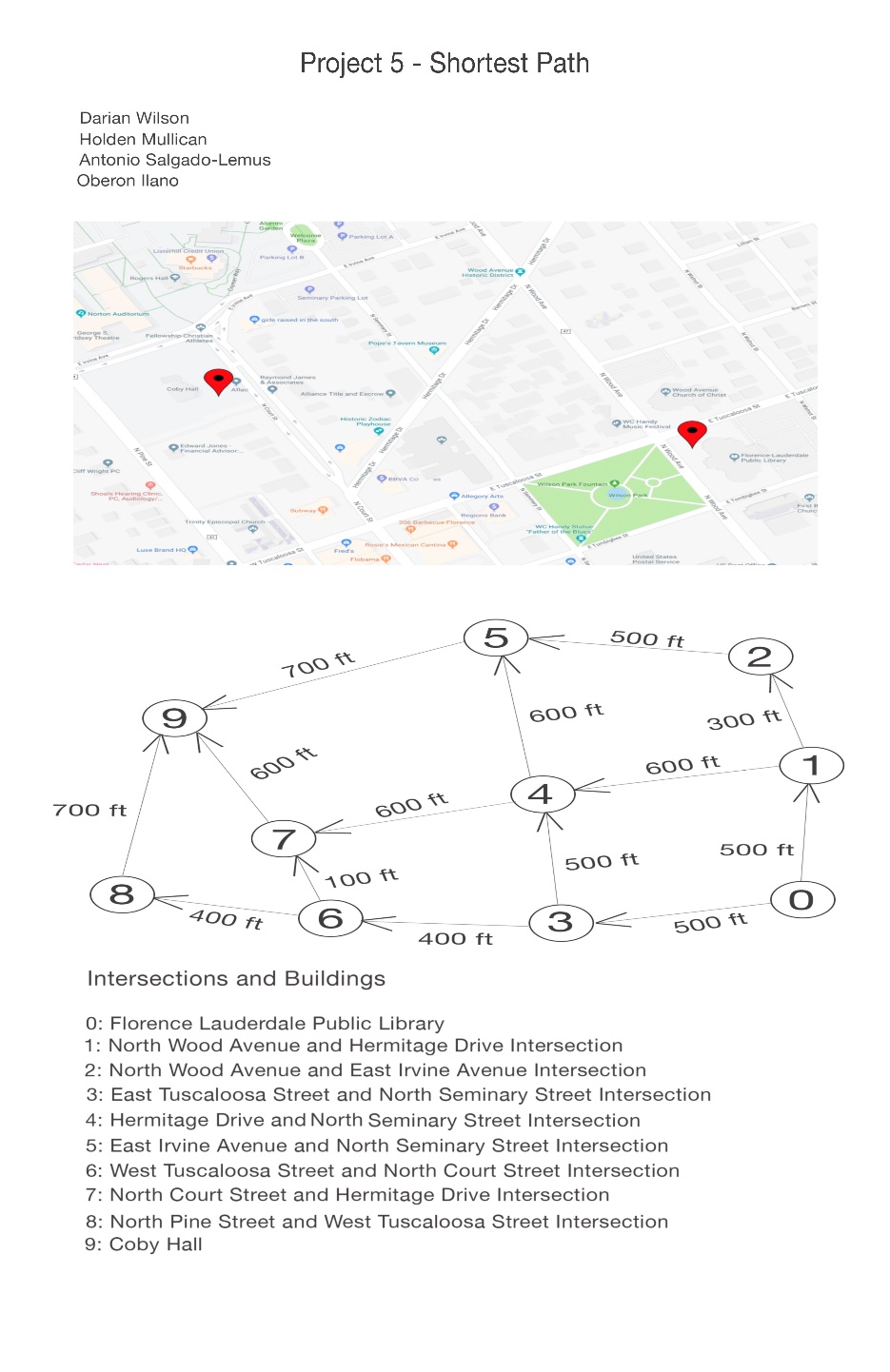
Input:

S

Output:

Press any key to continue…

Show Picture:



Case #3:

Main Menu Selection:

+ - Find and output shortest path from Florence Library to (+Numbered Location)

S - Show map (S)

Q - Quit (Q)

