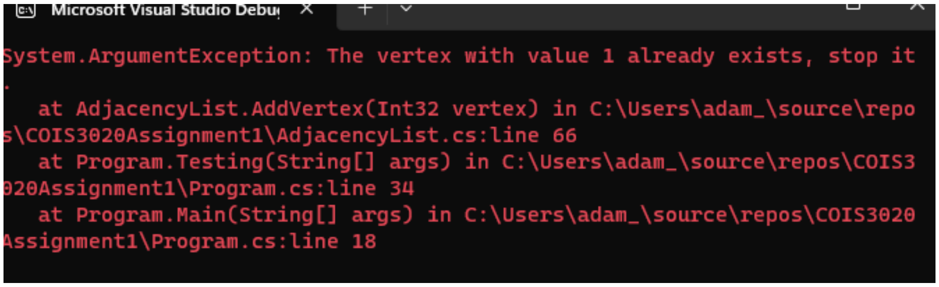
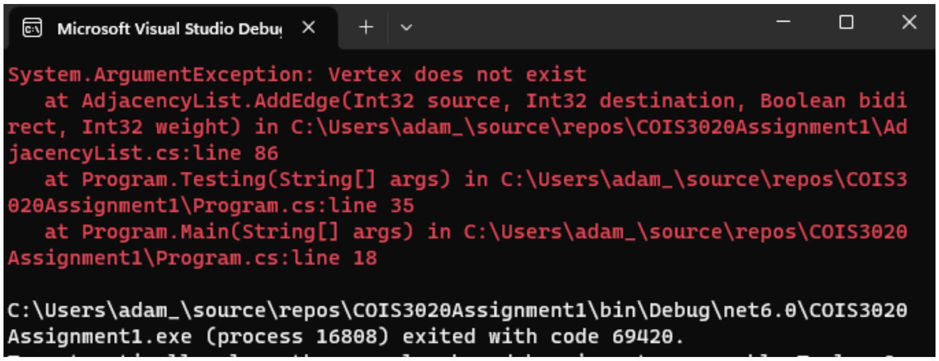
Testing Documentation

Test 1	
Description	Test control handling if attempting to input already existing vertex
Input	list.AddVertex(1); list.AddVertex(1);
Expected Output	Sorry, vertex already exists
Actual Output	 System.ArgumentException: The vertex with value 1 already exists, stop it . at AdjacencyList.AddVertex(Int32 vertex) in C:\Users\adam_\source\repo s\COIS3020Assignment1\AdjacencyList.cs:line 66 at Program.Testing(String[] args) in C:\Users\adam_\source\repos\COIS3 020Assignment1\Program.cs:line 34 at Program.Main(String[] args) in C:\Users\adam_\source\repos\COIS3020 Assignment1\Program.cs:line 18

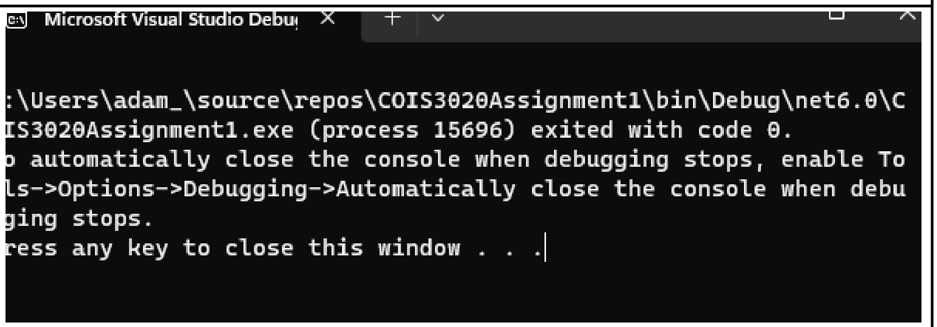
Test 2	
Description	Test error handling if attempting to add edge to non-existent vertex

