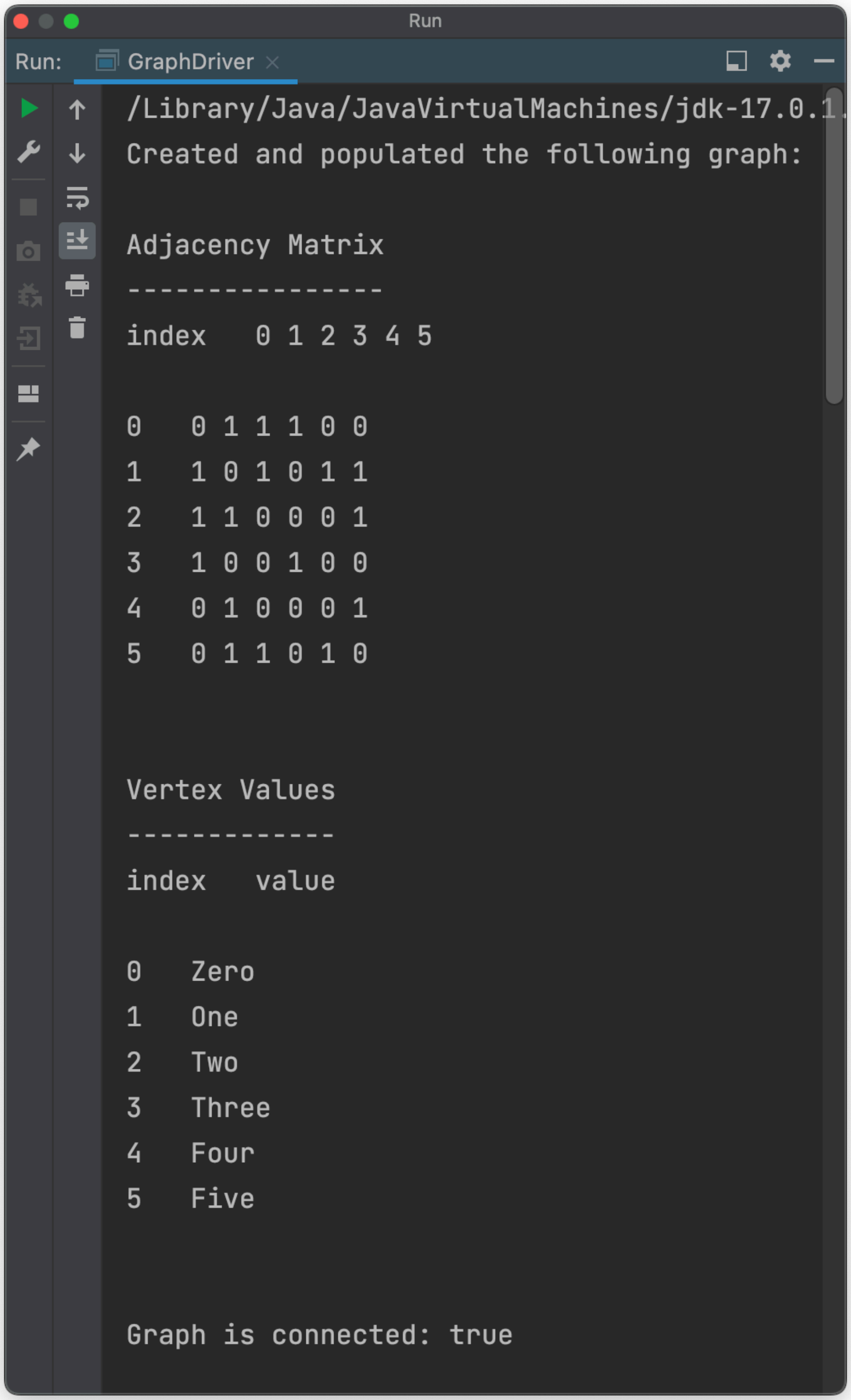
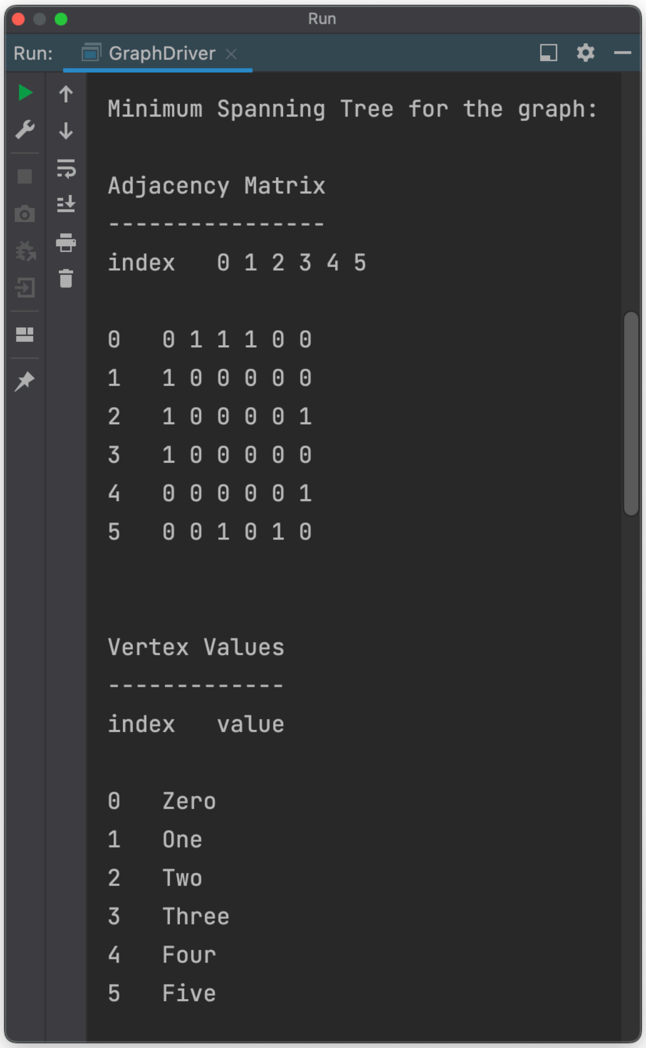
Question 2 Test Exhibits

