a.

Yes, the program can be compiled and executed successfully and should run as expected.

b.

Run program in preferred IDE and respond to all inputs as appropriate, program still has remnants of Project 0 and 1 so just answer the prompts for them regardless.

c.

Amado Rodriguez and Shaun Froseth wrote the majority of the code.

Nelly Sanchez-Cruz wrote most of this readme file and ran test cases to ensure the code would run as expected.

All three members contributed to the brainstorming process that allowed us to complete the

assignment.

```
C:\Users\odama\.jdks\openjdk-15.0.2\bin\java.exe "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA 2020.2.4\lib\idea_rt.jar=62967:C:\Program Files\JetBrains\IntelliJ IDEA 2020.2.4\bin" -Dfile.encoding=UTF-8 -classpath "C:\Users\odama\Documents\School\CSC 401\Homework\Sensor Network\out\production\Sensor Network" SensorNetwork
Enter the width (in meters):
10
Enter the height (in meters):
10
Enter the number of nodes:
10
Enter the Transmission range (in meters):
7
Enter the number of data nodes:
5
Enter the number of data packets each data node has:
1
Enter the storage capacity of each storage node:
2
Enter the graph traversal technique:
Recursive DFS: 0
DFS Using Stack: 1
BFS Using Queue: 2
0

Node List:
Node:1, xAxis: 7.7, yAxis: 6.0
Node:2, xAxis: 8.6, yAxis: 6.6
Node:3, xAxis: 2.6, yAxis: 2.2
Node:4, xAxis: 0.6, yAxis: 6.7
Node:5, xAxis: 2.1, yAxis: 0.5
Node:6, xAxis: 1.3, yAxis: 4.6
Node:7, xAxis: 2.7, yAxis: 9.1
Node:8, xAxis: 7.2, yAxis: 9.6
Node:9, xAxis: 1.2, yAxis: 1.7
Node:10, xAxis: 1.8, yAxis: 4.3

Executing Recursive DFS Algorithm
The network is fully connected with one connected component.
[10, 9, 6, 5, 4, 3, 8, 7, 2, 1]
Data Nodes:
```

```
1
2
3
5
9
Storage Nodes:
4
6
7
8
10
Please input the ID of a DN:
2
Please input the ID of a target SN:
5
Enter the shortest path algorithm desired:
Dijkstra's shortest path: 0
Bellman-Ford dynamic programming: 1
Shortest path between them with k edges: 2
2
Please enter the number of edges k:
1
There is no path with 1 edges.
```

```
C:\Users\odama\.jdks\openjdk-15.0.2\bin\java.exe "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA 2020.2.4\lib\idea_rt.jar=63038:C:\Program Files\JetBrains\IntelliJ IDEA 2020.2.4\bin" -Dfile.encoding=UTF-8 -classpath "C:\Users\odama\Documents\School\CSC 401\Homework\Sensor Network\out\production\Sensor Network" SensorNetwork
Enter the width (in meters):
100
Enter the height (in meters):
100
Enter the number of nodes:
100
Enter the Transmission range (in meters):
20