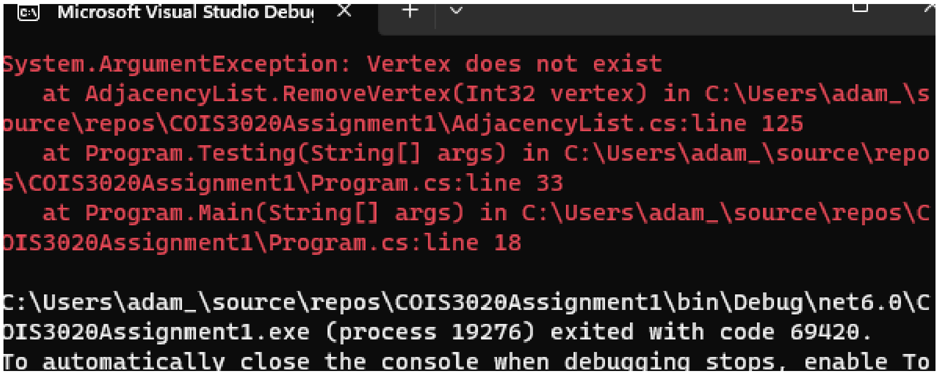
Input	list.AddVertex(1);
	list.AddVertex(2);
	list.AddEdge(1, 3, true, 1);

Expected Output	"Vertex does not exist"
Actual Output	 System.ArgumentException: Vertex does not exist at AdjacencyList.AddEdge(Int32 source, Int32 destination, Boolean bidi rect, Int32 weight) in C:\Users\adam_\source\repos\COIS3020Assignment1\Ad jacencyList.cs:line 86 at Program.Testing(String[] args) in C:\Users\adam_\source\repos\COIS3 020Assignment1\Program.cs:line 35 at Program.Main(String[] args) in C:\Users\adam_\source\repos\COIS3020 Assignment1\Program.cs:line 18 C:\Users\adam_\source\repos\COIS3020Assignment1\bin\Debug\net6.0\COIS3020 Assignment1.exe (process 16808) exited with code 69420.

Test 3	
Description	Test if implementation of remove Vertex successful

Input	list.AddVertex(1);
	list.RemoveVertex(1);
	list.AddVertex(1);

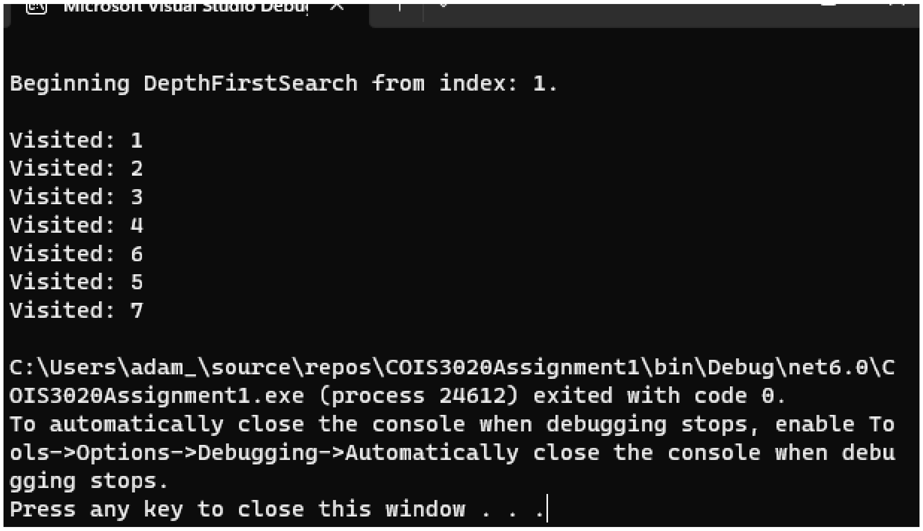
Expected Output	If successful, no output will occur otherwise error handling will occur
Actual Output	 :\Users\adam_\source\repos\COIS3020Assignment1\bin\Debug\net6.0\C IS3020Assignment1.exe (process 15696) exited with code 0. o automatically close the console when debugging stops, enable To ls->Options->Debugging->Automatically close the console when debu ging stops. ress any key to close this window . . .

