

**CSc 135 – Fall 2016 - Programming Languages
Homework 3**

due Wednesday December 7 at 11:59:00 pm

Note: make sure you are submitting your work on-line I will not accept files sent to me.

Write the following as Prolog rules:

1. Implement a rule "dogEnthusiast".
Someone is a "dogEnthusiast" if they own AT LEAST TWO dogs.
Assume that the only types of facts available are:

"owner" facts of the form owner(p,n), meaning that person p owns animal named n.
"breed" facts of the form breed(a,b), meaning that animal named a is of type b.

For example:

owner(fred,fido) means that a person named "fred" owns an animal named "fido".

breed(fido,dog) means that the animal named "fido" is a dog.

breed(leo,cat) means that the animal named "leo" is a cat.

dogEnthusiast(sam) would be true if sam owned two different animals that were dogs.

2. Implement "listPicker" from Homework #2 as a Prolog rule.

For example, the query: listPicker([42,3,99,7],[4,3,3],M)

should result in Prolog's response: M = [7,99,99].

You only need to handle lists that contain integers.

3. Implement a rule "crypto" that solves the following cryptarithmic multiplication problem:

TOCK * TOCK = GRIPTOCK

Each of the 4 letters (T,O,C,K) stands for a different digit.

The 4 letters (G,R,I,P) can each be matched by any digit.

The aim is to find a substitution of digits for the letters such that the above stated product is arithmetically correct.
Your program should find all answers.

It should be possible to query your solution in this manner:

?- crypto(G,R,I,P,T,O,C,K).

Your solution should then produce all of the combinations of the digits that satisfy the multiplication problem above. Don't get confused between the letter "O" and the number "0" (zero).

Make sure you never let T=C, or C=K, etc... all of the distinct letters T,O,C,K have to stand for distinct digits.

Don't be surprised if it takes your computer a while to solve this problem. It took Dr. Gordon's desktop computer about a minute to complete its answer using his solution... and he has a pretty fast computer.

Use generate-and-test!

4. Implement a rule "interleave" that takes two lists, and combines them into a single list by alternating the elements. For example:

?- interleave([3,4,5],[7,8,9],X) should respond X=[3,7,4,8,5,9]

If the two lists are different lengths, include the excess elements at the end of the result list, with a "999" inserted first. For example:

?- interleave([1,2],[4,5,6,7],X) results in X=[1,4,2,5,999,6,7]

?- interleave([1,2,3,4],[6,7],X) results in X=[1,6,2,7,999,3,4]

5. Convert "digitinc" from Homework #2 into a Prolog rule.
For example:

?- digitinc(22897,X) should respond X=33908.

Submit your solutions in a single file.