```
%%%% File: crypto.pro
%%%% Line Generating Random Crypto Problems

% Load global variables ADT
:- consult('/../../gv1.pro').

% establish the problem parameters
establishCryptoProblemParameters :-
  declare(lo, 0),
  declare(hi, 15).
:- establishCryptoProblemParameters.

% generate a random number with range
generateRandomCryptoNumber(N) :-
  valueof(lo, Lo),
  valueof(hi, Hi),
  HiPlus1 is Hi + 1,
  random(Lo, HiPlus1, N).

% generate 1 random crypto and add to KB
generateRandomCryptoProblem :-
  generateRandomCryptoNumber(N1),
  generateRandomCryptoNumber(N2),
  generateRandomCryptoNumber(N3),
  generateRandomCryptoNumber(N4),
  generateRandomCryptoNumber(N5),
  generateRandomCryptoNumber(G),
  addCryptoProblemToKB(N1,N2,N3,N4,N5,G).
```

```
% add crypto to KB
addCryptoProblemToKB(N1,N2,N3,N4,N5,G) :-
  retract(problem(_,_)),
  assert(problem(numbers(N1,N2,N3,N4,N5), goal(G))).
addCryptoProblemToKB :-
  assert(problem(numbers(N1,N2,N3,N4,N5), goal(G))).

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

% demo
demo :-
  generateRandomCryptoProblem,
  displayProblem.

% genone
genone :- demo.
```

```
% generate N amount of cryptos
generate(1) :- genone.
generate(N) :-
  genone,
  NM1 is N - 1,
  generate(NM1).
```