Test 4	
Description	Test error handling when attempting to remove Vertice that does not exist
Input	list.RemoveVertex(1);
Expected Output	"Vertex does not exist" other if unsuccessful no output will occur
Actual Output	 System.ArgumentException: Vertex does not exist at AdjacencyList.RemoveVertex(Int32 vertex) in C:\Users\adam_\s ource\repos\COIS3020Assignment1\AdjacencyList.cs:line 125 at Program.Testing(String[] args) in C:\Users\adam_\source\repo s\COIS3020Assignment1\Program.cs:line 33 at Program.Main(String[] args) in C:\Users\adam_\source\repos\C OIS3020Assignment1\Program.cs:line 18 C:\Users\adam_\source\repos\COIS3020Assignment1\bin\Debug\net6.0\C OIS3020Assignment1.exe (process 19276) exited with code 69420. To automatically close the console when debugging stops, enable To

Test 4	

Test 5	
Description	Testing DFS traversal from a specific vertex.

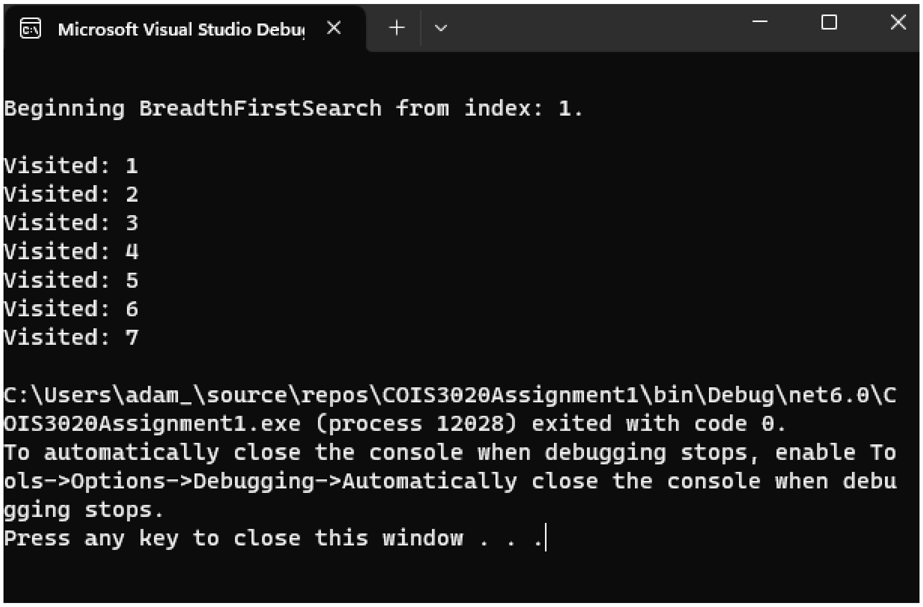
Input	Starting vertex = 1
	list.AddVertex(1);
	list.AddVertex(2);
	list.AddVertex(3);
	list.AddVertex(4);
	list.AddVertex(5);
	list.AddVertex(6);
	list.AddVertex(7);
	list.AddEdge(1, 2, true, 2);
	list.AddEdge(1, 3, true, 3);
	list.AddEdge(1, 4, true, 3);
	list.AddEdge(2, 3, true, 4);
	list.AddEdge(2, 5, true, 3);
	list.AddEdge(3, 4, true, 5);
	list.AddEdge(3, 5, true, 1);
	list.AddEdge(4, 6, true, 7);
	list.AddEdge(5, 6, true, 8);
	list.AddEdge(6, 7, true, 9);

	list.DepthFirstSearch(1);
Expected Output	Beginning DepthFirstSearch from index: 1. Visited: 1 Visited: 2 Visited: 3 Visited: 4 Visited: 6 Visited: 5 Visited: 7
Actual Output	 Beginning DepthFirstSearch from index: 1. Visited: 1 Visited: 2 Visited: 3 Visited: 4 Visited: 6 Visited: 5 Visited: 7 C:\Users\adam_\source\repos\COIS3020Assignment1\bin\Debug\net6.0\C OIS3020Assignment1.exe (process 24612) exited with code 0. To automatically close the console when debugging stops, enable To ols->Options->Debugging->Automatically close the console when debu gging stops. Press any key to close this window . . .

