CPE 111 Programming with Data Structures
International Sections, January 2021
Laboratory Exercise 9-1

Objective

This lab is intended to give you the opportunity to work with a graph implemented using an adjacency list.

Instructions

1. Download all files from the **demos/Lecture9** area of the website. Run make to build **graphTester**. Spend some time experimenting with this program to make sure you understand how it works.

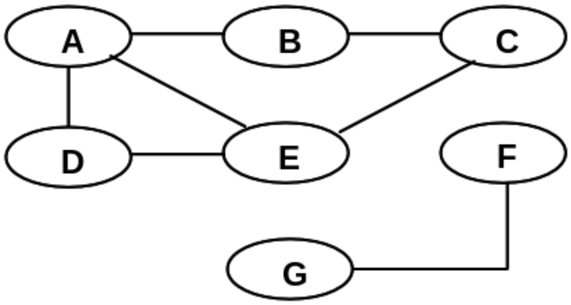
2. The header file **abstractGraph.h** includes a function ***isReachable***. There is also a placeholder in **linkedListGraph.c** for this function.

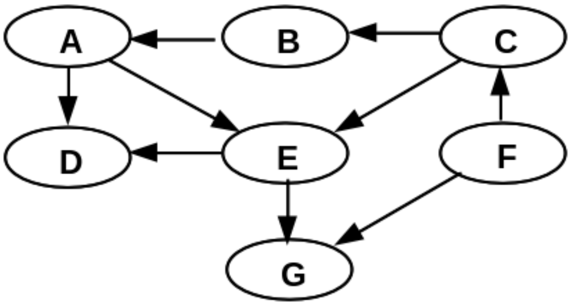
```
/* Return information as to whether two vertices are
 * connected by a path.
 * Arguments
 *    key1 -  Key for the start vertex
 *    key2 -  Key for the second vertex to check
 * Returns 1 if the two vertices are connected, 0 if they
 * are not. Returns -1 if either vertex does not exist.
 */
int isReachable(char* key1, char* key2);
```

Modify **linkedListGraph.c** to provide code for the function ***isReachable*** (that is, implement the function).

3. Modify the code in **graphTester.c** to ask the user for the first and second keys and call the function **isReachable**. There is already a menu item and a place in the switch statement for you to put your code. Be sure that you print the results correctly. If the function returns 1, print that vertex 2 is reachable from vertex 1. If the function returns 0, print that vertex 2 is not reachable from vertex 1. If the function returns 1, print an error message indicating that one of the vertices does not exist.

4. Compile and test your code until you are sure that it works. Then test it on the following undirected and directed graphs. (You will have to build these graphs using the other options in **graphTester**.)





Is E reachable from F?
Is C reachable from A?
Is G reachable from A?
Is G reachable from F?
Is C reachable from C?

5. Now add code so that if the second vertex is reachable from the first, the program will print the path from the first vertex to the second (that is, it prints the vertex keys, in order).

6. Upload your modified version of **linkedListGraph.c**, plus a text file called **Answers.txt** where you answer the questions above. (Create this file using gedit or Notepad or some other text editor, not Word!)

Be sure that you put your own name and student ID in the program header comment. If you upload a program with my name on it, you will get a zero score!