CSCI-1200 Computer Science II — Fall 2008
Homework 8 — Friendly Recusion

Additional info on friend functions

We've asked you to implement the output stream operator for the `Graph` class. We have seen how the output stream operator can be implemented as a non-member function. So outside of the class declaration you make a function with this prototype:

```
ostream& operator<<(ostream &ostr, const Graph &graph) {

  // body of function

}
```

Remember that non-member functions only have access to the public interface of the class. So to be able to print out all of the necessary info from within this function you will need to add a bunch of public accessor functions to the `Graph` data that should probably be private. There must be a better solution...

Often the solution is to make the function that needs private access a member function of the class. However we can't do this for the ostream operator because the first argument to the function is an ostream object, *not* an `Graph` object. Instead, we leave the function as a non-member function, but declare that it is a "friend" of the `Graph` class and should have access to the private variables and functions it ordinarily would not. To do this, add the following line within the public portion of your `Graph` declaration:

```
friend ostream& operator<<(ostream &ostr, const Graph &graph);
```

The syntax for the implementation of the function (shown above) does not change. The keyword "`friend`" only appears inside of the class declaration.

We will discuss friend functions in lecture on Friday.