Test 6	
Description	Testing BFS traversal from a specific vertex.
Input	Starting vertex = 1

Expected Output	list.AddVertex(1);
	list.AddVertex(2);
	list.AddVertex(3);
	list.AddVertex(4);
	list.AddVertex(5);
	list.AddVertex(6);

	list.AddVertex(7);
	list.AddEdge(1, 2, true, 2);
	list.AddEdge(1, 3, true, 3);
	list.AddEdge(1, 4, true, 3);
	list.AddEdge(2, 3, true, 4);
	list.AddEdge(2, 5, true, 3);
	list.AddEdge(3, 4, true, 5);
	list.AddEdge(3, 5, true, 1);
	list.AddEdge(4, 6, true, 7);
	list.AddEdge(5, 6, true, 8);
	list.AddEdge(6, 7, true, 9);
	Console.WriteLine(list.BreadthFirstSearch(1));

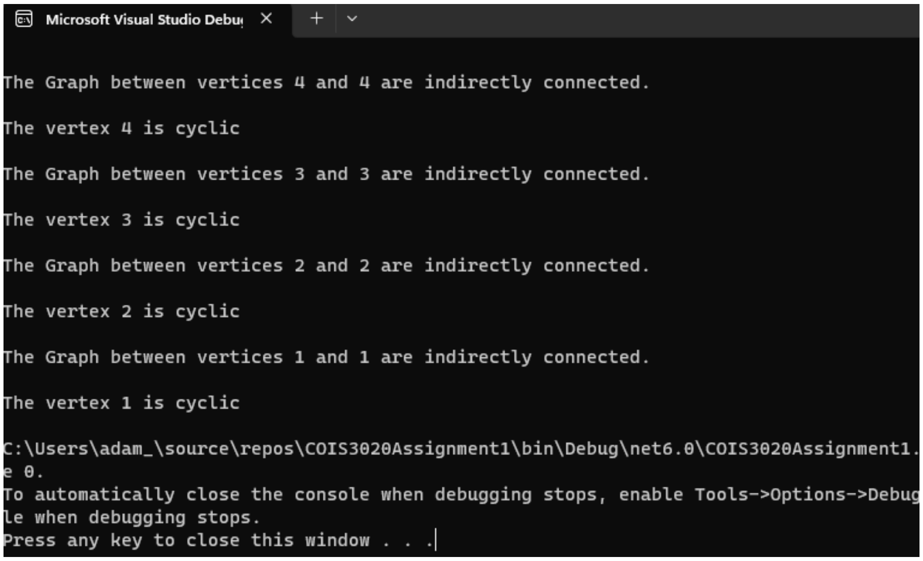
Actual Output	 Beginning BreadthFirstSearch from index: 1. Visited: 1 Visited: 2 Visited: 3 Visited: 4 Visited: 5 Visited: 6 Visited: 7 C:\Users\adam_\source\repos\COIS3020Assignment1\bin\Debug\net6.0\COIS3020Assignment1.exe (process 12028) exited with code 0. To automatically close the console when debugging stops, enable Tools->Options->Debugging->Automatically close the console when debugging stops. Press any key to close this window . . .

