Homework requirement:
From now on, use anonymous variables as described in Section 1.13. Points will be taken off for consult warnings because of singleton variables.

Exercises 2 – 5 are programming problems. Submit them in a single file named `a10.pl` electronically per the instructions for your course.

1. Study Bratko, Sections 3.3 to 3.4.

2. Do Bratko, Exercise 3.4.
   Name your predicate `my_reverse`, and do not use the built-in predicate `reverse`. The recursive rule should have two goals, one of which is `my_reverse`, but the other of which is `conc`.

3. Do Bratko, Exercise 3.5.

4. Do Bratko, Exercise 3.6.
   You can do it with only one rule with `conc`. Note that the name of the predicate is `shift`, *not* `my_shift`.

5. Do Bratko, Exercise 3.11.
   `flatten` is built-in to `gprolog`, so you must name your predicate `my_flatten`. Here are two queries to illustrate the two base cases.

```
?- my_flatten( [], X).

X = []

?- my_flatten( a, X).

X = [a]
```

   Note that `my_flatten/2` must produce a list, even if the first argument is not a list. This is an unusual problem, because the base cases must be placed *after* the single recursive rule. When you test your predicate you will probably get erroneous instantiations after the first correct unification. With the tools we have so far in the course, you cannot avoid the later erroneous results. Think about why you cannot.

```
?- my_flatten( [a,b,[c,d],[],[[[e]]],f], List).

List = [a,b,c,d,e,f]
```