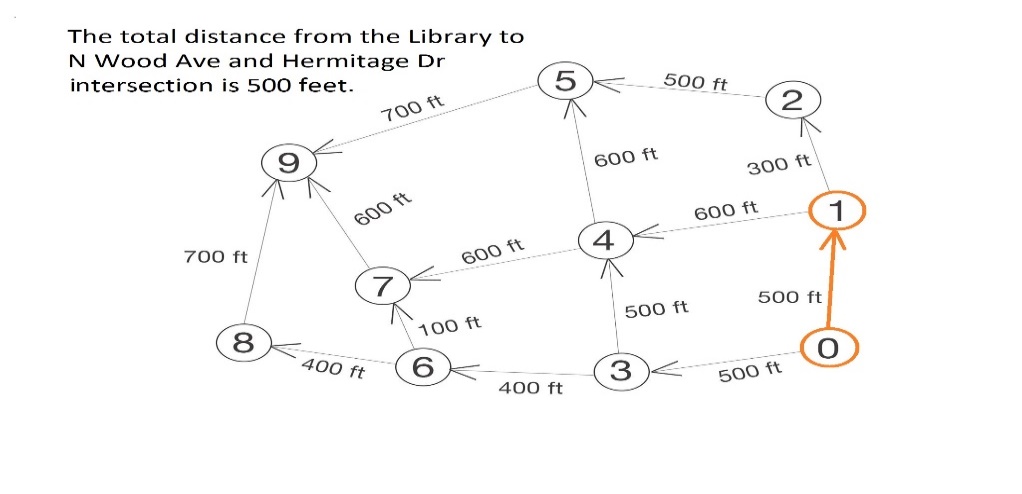
Input:

+1

Output:

The shortest path from Library to Intersection 1 (N Wood Ave & Hermitage Drive) is 500 feet. The shortest path is as follows: Library -> N Wood Ave & Hermitage Drive Intersection

Show Picture:



Case #4:

Main Menu Selection:

+ - Find and output shortest path from Florence Library to (+Numbered Location)

S - Show map (S)

Q - Quit (Q)

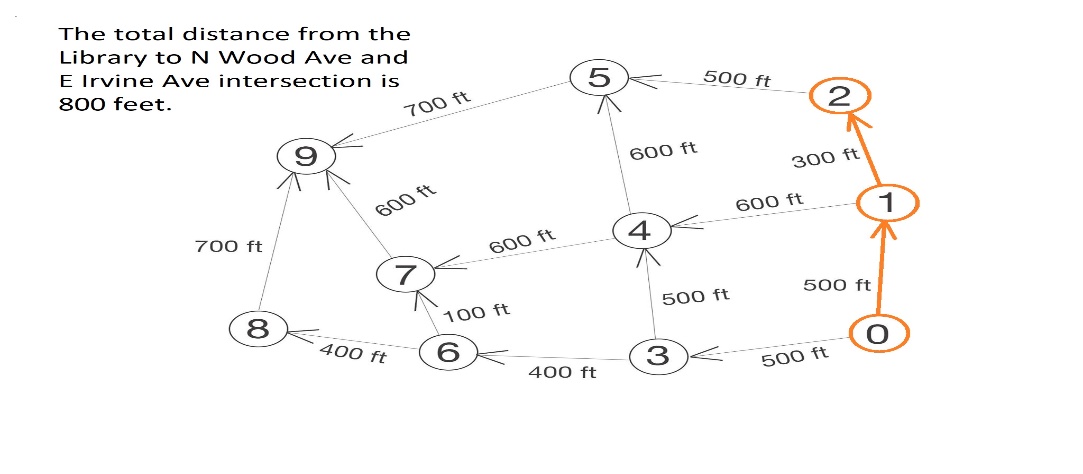
Input:

+2

Output:

The shortest path from Library to Intersection 2 (N Wood Ave & E Irvine Ave) is 800 feet. The shortest path is as follows: Library -> N Wood Ave & Hermitage Drive Intersection -> N Wood Ave & E Irvine Ave Intersection

Show Picture:



Case #5:

Main Menu Selection:

+ - Find and output shortest path from Florence Library to (+Numbered Location)

S - Show map (S)

Q - Quit (Q)

