Introduction
This project will be an introduction to graphs and some of the concepts involved with graphs. The main idea here will be to read in a set of vertices, a set of which vertices are connect, and then perform a depth first search to decide whether or not the graph is a connect graph.

Optional Assignment
This assignment is optional for your grade. You may opt to use the first two minor assignments as your final minor assignment grade, or you may choose to do this assignment and have your grade be based on three minor assignments. This assignment will not replace any of the previous assignments, but may lesson the impact any lost points may have on your final grade.

Input
Your input will come from a file named “graph.in” and will consist of the follow setup:

A list of nodes. The list of nodes will be terminated by a *. The nodes can be anything from numbers to letter or even words. Make your program not care what the list of nodes is.
A list of the connect nodes. This list will contain ordered pairs of nodes that are connected. This list will also be terminated by a *.

Output
Your output will be written to a file called “graph.out”. At the top will be your name, ID number and the project title, Project 5: Graphs. After that you should print a delimiter of some sorts and one of the two following statements:

The graph has 16 nodes, and is connected.

or

The graph has 16 nodes, and is disconnected.

These statements make the assumptions that the list had 16 nodes in it. The last word will depend on whether the graph is connected or not. In either case, the number of nodes should be computed and then printed in the message.

More details about the input and output will be forth coming.

Submitting your project
You will submit a single cpp file to the Curator System (read the Student Guide), and it will be archived. Your projects will be hand graded and the grades will be posted by the last day of class. Instructions for submitting are contained in the Student Guide. You will be allowed up to three submissions for this assignment, in case you need to correct
mistakes. Test your program thoroughly before submitting it. If you discover an error you may fix it and make another submission. Your last submission will be graded; so fixing an error after the due date will result in a late penalty.

The *Student Guide* and link to the submission client can be found at: http://www.cs.vt.edu/curator/

**Programming Standards:**
The TAs will be carefully evaluating your source code on this assignment for programming style, so you should observe good practice. See the Programming Standards page on the course website for specific requirements that should be observed in this course. Your submitted program will be assigned a score based upon the runtime testing performed by the Curator System. After that, your program will be given a brief evaluation by one of the TAs, who will consider:

- whether your implementation makes appropriate use of data structures and
- whether your design shows a good object-oriented decomposition of the given problem, and
- whether the internal documentation of your code is acceptable.

See the Programming Standards page on the course website for specific requirements that should be observed in this course. This evaluation will produce a second grade that will be kept separate from the grade for the output. At the end of the semester, both grades will be averaged and combined to produce a single overall project grade.

**Pledge:**
Each of your program submissions must be pledged to conform to the Honor Code requirements for this course.

Specifically, you **must** include the pledge statement provided below.

```c++
// On my honor:
//
// - I have not discussed the C++ language code in my program with
//   anyone other than my instructor or the teaching assistants
//   assigned to this course.
//
// - I have not used C++ language code obtained from another student,
//   or any other unauthorized source, either modified or unmodified.
//
// - If any C++ language code or documentation used in my program
//   was obtained from another source, such as a textbook or course
//   notes, that has been clearly noted with a proper citation in
//   the comments of my program.
//
// - I have not designed this program in such a way as to defeat or
//   interfere with the normal operation of the Curator System.
//
// <Student's Name>
```