Test 7	
Description	Test feedback on CycleDetection method
Input	list.AddVertex(1);

	list.AddVertex(2);
	list.AddVertex(3);
	list.AddVertex(4);
	list.AddVertex(5);
	list.AddVertex(6);
	list.AddVertex(7);
	list.AddEdge(1, 2, true, 2);
	list.AddEdge(1, 3, true, 3);
	list.AddEdge(1, 4, true, 3);
	list.AddEdge(2, 3, true, 4);
	list.AddEdge(2, 5, true, 3);
	list.AddEdge(3, 4, true, 5);
	list.AddEdge(3, 5, true, 1);
	list.AddEdge(4, 6, true, 7);
	list.AddEdge(5, 6, true, 8);
	list.AddEdge(6, 7, true, 9);
	list.CycleDetect(4);
	list.CycleDetect(3);
	list.CycleDetect(2);
	list.CycleDetect(1);

Expected Output	The vertex 4 is cyclic The vertex 3 is cyclic The vertex 2 is cyclic The vertex 1 is cyclic

<table>
<tr><td><b>Actual Output</b></td><td>



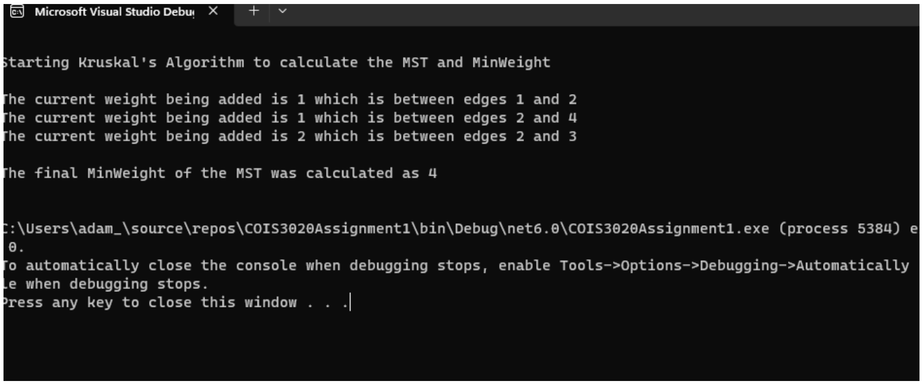
```
The Graph between vertices 4 and 4 are indirectly connected.

The vertex 4 is cyclic

The Graph between vertices 3 and 3 are indirectly connected.

The vertex 3 is cyclic

The Graph between vertices 2 and 2 are indirectly connected.

The vertex 2 is cyclic

The Graph between vertices 1 and 1 are indirectly connected.

The vertex 1 is cyclic

C:\Users\adam_\source\repos\COIS3020Assignment1\bin\Debug\net6.0\COIS3020Assignment1.
e 0.
To automatically close the console when debugging stops, enable Tools->Options->Debug
le when debugging stops.
Press any key to close this window . . .
```

</td></tr>
</table>

Test 8	
Description	Test feedback on MST function with implementation of Kruskal’s algorithm

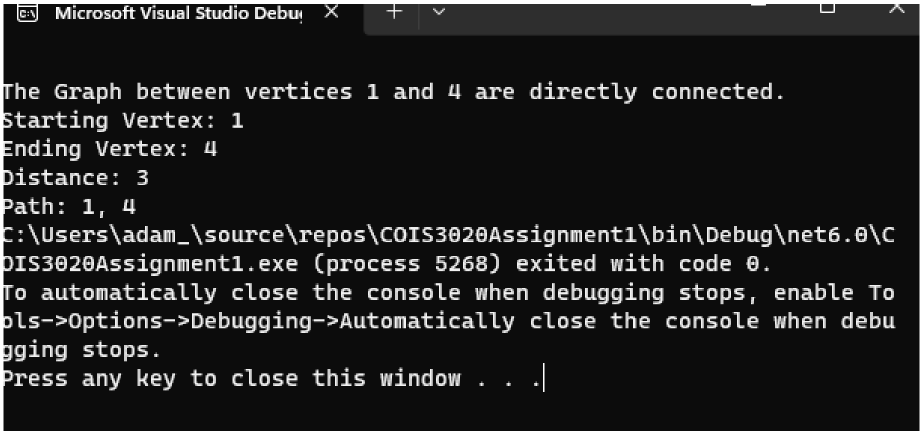
Input	list2.AddVertex(1);
	list2.AddVertex(2);
	list2.AddVertex(3);
	list2.AddVertex(4);
	list2.AddEdge(1, 2, true, 1);
	list2.AddEdge(2, 4, true, 1);

	list2.AddEdge(2, 3, true, 2);
	list2.MST();