Test exhibit showing `.addVertex()`, `.addEdge()`, and `.isConnected()` functionality



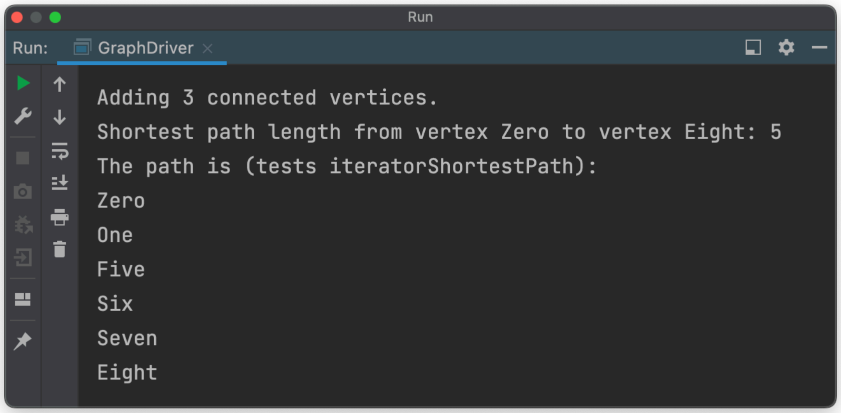
```
/Library/Java/JavaVirtualMachines/jdk-17.0.1
Created and populated the following graph:

Adjacency Matrix
----------------
index   0 1 2 3 4 5

0   0 1 1 1 0 0
1   1 0 1 0 1 1
2   1 1 0 0 0 1
3   1 0 0 1 0 0
4   0 1 0 0 0 1
5   0 1 1 0 1 0


Vertex Values
-------------
index   value

0   Zero
1   One
2   Two
3   Three
4   Four
5   Five


Graph is connected: true
```

Test exhibit showing `.getMST()` functionality and the above graphs minimum spanning tree:



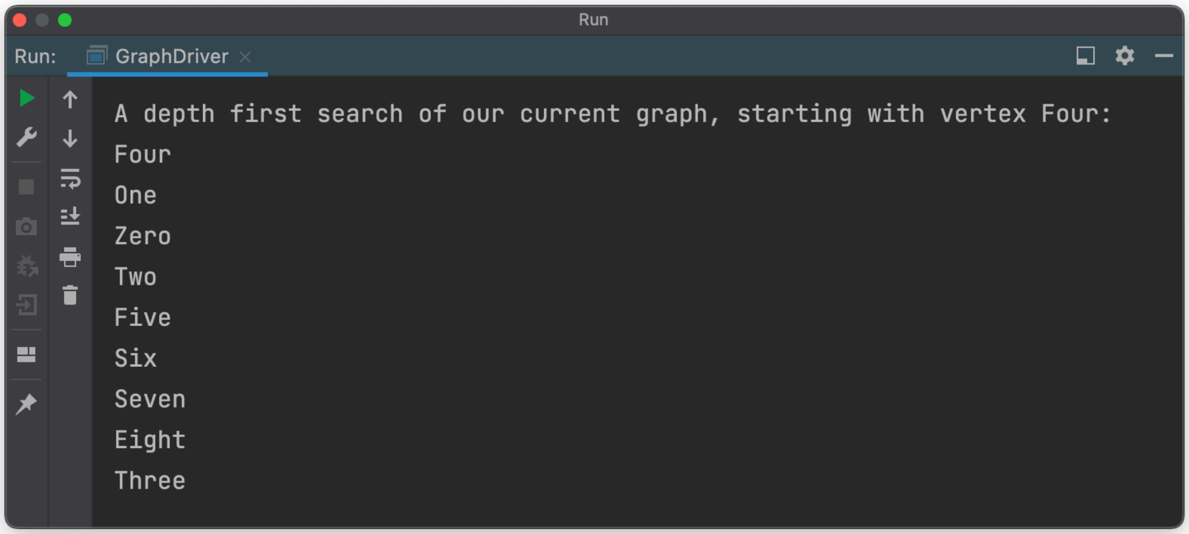
```
Minimum Spanning Tree for the graph:

Adjacency Matrix
----------------
index   0 1 2 3 4 5

0   0 1 1 1 0 0
1   1 0 0 0 0 0
2   1 0 0 0 0 1
3   1 0 0 0 0 0
4   0 0 0 0 0 1
5   0 0 1 0 1 0



Vertex Values
-------------
index   value

0   Zero
1   One
2   Two
3   Three
4   Four
5   Five
```

