CSE 341, Winter 2014, Assignment 5
Prolog Warmup
Due: Wednesday February 12, 10:00pm

9 points total (3 points each)

Include appropriate unit tests for each of your top-level rules. You can use helper rules as needed.

You can use up to 2 late days for this assignment.

Turnin: Turn in two files named `prolog_warmup.pl` (with the rules themselves) and `prolog_warmup_tests.pl` (with the unit tests). The TAs should then be able to run the unit tests. There is a starter file for the unit tests linked from the assignments page.

Hint: if you don't want to deal with the unit tests right off, just write the `prolog_warmup.pl` program, one rule at a time, testing it from the command line as you go. Then get everything working with the unit tests. (This is of course contrary to the best-practice approach of writing the tests first, but for this new language, you might find it simpler not to deal with unit tests right off.)

1. Write a Prolog rule `repeat` that succeeds if the second argument is a list with only 0 or more occurences of the first argument. For example, these goals should succeed:

   ```
   repeat(squid,[]).
   repeat(squid,[squid,squid,squid,squid]).
   ```

   and this should fail:

   ```
   repeat(squid,[squid,squid,squid,clam]).
   ```

   The starter file includes enough unit tests for this rule, although you can add some others if you want.

   In addition, try your goal with a variable for either the first or second argument. You don't need to include any output from this however. Backtrack a few times if there are more answers available. For example try:

   ```
   repeat(clam,Xs).
   repeat(X,[squid,squid,squid]).
   ```

2. Write a Prolog rule `average` that computes the average of a list of numbers. Fail if the list is empty. (You don't need to worry about lists of things that aren't numbers.) There are suitable unit tests already in the starter file.

3. Prolog sentences: write a Prolog rule `sentence` that succeeds if the first argument is the definite article (in other words, the atom "the"), the second article is a noun, and the third article is a verb. Define at least four facts about nouns (so that Prolog has at least four possible nouns to use in sentences), and also at least four facts about verbs. For example, your facts might include:

   ```
   noun(octopus).
   verb(swims).
   ```

Then the goal `sentence(the, octopus, swims).` should succeed.

There are a few unit tests for these rules in the starter file — modify these as needed if you use other facts. You should have at least 4 unit tests, including one for a goal that fails.

In addition, try your goal with variables for all three arguments. Backtrack if there are more answers available. You don't need to turn in all the output for this. *However, in a comment in your code, say how many different answers you found, and explain why Prolog found that number of answers.*