```
%%loads file in to use its rules
:- consult('~/Documents/CSC366/Assignments/assignment03/gv2.pro').

establishCryptoProblemParameters :-
      declare(lo, 0),
      declare(hi, 15).

%to ensure above rule is immediately ready for use
:-establishCryptoProblemParameters.

generateRandomCryptoNumber(N) :-
      valueOf(lo, Lo),
      valueOf(hi, Hi),
      NHi is Hi + 1,
      random(Lo, NHi, N).

generateRandomCryptoProblem :-
      generateRandomCryptoNumber(N1),
      generateRandomCryptoNumber(N2),
      generateRandomCryptoNumber(N3),
      generateRandomCryptoNumber(N4),
      generateRandomCryptoNumber(N5),
      generateRandomCryptoNumber(N6),
      addCryptoProblemToKnowledgeBase(N1, N2, N3, N4, N5, N6).

addCryptoProblemToKnowledgeBase(N1, N2, N3, N4, N5, N6) :-
      retract(problem(_,_)),
      assert(problem(numbers(N1, N2, N3, N4, N5), goal(N6))).

addCryptoProblemToKnowledgeBase(N1, N2, N3, N4, N5, N6) :-
      assert(problem(numbers(N1, N2, N3, N4, N5), goal(N6))).

displayProblem :-
      problem(numbers(N1, N2, N3, N4, N5), goal(N6)),
      write('Numbers = {'),
      write(N1), write(', '),
      write(N2), write(', '),
      write(N3), write(', '),
      write(N4), write(', '),
      write(N5), write('} Goal = '),
      write(N6), nl.

genone :-
      generateRandomCryptoProblem,
      displayProblem.
generate(1) :-
      genone.
generate(N) :-
      genone,
      NM1 is N - 1,
      generate(NM1).
```