READ THIS FIRST:

Do your best to do every item on your own; if you cannot immediately do an item, go on to others and then come back to it later. Please ask your professor whenever you have a question.

Due: Monday, March 20, 2017.

Goals:

- Practice implementing your own DFAs in Java
- Practice getting around in and using GitHub.
- Explains some key concepts we covered in class.
- Get some easy homework points.

Background: DFAs are a useful tool to represent languages and do other cool stuff in computer science, such as the lexicon part of a compiler, syntax highlightin on IDEs, to name a few. In this assignment you will get to do some Java programming to implement a DFA.

Instructions: Using a table-driven DFA approach, write a Java class `ManWolf` that takes a string from the command line and reports whether or not it represents a solution to the man-wolf-goat-cabbage problem of Chapter 2 in the textbook. You will implement your `ManWolf` class (`ManWolf.java`) in a driver java file named `driverDFA.java` that contains only the part that reads from standard input, calls the functions of the `ManWolf` class, and prints the result to standard output. For example, it should have this exact same behavior:

```
> java driverDFA gncgwng
That is a solution.
> java driverDFA ggggggggg
That is not a solution.
```

Resources:

- Chapter 2 of the textbook has the DFA for the man-wolf-goat-cabbage problem
- Chapter 4 of the textbook contains **a lot** of Java code you can reuse, and it also contains an example of the table-driven DFA approach.
- The coding style guidelines are here: `http://www.reev.us/cmpt440s17/style.html`

Deliverables:

- a `ManWolf.java` file containing the abstract class of the man-wolf-goat-cabbage problem, i.e., class definition, all its attributes, all its methods, following Dr. Rivas' Java coding style guidelines.
- a `driverDFA.java` file containing the implementation of the class above, following Dr. Rivas' Java coding style guidelines.
- an excel, or csv file that contains the state-transition table
- an excel, or csv file that contains the description of each state

Submission: Push your four files (see deliverables) to your GitHub repository **before the due date** (see the top of this document). **Make sure you follow the style guidelines** for Java provided by your professor in the Resources section above. Remember to include your name, the date, and the assignment in the (copious, meaningful, and accurate) check-in messages.