<Output>

Enter a file name: input\_a.txt

Enter two vertices (integer indexes): 0, 5

The number of vertices is 6

Vertex 0: (0, 1) (0, 2)

Vertex 1: (1, 0) (1, 3)

Vertex 2: (2, 0) (2, 3)

Vertex 3: (3, 1) (3, 2) (3, 4) (3, 5)

Vertex 4: (4, 2) (4, 3) (4, 5)

Vertex 5: (5, 3) (5, 4)

The path is 0 1 3 5

<End Output>

<Output>

Enter a file name: input\_b.txt

Enter two vertices (integer indexes): 3, 4

The number of vertices is 6

Vertex 0: (0, 1) (0, 2) (0, 3)

Vertex 1: (1, 0) (1, 3)

Vertex 2: (2, 0) (2, 3)

Vertex 3: (3, 0) (3, 1) (3, 2)

Vertex 4: (4, 5)

Vertex 5: (5, 4)

Graph not connected!

<End Output>

<Output>

Enter a file name: input\_c.txt

Enter two vertices (integer indexes): 2, 1

The number of vertices is 4

Vertex 0: (0, 1) (0, 2)

Vertex 1: (1, 0) (1, 3)

Vertex 2: (2, 0) (2, 3)

Vertex 3: (3, 1) (3, 2)

The path is 2 0 1

<End Output>

<Output>

Enter a file name: input\_d.txt

Enter two vertices (integer indexes): 3, 2

The number of vertices is 5

Vertex 0: (0, 1) (0, 2) (0, 3)

Vertex 1: (1, 0) (1, 3)

Vertex 2: (2, 0) (2, 3) (2, 4)

Vertex 3: (3, 0) (3, 1) (3, 2) (3, 4)

Vertex 4: (4, 2) (4, 3)

The path is 3 2

<End Output>

First it will take an input from the user on what file it is wanting to search for a connection between two vertices

Then it will loop through the verticies finding what all they are connected to and printing them out

Then it will look through it and pring out the shortest path to the two vertices that the user put in

Will print out what the path is for the shortest connection.