Problem Statement
The goal of this assignment is to get acquainted with the programming language Prolog!

Requirements Analysis
This assignment has two parts (worth 67% and 33% of the grade, respectively). For the first part Patrick needs a magic square, while for the second Sandy needs to explore the versatility of the append/3 relation. In a magic square, the sum of the values in the rows, columns, and diagonals are all the same. Each value is a single digit number and each digit can be used exactly once.

Design: Magic Squares
Write a Prolog program using generate and test to find magic squares of size three by three. With generate and test you first generate a proposed solution, in this case a permutation of the numbers 1,2,· · · , 9 and then test if the proposed solution is a solution.

While most Prolog systems include a predefined permutation rule, for this assignment, please code your own.

Design: Universal append
Using only the append relation, formulate queries to determine the following.
  a) The third element of a list.
  b) The last element of a list.
  c) All but the last element of a list.
  d) Whether a list is a concatenation of three copies of the same sublist.
  e) Whether a list Y is formed by inserting an element A somewhere in a list X.

Then produce three tests for each.

What to hand in
(1) Email your code and output as four attachments
   <your name>-square.pl
   <your name>-square.out (collected using script(1))
   <your name>-appends.pl
   <your name>-appends.out
   Consider cleanup and annotating your output.

Notes
(1) Man script(1).
(2) There exist solutions to these problems on the web. I fully believe that you have mastered the zen of downloading “stuff” off the web. Your goal here is not to practice this zen art, but to learn some Prolog. To that end please avoid Mr. Internet once you start writing your code.