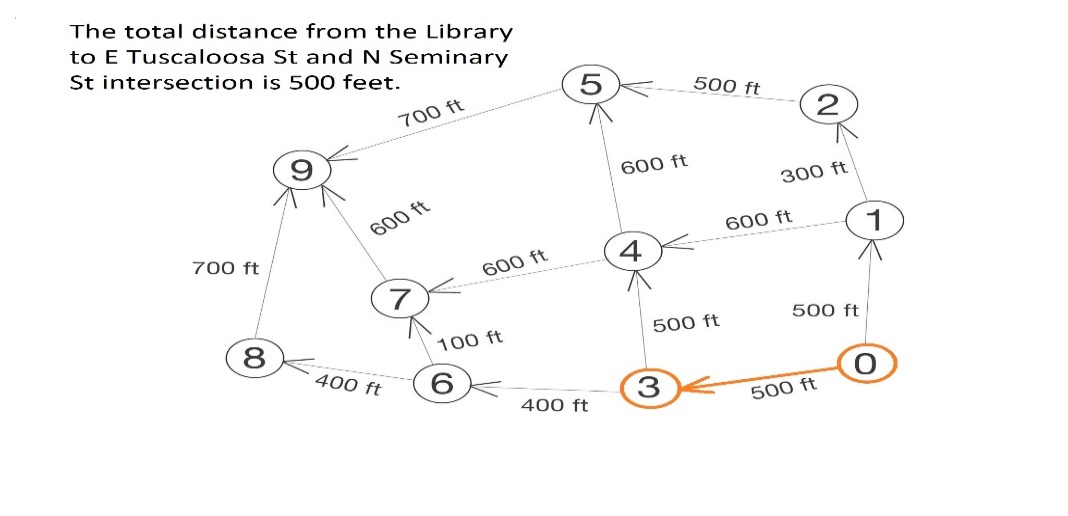
Input:

+3

Output:

The shortest path from Library to Intersection 3 (E Tuscaloosa St & N Seminary St) is 500 feet. The shortest path is as follows: Library -> E Tuscaloosa St & N Seminary St Intersection

Show Picture:



Case #6:

Main Menu Selection:

+ - Find and output shortest path from Florence Library to (+Numbered Location)

S - Show map (S)

Q - Quit (Q)

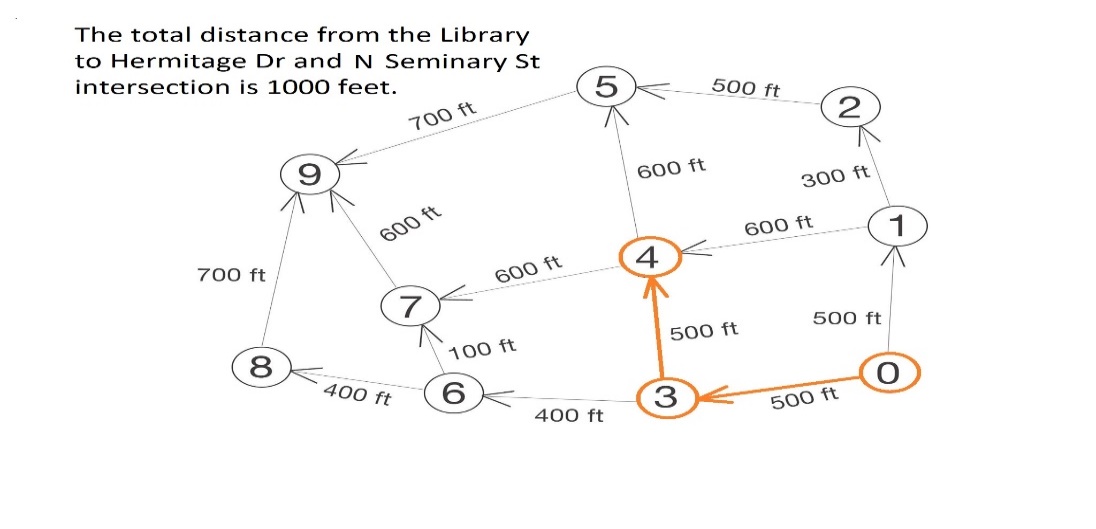
Input:

+4

Output:

The shortest path from Library to Intersection 4 (Hermitage Drive & N Seminary St) is 1000 feet. The shortest path is as follows: Library -> E Tuscaloosa St & N Seminary St Intersection -> Hermitage Drive & N Seminary St Intersection

Show Picture:



Case #7:

Main Menu Selection:

+ - Find and output shortest path from Florence Library to (+Numbered Location)

S - Show map (S)

Q - Quit (Q)

Input:

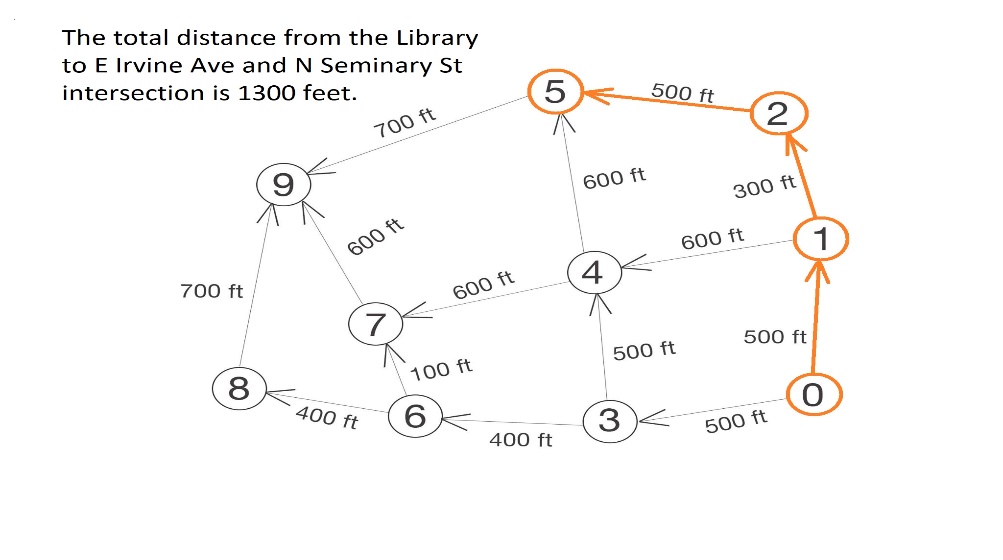
+5

Output:

The shortest path from Library to Intersection 5 (E Irvine Ave & N Seminary St) is 1300 feet. The shortest path is as follows: Library -> N Wood Ave & Hermitage Drive Intersection -> N Wood Ave & E Irvine Ave Intersection -> E Irvine Ave & N

Seminary St Intersection

Show Picture:



Case #8:

Main Menu Selection:

+ - Find and output shortest path from Florence Library to (+Numbered Location)

S - Show map (S)

Q - Quit (Q)

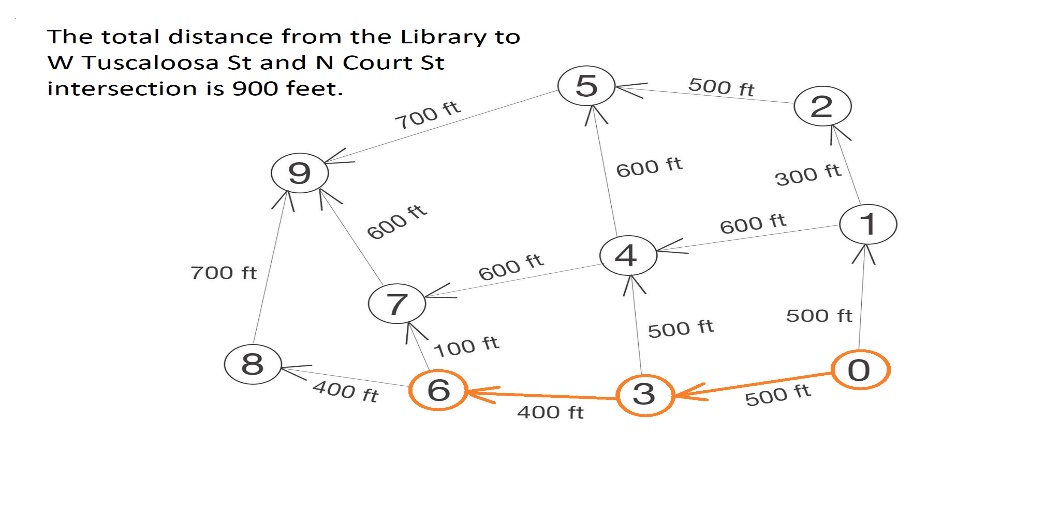
Input:

+6

Output:

The shortest path from Library to Intersection 6 (W Tuscaloosa St & N Court St) is 900 feet. The shortest path is as follows: Library -> E Tuscaloosa St & N Seminary St Intersection -> W Tuscaloosa St & N Court St Intersection