Enter the number of data nodes:
20
Enter the number of data packets each data node has:
3
Enter the storage capacity of each storage node:
1
Enter the graph traversal technique:
Recursive DFS: 0
DFS Using Stack: 1
BFS Using Queue: 2
0

Node List:
Node:1, xAxis: 92.3, yAxis: 86.1
Node:2, xAxis: 45.8, yAxis: 80.5
Node:3, xAxis: 16.1, yAxis: 61.7
Node:4, xAxis: 44.5, yAxis: 57.9
Node:5, xAxis: 88.7, yAxis: 96.5
Node:6, xAxis: 23.3, yAxis: 6.8
Node:7, xAxis: 18.2, yAxis: 62.3
Node:8, xAxis: 40.6, yAxis: 59.2
Node:9, xAxis: 33.2, yAxis: 58.7
Node:10, xAxis: 16.4, yAxis: 14.8
Node:11, xAxis: 75.2, yAxis: 6.3
Node:12, xAxis: 23.0, yAxis: 83.2
Node:13, xAxis: 31.9, yAxis: 25.0
Node:14, xAxis: 85.2, yAxis: 85.6
Node:15, xAxis: 94.5, yAxis: 16.3
Node:16, xAxis: 22.3, yAxis: 20.0
```

```
Node:17, xAxis: 45.2, yAxis: 88.1
Node:18, xAxis: 43.9, yAxis: 28.0
Node:19, xAxis: 31.0, yAxis: 35.8
Node:20, xAxis: 60.9, yAxis: 40.3
Node:21, xAxis: 64.2, yAxis: 20.5
Node:22, xAxis: 6.9, yAxis: 18.3
Node:23, xAxis: 92.9, yAxis: 65.1
Node:24, xAxis: 73.1, yAxis: 55.6
Node:25, xAxis: 22.9, yAxis: 24.5
Node:26, xAxis: 48.4, yAxis: 49.7
Node:27, xAxis: 39.2, yAxis: 70.7
Node:28, xAxis: 21.4, yAxis: 4.3
Node:29, xAxis: 61.8, yAxis: 74.8
Node:30, xAxis: 46.3, yAxis: 79.9
Node:31, xAxis: 96.2, yAxis: 61.8
Node:32, xAxis: 93.3, yAxis: 62.9
Node:33, xAxis: 85.5, yAxis: 58.1
Node:34, xAxis: 9.3, yAxis: 87.9
Node:35, xAxis: 46.6, yAxis: 52.7
Node:36, xAxis: 88.2, yAxis: 8.5
Node:37, xAxis: 79.6, yAxis: 44.5
Node:38, xAxis: 8.4, yAxis: 79.6
Node:39, xAxis: 71.5, yAxis: 55.5
Node:40, xAxis: 30.4, yAxis: 83.5
Node:41, xAxis: 76.8, yAxis: 44.5
Node:42, xAxis: 59.3, yAxis: 4.9
Node:43, xAxis: 45.9, yAxis: 72.8
Node:44, xAxis: 65.4, yAxis: 36.2
Node:45, xAxis: 90.3, yAxis: 34.5
Node:46, xAxis: 30.3, yAxis: 72.5
Node:47, xAxis: 19.2, yAxis: 83.1
Node:48, xAxis: 27.0, yAxis: 39.2
Node:49, xAxis: 54.0, yAxis: 58.2
Node:50, xAxis: 71.1, yAxis: 78.3
Node:51, xAxis: 31.9, yAxis: 39.7
Node:52, xAxis: 83.6, yAxis: 10.8
Node:53, xAxis: 35.7, yAxis: 33.8
Node:54, xAxis: 84.4, yAxis: 81.7
Node:55, xAxis: 48.3, yAxis: 26.6
Node:56, xAxis: 98.2, yAxis: 88.8
Node:57, xAxis: 14.7, yAxis: 25.0
Node:58, xAxis: 89.9, yAxis: 96.3
Node:59, xAxis: 8.1, yAxis: 39.5
Node:60, xAxis: 63.2, yAxis: 33.0
```

```
Node:61, xAxis: 40.2, yAxis: 56.6
Node:62, xAxis: 50.4, yAxis: 79.7
Node:63, xAxis: 26.1, yAxis: 87.9
Node:64, xAxis: 92.5, yAxis: 57.2
Node:65, xAxis: 10.5, yAxis: 82.3
Node:66, xAxis: 58.8, yAxis: 37.9
Node:67, xAxis: 92.8, yAxis: 36.0
Node:68, xAxis: 84.0, yAxis: 14.5