Expected Output	Starting Kruskal's Algorithm to calculate the MST and MinWeight The current weight being added is 1 which is between edges 1 and 2 The current weight being added is 1 which is between edges 2 and 4 The current weight being added is 2 which is between edges 2 and 3 The final MinWeight of the MST was calculated as 4
Actual Output	 Starting Kruskal's Algorithm to calculate the MST and MinWeight The current weight being added is 1 which is between edges 1 and 2 The current weight being added is 1 which is between edges 2 and 4 The current weight being added is 2 which is between edges 2 and 3 The final MinWeight of the MST was calculated as 4 C:\Users\adam_\source\repos\COIS3020Assignment1\bin\Debug\net6.0\COIS3020Assignment1.exe (process 5384) e 0. To automatically close the console when debugging stops, enable Tools->Options->Debugging->Automatically le when debugging stops. Press any key to close this window . . .

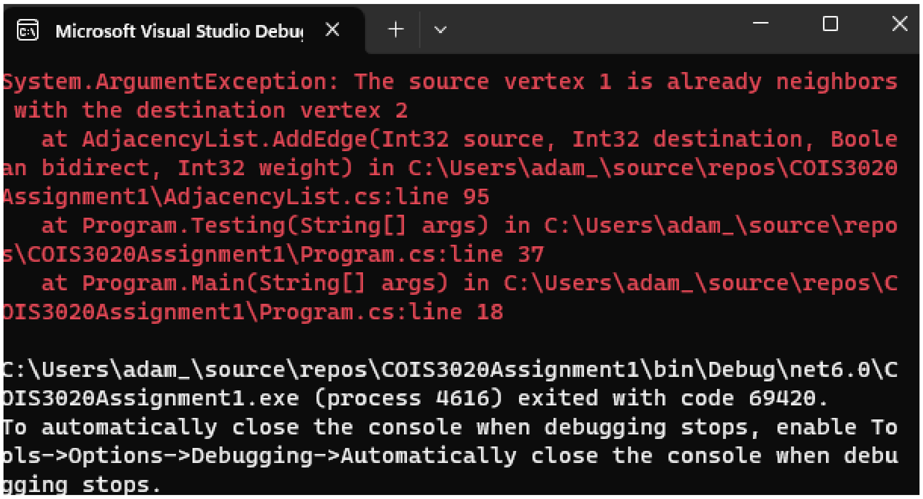
Test 9	
Description	Feedback on ShortestPath method to determine shortest path that is predetermined
Input	list.AddVertex(1);

	list.AddVertex(2);
	list.AddVertex(3);
	list.AddVertex(4);
	list.AddEdge(1, 2, true, 2);
	list.AddEdge(1, 4, true, 3);
	list.AddEdge(2, 4, true, 2);
	Console.Write(list.ShortPath(1, 4));

Expected Output	Since 1 and 4 are directly connected, it should output as such and list their distance The Graph between vertices 1 and 4 are directly connected. Starting Vertex: 1 Ending Vertex: 4 Distance: 3 Path: 1, 4
Actual Output	 The Graph between vertices 1 and 4 are directly connected. Starting Vertex: 1 Ending Vertex: 4 Distance: 3 Path: 1, 4 C:\Users\adam_\source\repos\COIS3020Assignment1\bin\Debug\net6.0\C OIS3020Assignment1.exe (process 5268) exited with code 0. To automatically close the console when debugging stops, enable To ols->Options->Debugging->Automatically close the console when debu gging stops. Press any key to close this window . . .

