

```
% crypto.pro
% Prolog code to generate and display a random crypto problem

% Uses gv.pro
:-consult('gv.pro').

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

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

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

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

displayProblem :-
    problem(numbers(N1,N2,N3,N4,N5),goal(G)),
    write('Problem: numbers = {'),
    write(N1),write(','),
    write(N2),write(','),
    write(N3),write(','),
    write(N4),write(','),
    write(N5),write('} and goal = '),
    write(G),nl.

genone :-
    generateRandomCryptoProblem,
    displayProblem.
```