1. Download the following files and create a workspace using them:

<https://markbowman.org/231/Lab24.zip>

The code file contains global arrays with node names and edge information. Use them to complete the lab, but don’t create any other global variables.

2. Compile and run the program. You should see this displayed:

Sample Run

A B C D E

A -1.00 2.00 1.70 -1.00 -1.00

B -1.00 -1.00 3.50 1.50 -1.00

C -1.00 -1.00 -1.00 2.40 -1.00

D -1.00 -1.00 -1.00 -1.00 4.10

E 9.99 -1.00 -1.00 -1.00 -1.00

Text

Description automatically generated

3. Write code that will prompt the user for a source name, then set source\_index to the matching string in the name[] array.

4. Write code that will prompt the user for a target name, then set target\_index to the matching string in the name[] array.

5. Using source\_index, target\_index, and the edge[][] array, display the distance between the nodes, or a warning if there is no valid edge.

Sample Runs

A B C D E

A -1.00 2.00 1.70 -1.00 -1.00

B -1.00 -1.00 3.50 1.50 -1.00

C -1.00 -1.00 -1.00 2.40 -1.00

D -1.00 -1.00 -1.00 -1.00 4.10

E 9.99 -1.00 -1.00 -1.00 -1.00

Enter source: A

Enter target: C

A to C = 1.70

A B C D E

A -1.00 2.00 1.70 -1.00 -1.00

B -1.00 -1.00 3.50 1.50 -1.00

C -1.00 -1.00 -1.00 2.40 -1.00

D -1.00 -1.00 -1.00 -1.00 4.10

E 9.99 -1.00 -1.00 -1.00 -1.00

Enter source: B

Enter target: E

No edge from B to E

6. Test your program with three valid and one invalid edge, and save the outputs.

7. Create a loop that will allow the user to move from node to node along valid edges and sum up the total distance. Prompt the user for a starting node, then loop until they enter a target of "Quit". Do not use a break to exit the loop.

Sample Run

A B C D E

A -1.00 2.00 1.70 -1.00 -1.00

B -1.00 -1.00 3.50 1.50 -1.00

C -1.00 -1.00 -1.00 2.40 -1.00

D -1.00 -1.00 -1.00 -1.00 4.10

E 9.99 -1.00 -1.00 -1.00 -1.00

Enter source: A

Enter target: C

A to C = 1.70

Enter target: D

C to D = 2.40

Enter target: B

No edge from D to B

Enter target: E

D to E = 4.10

Enter target: Quit

Total distance = 8.20

8. Test your program two times, with valid and invalid edges, and save the outputs.

What to Hand In

Create a MS-Word document and paste a copy of your code, and the sample runs from steps 6 and 8 to show that the program works. Your code will be graded on its functionality, appearance, and comments.

Extra Credit (3 points)

Update your code to allow the user to add additional nodes and edges to the graph. Test the program with at least one new node and two new edges. Save output from two runs, using these additions to the graph structure.

Create a MS-Word document and paste a copy of your code, and the sample runs. Your code will be graded on its functionality, appearance, and comments.