Lab 13

In this lab you will develop your graph package of software. Based on direction from the slides, finish the implementation of the operations on a Graph:

```
boolean areAdjacent(Vertex u, Vertex v)
List getListOfAdjacentVerts (Vertex u)
Graph getSpanningTree()
List getConnectedComponents()
boolean isConnected()
boolean hasPathBetween(Vertex u, Vertex v)
boolean containsCycle()
boolean isTree()
boolean isBipartite()
```

DFS and the spanning tree algorithm have already been implemented. You will need to use observations given in the slides to provide the connected components of the graph, determine whether the graph has a cycle, and to determine if there is a path joining two given vertices.

You will also implement BFS, and, in a subclass, implement the additional work needed to determine if the graph has an odd cycle (so you can determine whether it is a bipartite graph).

Finally, I have provided a second constructor in Graph that accepts an array of Edges (in the form of Objects). One use for this constructor is that it allows you to return a spanning tree as a Graph object after performing your spanning tree algorithm.

The toString method that is provided may not be suitable – you should modify it as necessary so that you can display test results in a useful way.