Node:69, xAxis: 39.1, yAxis: 35.3
Node:70, xAxis: 7.6, yAxis: 79.9
Node:71, xAxis: 54.6, yAxis: 27.5
Node:72, xAxis: 83.1, yAxis: 87.6
Node:73, xAxis: 0.2, yAxis: 18.6
Node:74, xAxis: 88.7, yAxis: 87.8
Node:75, xAxis: 77.0, yAxis: 62.5
Node:76, xAxis: 11.3, yAxis: 59.7
Node:77, xAxis: 67.6, yAxis: 17.8
Node:78, xAxis: 95.1, yAxis: 87.7
Node:79, xAxis: 20.7, yAxis: 21.3
Node:80, xAxis: 41.2, yAxis: 64.7
Node:81, xAxis: 89.4, yAxis: 40.9
Node:82, xAxis: 22.5, yAxis: 36.6
Node:83, xAxis: 81.2, yAxis: 54.3
Node:84, xAxis: 29.2, yAxis: 44.6
Node:85, xAxis: 64.0, yAxis: 18.5
Node:86, xAxis: 92.2, yAxis: 61.4
Node:87, xAxis: 73.5, yAxis: 81.9
Node:88, xAxis: 41.3, yAxis: 61.5
Node:89, xAxis: 8.7, yAxis: 20.9
Node:90, xAxis: 43.4, yAxis: 14.6
Node:91, xAxis: 63.6, yAxis: 72.7
Node:92, xAxis: 29.8, yAxis: 42.7
Node:93, xAxis: 36.0, yAxis: 72.1
Node:94, xAxis: 67.0, yAxis: 32.8
Node:95, xAxis: 75.6, yAxis: 96.8
Node:96, xAxis: 47.8, yAxis: 57.1
Node:97, xAxis: 80.2, yAxis: 70.0
Node:98, xAxis: 77.2, yAxis: 11.9
Node:99, xAxis: 28.4, yAxis: 69.2
Node:100, xAxis: 9.6, yAxis: 48.5

Executing Recursive DFS Algorithm
The network is fully connected with one connected component.
```

```
[78, 95, 14, 74, 58, 56, 72, 31, 23, 86, 24, 90, 94, 60, 77, 42, 85, 11, 45, 15, 68, 52, 36, 98, 21, 55, 61, 62, 29, 30, 27, 93, 63, 47, 70, 12, 46, 88, 26, 92, 53, 84, 51, 59, 89, 73, 22, 57, 82, 28, 79, 25, 6, 16, 10, 13, 48, 76, 100, 7, 9, 8, 43, 91, 87, 50, 75, 41, 20, 44, 71, 69, 19, 18, 66, 35, 4, 99, 3, 38, 34, 65, 40, 17, 2, 80, 96, 49, 39, 83, 81, 67, 37, 33, 64, 32, 97, 54, 5, 1]
Data Nodes:
32
99
68
4
70
6
72
73
41
80
49
18
50
83
20
53
86
88
57
61
Storage Nodes:
1
2
3
5
7
8
9
10
11
12
13
14
15
16
```

```
163 17
164 19
165 21
166 22
167 23
168 24
169 25
170 26
171 27
172 28
173 29
174 30
175 31
176 33
177 34
178 35
179 36
180 37
181 38
182 39
183 40
184 42
185 43
186 44
187 45
188 46
189 47
190 48
191 51
192 52
193 54
194 55
195 56
196 58
197 59
198 60
199 62
200 63
201 64
202 65
203 66
204 67
205 69
206 71
```

```
74
75
76
77
78
79
81
82
84
85
87
89
90
91
92
93
94
95
96
97
98
100
Please input the ID of a DN:
4
Please input the ID of a target SN:
3
Enter the shortest path algorithm desired:
Dijkstra's shortest path: 0
Bellman-Ford dynamic programming: 1
Shortest path between them with k edges: 2
2
Please enter the number of edges k:
10
The cost of offloading one data packet from DN# 4 to SN# 3 using 10 edges is:
0.0012110041476543687 Joules.
The minimum energy cost of offloading all data packets from this node is:
0.003633012442963106 Joules.