Test 10	
Description	Test feedback on attempting to add edge where edge already exists between pair of vertices

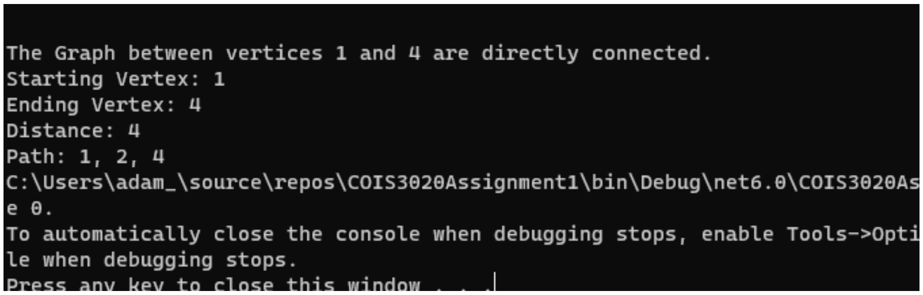
Input	list.AddVertex(1);
	list.AddVertex(2);
	list.AddEdge(1, 2, true, 3);
	list.AddEdge(1, 2, true, 3);

Expected Output	The source vertex 1 is already neighbors with the destination vertex 2
Actual Output	 System.ArgumentException: The source vertex 1 is already neighbors with the destination vertex 2 at AdjacencyList.AddEdge(Int32 source, Int32 destination, Boole an bidirect, Int32 weight) in C:\Users\adam_\source\repos\COIS3020 Assignment1\AdjacencyList.cs:line 95 at Program.Testing(String[] args) in C:\Users\adam_\source\repo s\COIS3020Assignment1\Program.cs:line 37 at Program.Main(String[] args) in C:\Users\adam_\source\repos\C OIS3020Assignment1\Program.cs:line 18 C:\Users\adam_\source\repos\COIS3020Assignment1\bin\Debug\net6.0\C OIS3020Assignment1.exe (process 4616) exited with code 69420. To automatically close the console when debugging stops, enable To ols->Options->Debugging->Automatically close the console when debu gging stops.

Test 11

Description	Feedback on ShortestPath method to determine shortest path that is predetermined, where direct path has weight greater than indirect path

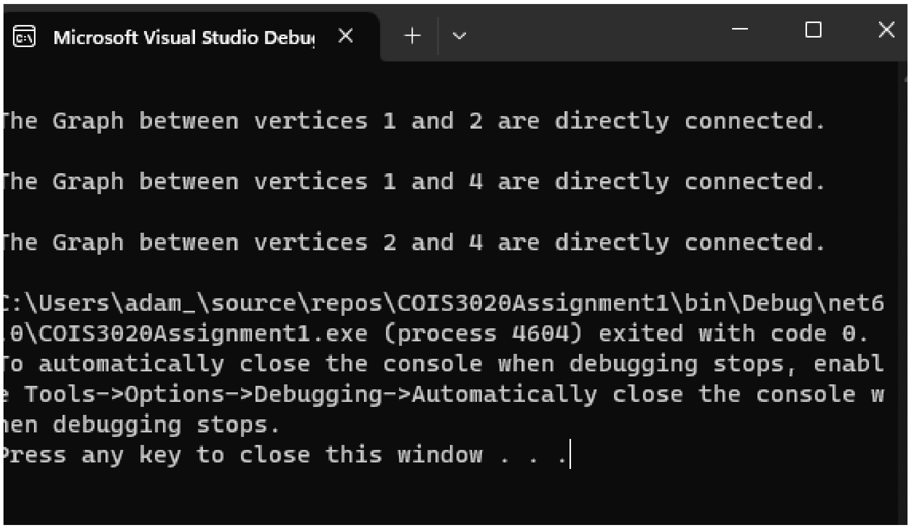
Input	list.AddVertex(1);
	list.AddVertex(2);
	list.AddVertex(3);
	list.AddVertex(4);
	list.AddEdge(1, 2, true, 2);
	list.AddEdge(1, 4, true, 6);
	list.AddEdge(2, 4, true, 2);
	Console.Write(list.ShortPath(1, 4));