showing `.shortestPathLength()` and `.iteratorShortestPath()` functionality:



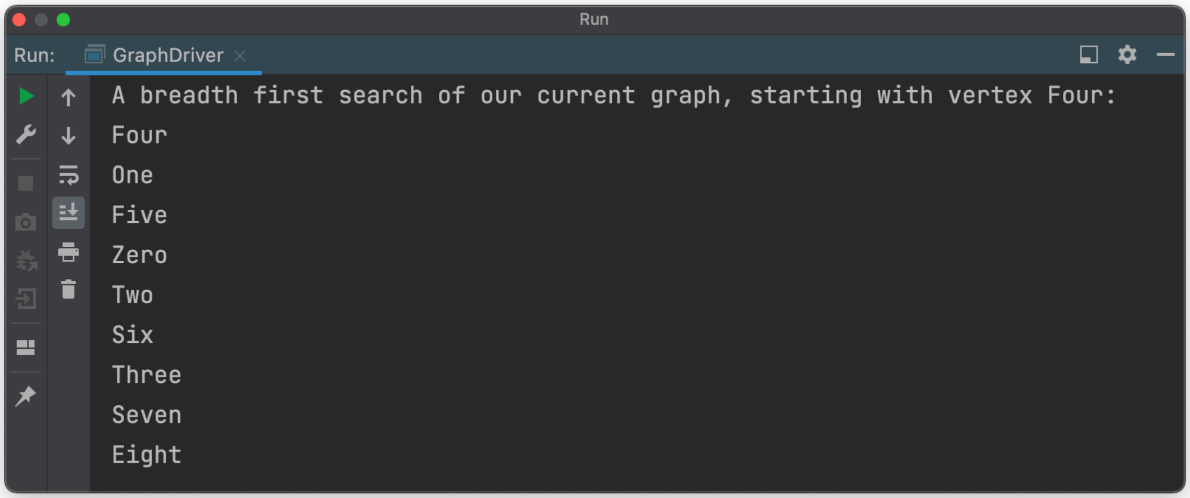
```
Adding 3 connected vertices.
Shortest path length from vertex Zero to vertex Eight: 5
The path is (tests iteratorShortestPath):
Zero
One
Five
Six
Seven
Eight
```

showing `.iteratorDFS()` (depth first search) functionality, starting with vertex Four:



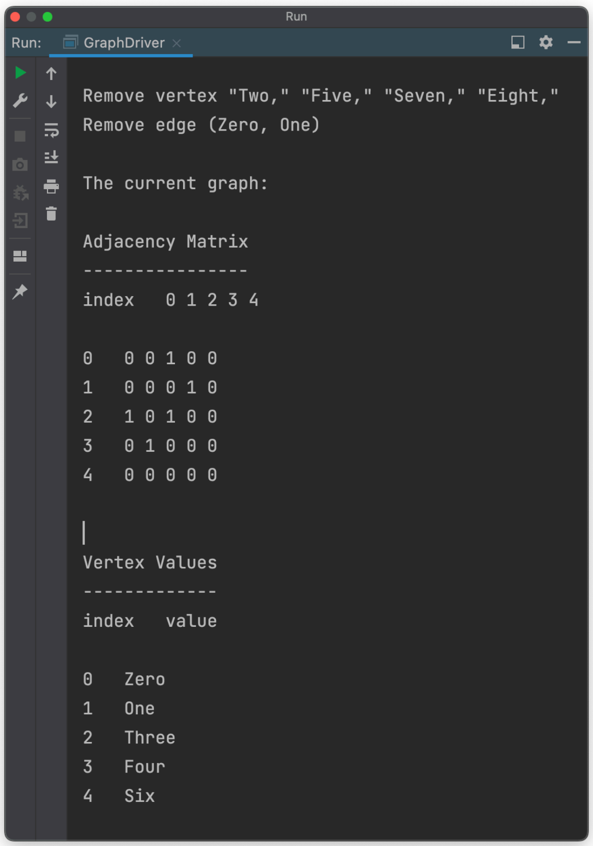
```
A depth first search of our current graph, starting with vertex Four:
Four
One
Zero
Two
Five
Six
Seven
Eight
Three
```

showing `.iteratorBFS()` (breadth first search) functionality, starting with vertex Four:



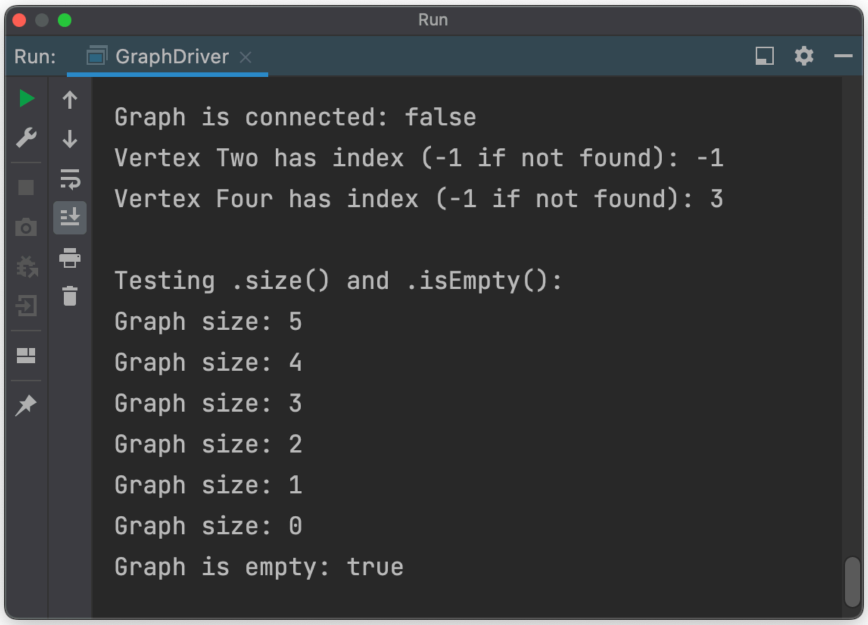
```
A breadth first search of our current graph, starting with vertex Four:
Four
One
Five
Zero
Two
Six
Three
Seven
Eight
```

showing `.removeVertex()` and `.removeEdge()` functionality:



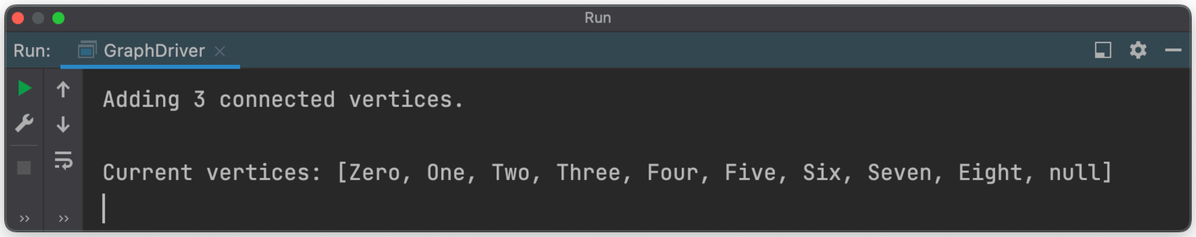
```
Remove vertex "Two," "Five," "Seven," "Eight,"
Remove edge (Zero, One)

The current graph:

Adjacency Matrix
----------------
index   0 1 2 3 4

0   0 0 1 0 0
1   0 0 0 1 0
2   1 0 1 0 0
3   0 1 0 0 0
4   0 0 0 0 0

|
Vertex Values
-------------
index   value

0   Zero
1   One
2   Three
3   Four
4   Six
```

Test Exhibit showing `.isConnected()` for a disconnected graph, `.getIndex()`, `.size()` and `.isEmpty()` functionality for a range of cases:



```
Graph is connected: false
Vertex Two has index (-1 if not found): -1
Vertex Four has index (-1 if not found): 3

Testing .size() and .isEmpty():
Graph size: 5
Graph size: 4
Graph size: 3
Graph size: 2
Graph size: 1
Graph size: 0
Graph is empty: true
```

showing `.getVertices()` — input as `Arrays.toString(graph.getVertices());`
Note also that this implicitly shows `expandCapacity()` is functioning.



```
Adding 3 connected vertices.

Current vertices: [Zero, One, Two, Three, Four, Five, Six, Seven, Eight, null]
```