```

```
C:\Users\odama\.jdks\openjdk-15.0.2\bin\java.exe "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA 2020.2.4\lib\idea_rt.jar=63288:C:\Program Files\JetBrains\IntelliJ IDEA 2020.2.4\bin" -Dfile.encoding=UTF-8 -classpath "C:\Users\odama\Documents\School\CSC 401\Homework\Sensor Network\out\production\Sensor Network" SensorNetwork
Enter the width (in meters):
100
Enter the height (in meters):
100
Enter the number of nodes:
100
Enter the Transmission range (in meters):
20
Enter the number of data nodes:
20
Enter the number of data packets each data node has:
2
Enter the storage capacity of each storage node:
2
Enter the graph traversal technique:
Recursive DFS: 0
DFS Using Stack: 1
BFS Using Queue: 2
0

Node List:
Node:1, xAxis: 24.2, yAxis: 17.7
Node:2, xAxis: 43.4, yAxis: 18.3
Node:3, xAxis: 17.3, yAxis: 74.9
Node:4, xAxis: 89.3, yAxis: 81.5
Node:5, xAxis: 29.3, yAxis: 40.7
Node:6, xAxis: 26.8, yAxis: 88.4
Node:7, xAxis: 46.8, yAxis: 5.0
Node:8, xAxis: 77.4, yAxis: 92.7
Node:9, xAxis: 18.5, yAxis: 31.4
Node:10, xAxis: 73.9, yAxis: 29.3
Node:11, xAxis: 33.3, yAxis: 29.7
Node:12, xAxis: 68.4, yAxis: 59.2
Node:13, xAxis: 24.2, yAxis: 87.4
Node:14, xAxis: 90.0, yAxis: 91.3
Node:15, xAxis: 65.9, yAxis: 71.3
Node:16, xAxis: 60.6, yAxis: 84.6
```

```
Node:17, xAxis: 20.3, yAxis: 30.8
Node:18, xAxis: 86.0, yAxis: 39.2
Node:19, xAxis: 43.8, yAxis: 55.5
Node:20, xAxis: 80.7, yAxis: 49.8
Node:21, xAxis: 14.4, yAxis: 67.5
Node:22, xAxis: 50.1, yAxis: 8.9
Node:23, xAxis: 52.5, yAxis: 56.3
Node:24, xAxis: 50.9, yAxis: 14.7
Node:25, xAxis: 7.7, yAxis: 96.4
Node:26, xAxis: 73.2, yAxis: 60.4
Node:27, xAxis: 59.2, yAxis: 25.2
Node:28, xAxis: 39.4, yAxis: 55.7
Node:29, xAxis: 30.1, yAxis: 92.9
Node:30, xAxis: 95.1, yAxis: 59.5
Node:31, xAxis: 47.2, yAxis: 42.0
Node:32, xAxis: 74.6, yAxis: 81.1
Node:33, xAxis: 87.1, yAxis: 17.0
Node:34, xAxis: 61.3, yAxis: 44.7
Node:35, xAxis: 39.1, yAxis: 49.6
Node:36, xAxis: 23.9, yAxis: 94.5
Node:37, xAxis: 77.0, yAxis: 22.7
Node:38, xAxis: 6.9, yAxis: 61.0
Node:39, xAxis: 80.3, yAxis: 61.7
Node:40, xAxis: 67.9, yAxis: 0.7
Node:41, xAxis: 64.1, yAxis: 67.1
Node:42, xAxis: 9.2, yAxis: 67.2
Node:43, xAxis: 78.6, yAxis: 55.6
Node:44, xAxis: 22.5, yAxis: 72.4
Node:45, xAxis: 16.7, yAxis: 46.2
Node:46, xAxis: 74.0, yAxis: 48.1
Node:47, xAxis: 24.8, yAxis: 88.7
Node:48, xAxis: 87.6, yAxis: 22.5
Node:49, xAxis: 46.0, yAxis: 7.3
Node:50, xAxis: 47.2, yAxis: 89.0
Node:51, xAxis: 39.2, yAxis: 0.0
Node:52, xAxis: 25.5, yAxis: 50.4
Node:53, xAxis: 4.1, yAxis: 12.8
Node:54, xAxis: 67.7, yAxis: 92.8
Node:55, xAxis: 17.6, yAxis: 86.7
Node:56, xAxis: 98.7, yAxis: 57.1
Node:57, xAxis: 26.8, yAxis: 56.7
Node:58, xAxis: 2.6, yAxis: 48.3
Node:59, xAxis: 74.0, yAxis: 67.3
Node:60, xAxis: 18.6, yAxis: 49.0
```

```
Node:61, xAxis: 31.2, yAxis: 58.9
