Exercise 3- Graph Traversal

isConnected() & isCyclic() (15 points)

Expand on your graph representation project. For a given file typed by the user: (for example test1.tx). Your program should display

1. Whether or not the graph is connected
2. Whether or not there is a cycle

Sample console interaction input:

Enter a text file:
tree.text
The graph is connected
There is no cycle in the graph

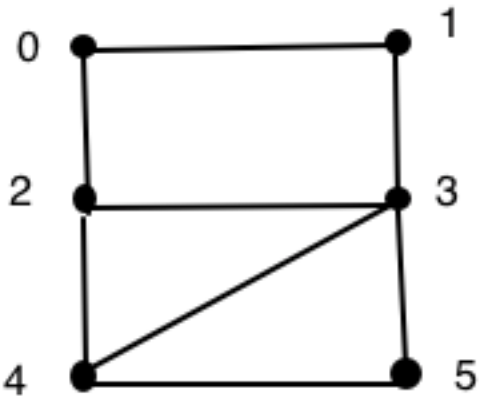
It must also display the cycles in the graph on a JavaFX window (similar to the previous exercise)

Part 1- isConnected() (10 points)

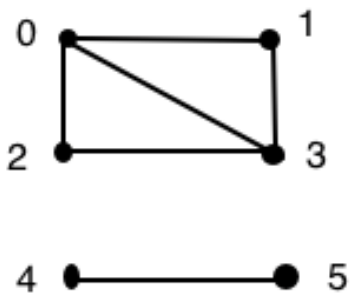
Assume the text file follows the same format as the previous graph representation project where the first line in the file contains a number that indicates the number of vertices (n). The vertices are labeled as 0, 1, …n, n-1. Each subsequent line with the formate u v1 v2…. Describes the edges (u, v1), (u, v2) and so on. **Your program should prompt the user to enter the name of the file, then it should read data from the file, and invoke dfs() or bfs() to decide whether the graph is connected or not.** You will need to implement dfs or bfs in your code as a method, I recommend the following two resources:
https://www.geeksforgeeks.org/depth-first-search-or-dfs-for-a-graph/

https://www.geeksforgeeks.org/breadth-first-search-or-bfs-for-a-graph/

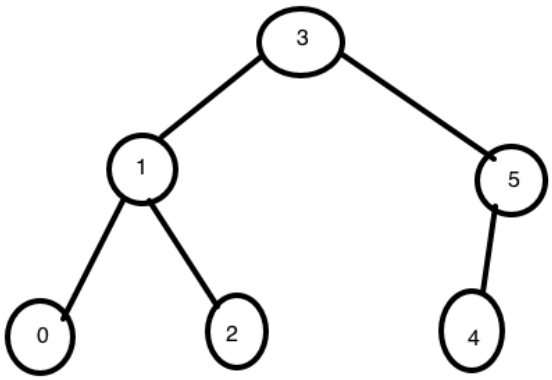
File example of connected.txt

6 0 1 2 1 0 3 2 0 3 4 3 1 2 4 5 4 2 3 5 5 3 4	

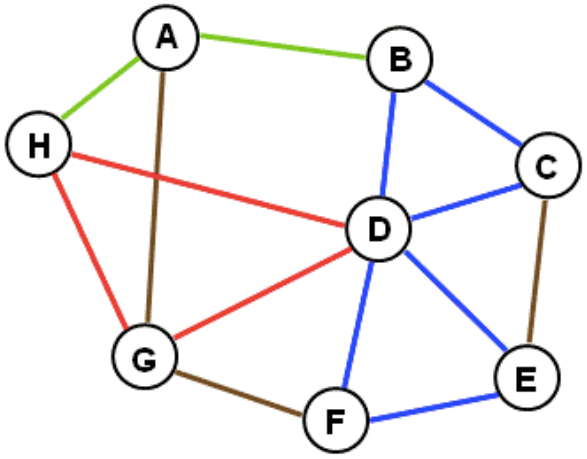
File example of notConnected.txt

6 0 1 2 3 1 0 3 2 0 3 3 0 1 2 4 5 5 4	

File example of tree.txt

6 **0 1** **1 0 2 3** **2 1** **3 1 5** **4 5** **5 3 4**	

Part 2-findCycles() (5 points)



A graph has a cycle if there is a path such that you can begin and end at the same vertex visiting every vertex one time without any vertex visited twice (except for the first and last). **You must use either bfs or dfs (you decide) to find if there is a cycle through each of the vertices! Then, print out one of the cycles, both in console and on the screen, they can all be on the same screen.**

Sample output:
A: True, A B C D E F G H
B: True, B C D E F…..