CSC 2053 Undirected Graphs

1. Draw the graph represented in this file and its adjacency list representation:

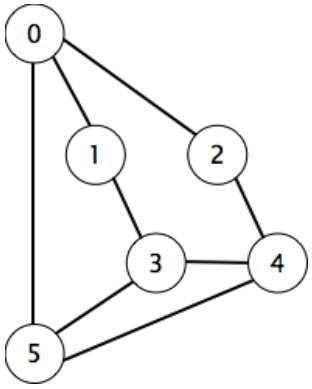
graph: adjacency list representation:

```
6
8
0 5
2 4
2 3
1 2
0 1
3 4
3 5
0 2
```

2. Download `Graph.java`

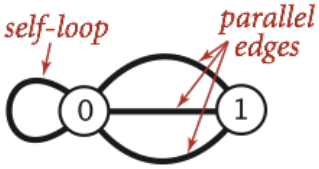
Run it with the above graph as input, check that the adjacency list representation matches what you wrote, above.

3. Create an input file corresponding to this graph:



Run `Graph.java` with the above graph as input, check the adjacency list representation to verify that you got all the edges.

4. Create an input file corresponding to this graph:



Run `Graph.java` with the above graph as input. Check the adjacency list representation to verify that you got all the edges.

public class	Graph	
	Graph(int V)	*create a V-vertex graph with no edges*
	Graph(In in)	*read a graph from input stream in*
int	V()	*number of vertices*
int	E()	*number of edges*
void	addEdge(int v, int w)	*add edge v-w to this graph*
Iterable<Integer>	adj(int v)	*vertices adjacent to v*
String	toString()	*string representation*

API for an undirected graph

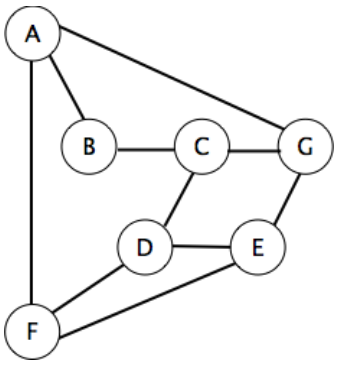
5. Download `GraphClient.java`

Add another method that determines whether a graph has an Euler circuit.
Note: You do NOT need to find the circuit, just determine whether the graph has one.

6. Download `SymbolGraph.java`

Run it with **`routes.txt`** and **`movies.txt`** as input.

7. Create an input file corresponding to this graph:



Run `Graph.java` with the above graph as input, check the adjacency list representation to verify that you got all the edges.

8. Add a `toString()` method to `SymbolGraph.java` that outputs the graph using symbols instead of numbers

This is similar to the toString() method in Graph.java (but you need to replace the numbers by the corresponding strings). Modify the unit test to also print the graph, and make sure it prints the information correctly. It should look like this:

```
> java SymbolGraph routes.txt " "
```

```
Done reading routes.txt
10 vertices, 18 edges
JFK: ORD ATL MCO
MCO: HOU ATL JFK
ORD: ATL JFK PHX DFW HOU DEN
DEN: LAS PHX ORD
HOU: MCO DFW ATL ORD
DFW: HOU ORD PHX
PHX: LAS LAX DEN ORD DFW
ATL: MCO ORD HOU JFK
LAX: LAS PHX
LAS: PHX LAX DEN
```

public class SymbolGraph	
SymbolGraph(String filename, String delim)	*build graph specified in* filename *using* delim *to separate vertex names*
boolean contains(String key)	*is* key *a vertex?*
int index(String key)	*index associated with* key
String name(int v)	*key associated with index* v
Graph G()	*underlying* Graph

9. Create a client `BiggestAirport.java`

Use SymbolGraph to find the airport with the largest number of connections in routes.txt

10. Create a `SymbolGraphClient.java`

It should have the same functionality (including the Euler circuit method), but uses instead symbol graphs.

Try it with **`routes.txt`** and **`movies.txt`** as input.