Node:62, xAxis: 30.7, yAxis: 78.7
Node:63, xAxis: 0.9, yAxis: 68.9
Node:64, xAxis: 86.4, yAxis: 71.6
Node:65, xAxis: 2.2, yAxis: 87.2
Node:66, xAxis: 64.6, yAxis: 30.0
Node:67, xAxis: 97.0, yAxis: 3.2
Node:68, xAxis: 26.6, yAxis: 46.4
Node:69, xAxis: 27.8, yAxis: 82.8
Node:70, xAxis: 29.1, yAxis: 78.3
Node:71, xAxis: 31.7, yAxis: 16.9
Node:72, xAxis: 44.3, yAxis: 1.2
Node:73, xAxis: 68.0, yAxis: 90.3
Node:74, xAxis: 35.3, yAxis: 63.9
Node:75, xAxis: 46.0, yAxis: 31.7
Node:76, xAxis: 57.8, yAxis: 16.0
Node:77, xAxis: 68.1, yAxis: 16.0
Node:78, xAxis: 16.9, yAxis: 84.4
Node:79, xAxis: 10.9, yAxis: 9.7
Node:80, xAxis: 87.1, yAxis: 26.5
Node:81, xAxis: 48.8, yAxis: 66.7
Node:82, xAxis: 20.2, yAxis: 1.3
Node:83, xAxis: 90.4, yAxis: 8.2
Node:84, xAxis: 98.7, yAxis: 55.7
Node:85, xAxis: 35.4, yAxis: 42.6
Node:86, xAxis: 63.4, yAxis: 44.8
Node:87, xAxis: 82.2, yAxis: 12.3
Node:88, xAxis: 39.4, yAxis: 67.0
Node:89, xAxis: 9.2, yAxis: 87.0
Node:90, xAxis: 5.8, yAxis: 20.2
Node:91, xAxis: 32.6, yAxis: 56.4
Node:92, xAxis: 8.5, yAxis: 58.0
Node:93, xAxis: 91.7, yAxis: 15.2
Node:94, xAxis: 94.0, yAxis: 50.5
Node:95, xAxis: 48.7, yAxis: 74.3
Node:96, xAxis: 49.8, yAxis: 40.0
Node:97, xAxis: 96.0, yAxis: 76.0
Node:98, xAxis: 74.9, yAxis: 9.1
Node:99, xAxis: 53.9, yAxis: 40.2
Node:100, xAxis: 65.2, yAxis: 70.8

Executing Recursive DFS Algorithm
The network is fully connected with one connected component.
```

```
[79, 90, 53, 82, 72, 77, 25, 94, 30, 84, 56, 14, 97, 4, 8, 54, 73, 95, 15, 12, 59, 43, 26, 41, 61, 28, 63, 92, 57, 52, 45, 91, 23, 100, 81, 88, 62, 47, 78, 44, 13, 29, 50, 16, 32, 64, 39, 46, 10, 18, 20, 86, 31, 11, 85, 75, 76, 40, 93, 87, 67, 83, 98, 33, 80, 48, 37, 27, 24, 7, 22, 51, 2, 49, 71, 9, 5, 60, 58, 38, 21, 42, 89, 6, 69, 36, 55, 65, 3, 70, 74, 19, 99, 66, 34, 96, 35, 68, 17, 1]
Data Nodes:
67
68
69
6
39
72
10
11
12
78
47
16
56
89
25
58
59
28
30
95
Storage Nodes:
1
2
3
4
5
7
8
9
13
14
15
17
18
19
```

```
163 20
164 21
165 22
166 23
167 24
168 26
169 27
170 29
171 31
172 32
173 33
174 34
175 35
176 36
177 37
178 38
179 40
180 41
181 42
182 43
183 44
184 45
185 46
186 48
187 49
188 50
189 51
190 52
191 53
192 54
193 55
194 57
195 60
196 61
197 62
198 63
199 64
200 65
201 66
202 70
203 71
204 73
205 74
206 75
```

```
76
77
79
80
81
82
83
84
85
86
87
88
90
91
92
93
94
96
97
98
99
100
Please input the ID of a DN:
11
Please input the ID of a target SN:
65
Enter the shortest path algorithm desired:
Dijkstra's shortest path: 0
Bellman-Ford dynamic programming: 1
Shortest path between them with k edges: 2
1
The minimum energy cost of offloading one data packet from DN# 11 to SN# 65 is:
0.0032313093701760542 Joules.
The minimum energy cost of offloading all data packets from this node is:
0.006462618740352108 Joules.
Using this path:
65 -> 78 -> 70 -> 74 -> 91 -> 85 -> 11
```