Expected Output	Since 1 and 4 are directly connected, but their distance outweighs that of the indirect path so it should output the shortest path as 1,2,4 The Graph between vertices 1 and 4 are directly connected. Starting Vertex: 1 Ending Vertex: 4 Distance: 4 Path: 1, 2, 4
Actual Output	 The Graph between vertices 1 and 4 are directly connected. Starting Vertex: 1 Ending Vertex: 4 Distance: 4 Path: 1, 2, 4 C:\Users\adam_\source\repos\COIS3020Assignment1\bin\Debug\net6.0\COIS3020As e 0. To automatically close the console when debugging stops, enable Tools->Opti le when debugging stops. Press any key to close this window . . .

Test 12	
Description	Determine feedback on graph connectivity function where graph is connected

Input	list.AddVertex(1);
	list.AddVertex(2);
	list.AddVertex(3);
	list.AddVertex(4);
	list.AddEdge(1, 2, true, 2);
	list.AddEdge(1, 4, true, 3);
	list.AddEdge(2, 4, true, 2);
	
	list.VerConnect(1, 2);
	list.VerConnect(1, 4);
	list.VerConnect(2, 4);

Expected Output	The Graph between vertices 1 and 2 are directly connected. The Graph between vertices 1 and 4 are directly connected. The Graph between vertices 2 and 4 are directly connected.

Actual Output	 The Graph between vertices 1 and 2 are directly connected. The Graph between vertices 1 and 4 are directly connected. The Graph between vertices 2 and 4 are directly connected. C:\Users\adam_\source\repos\COIS3020Assignment1\bin\Debug\net6.0\COIS3020Assignment1.exe (process 4604) exited with code 0. To automatically close the console when debugging stops, enable Tools->Options->Debugging->Automatically close the console when debugging stops. Press any key to close this window . . .

Test 13	
Description	Determine feedback on graph connectivity function where no connections exist

Input	list.AddVertex(1);
	list.AddVertex(2);
	list.AddVertex(3);
	list.AddVertex(4);
	list.AddEdge(1, 2, true, 2);
	list.AddEdge(1, 4, true, 3);
	list.AddEdge(2, 4, true, 2);
	Console.Write(list.ShortPath(1, 4));

Expected Output	list.AddVertex(1);
	list.AddVertex(2);
	list.AddVertex(3);
	list.AddVertex(4);

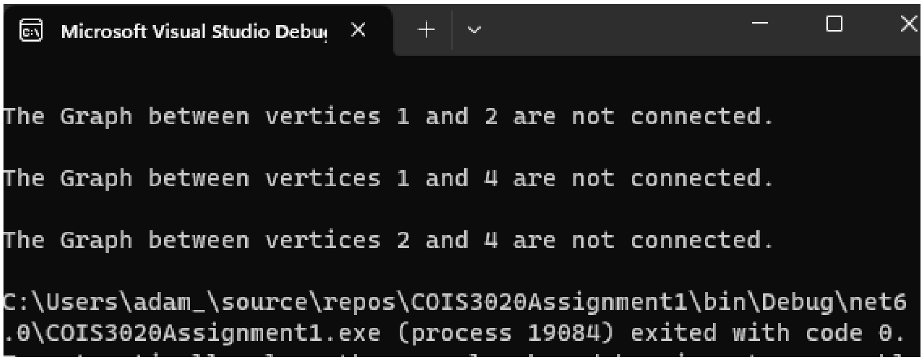
<table>
<tr>
<td></td>
<td>

list.VerConnect(1, 2);
list.VerConnect(1, 4);
list.VerConnect(2, 4);

</td>
</tr>
<tr>
<td>

Actual Output

</td>
<td>



```
The Graph between vertices 1 and 2 are not connected.

The Graph between vertices 1 and 4 are not connected.

The Graph between vertices 2 and 4 are not connected.

C:\Users\adam_\source\repos\COIS3020Assignment1\bin\Debug\net6
.0\COIS3020Assignment1.exe (process 19084) exited with code 0.
```

</td>
</tr>
</table>