CS 3610 Driving Directions (No Late Submission)

Objectives: Implement a graph and use it to find the shortest path between two nodes.

The graph will be given in a file where each line represents a vertex node and its neighbor vertices and their edge weights (distances).

For example, the following line shows the distance between Akron, Cleveland, and Cambridge

```
Akron 39 Cleveland 125 Columbus  83 Cambridge
```

Akron is 39 miles from Cleveland
Akron is 125 miles from Columbus
Akron is 83 miles from Cambridge

Sample Graph File:

```
Akron 39 Cleveland 125 Columbus 83 Cambridge
Columbus 125 Akron 80 Cambridge 148 Cleveland
Cambridge 80 Columbus 83 Akron 103 Athens
Athens 74 Columbus 103 Cambridge
Cleveland 39 Akron 148 Columbus
Toledo 118 Cleveland 135 Akron 142 Toledo
```

Sample directed graph file:

```
S 2 B 4 A
A 1 B 4 D 2 C
B 2 D
C 3 T
D 3 T
T
```

Write a C++ program that creates a graph from the given file and determines the shortest route between two cities or all cities given a starting point.

Your program should input the graph file, the source, and the destination from the command line.
Test your program with several graphs (provided).

Sample Runs:

```
./route mygraph.txt Athens Cleveland
```
Display the route, distances, and total distance

```
./route mygraph.txt Athens    London
```
Display no route exists, destination unreachable

```
./route mygraph.txt London Athens
```
Display no route exists, source location does not exist

```
./route mygraph.txt Athens
```
Display the shortest paths from Athens to all the destinations

```
./route
```
Display a usage message and show all the available vertices

Sample output:
City 1 <distance> City 2 <distance> City 3 Total: <total distance>

```
Athens (74) -> Columbus (148) -> Cleveland        Total: 222
```

Display error and usage messages if the number of arguments is incorrect or the file does not exist.

Grading:
Programs that contain syntax errors will earn zero points.
Programs that use global variables, other than constants, will earn zero points.

(90 points):

- (20 points) Creating the graph
- (40 points) Finding shortest paths
- (20 points) 5 points for each of the options above
- (5 points) Error detection
- (5 points) Style and documentations

Follow the coding style outline on GitHub:

https://github.com/nasseef/cs2400/blob/master/docs/coding-style.md