Show Picture:



Case #9:

Main Menu Selection:

+ - Find and output shortest path from Florence Library to (+Numbered Location)

S - Show map (S)

Q - Quit (Q)

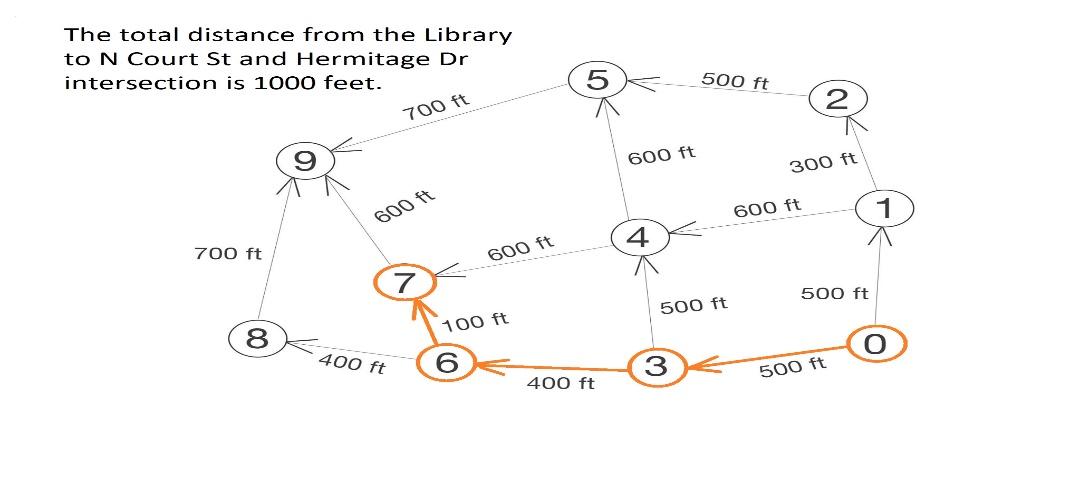
Input:

+7

Output:

The shortest path from Library to Intersection 7 (N Court St & Hermitage Drive) is 1000 feet. The shortest path is as follows: Library -> E Tuscaloosa St & N Seminary St Intersection -> W Tuscaloosa St & N Court St Intersection -> N Court St & Hermitage Drive Intersection

Show Picture:



Case #10:

Main Menu Selection:

+ - Find and output shortest path from Florence Library to (+Numbered Location)

S - Show map (S)

Q - Quit (Q)

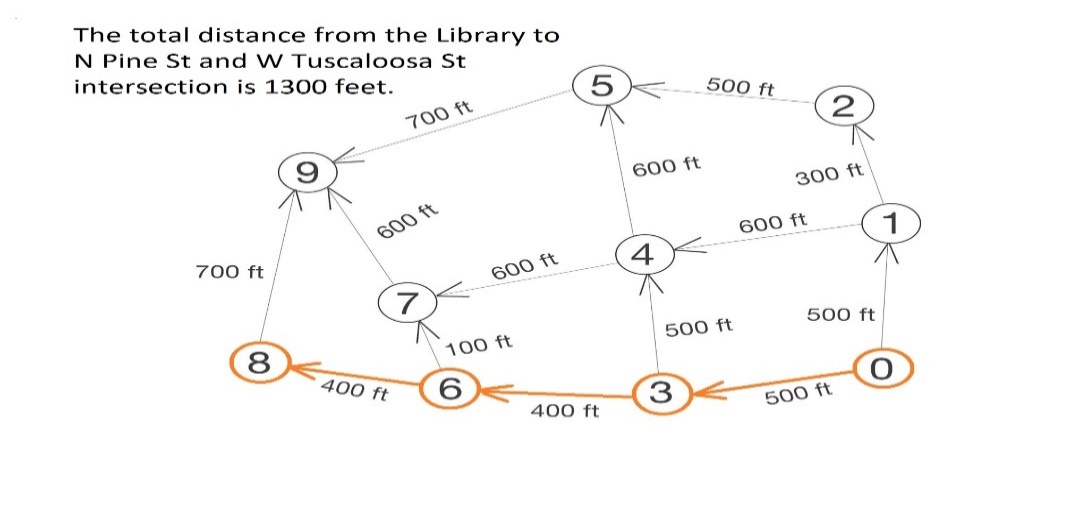
Input:

+8

Output:

The shortest path from Library to Intersection 8 (N Pine St & W Tuscaloosa St) is 1300 feet. The shortest path is as follows: Library -> E Tuscaloosa St & N Seminary St Intersection -> W Tuscaloosa St & N Court St Intersection -> N Pine St & W Tuscaloosa St Intersection

Show Picture:



Case #11:

Main Menu Selection:

+ - Find and output shortest path from Florence Library to (+Numbered Location)

S - Show map (S)

Q - Quit (Q)

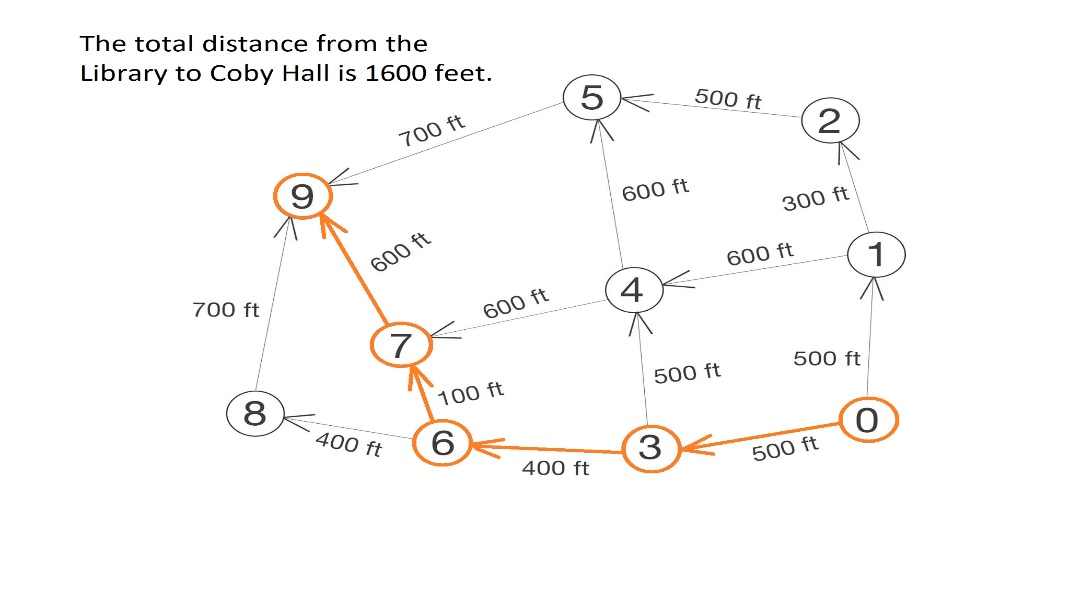
Input:

+9

Output:

The shortest path from Library to Coby Hall is 1600 feet. The shortest path is as follows: Library -> E Tuscaloosa St & N Seminary St Intersection -> W Tuscaloosa St & N Court St Intersection -> N Court St & Hermitage Drive Intersection -> Coby Hall

Show Picture:



Case #12:

Main Menu Selection:

+ - Find and output shortest path from Florence Library to (+Numbered Location)

S - Show map (S)

Q - Quit (Q)

Input:

A

Output:

Invalid Choice

Case #13:

Main Menu Selection:

+ - Find and output shortest path from Florence Library to (+Numbered Location)

S - Show map (S)

Q - Quit (Q)

Input:

+?

Output:

Invalid Entry

Main Menu Selection:

+ - Find and output shortest path from Florence Library to (+Numbered Location)

S - Show map (S)

Q - Quit (Q)

Input: Q

Output: Thanks for testing

Exit Program

Case #14:

Main Menu Selection:

+ - Find and output shortest path from Florence Library to (+Numbered Location)

S - Show map (S)

Q - Quit (Q)

Input:

+a

Output:

Invalid Entry

**Algorithm/Solution Overview:**

* Set all vertices distances = infinity except for the source vertex, set the source distance = 0.
* Push the source vertex in a min-priority queue in the form (distance, vertex), as the comparison in the min-priority queue will be according to vertices distances.
* Pop the vertex with the minimum distance from the priority queue (at first the popped vertex = source).
* Update the distances of the connected vertices to the popped vertex in case of "current vertex distance + edge weight < next vertex distance", then push the vertex  
  with the new distance to the priority queue.
* If the popped vertex is visited before, just continue without using it.
* Apply the same algorithm again until the priority queue is empty.