CSc 135 – Spring 2017 - Programming Languages
Homework 3

due Saturday May 13 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. Using only the append relation (i.e. append (L1, L2, L3)), formulate queries to determine the following (Note although your answer should use L3 = [1,2,3,4,5,6,7,8] it should work with any list of five elements or more given as L3)
   a. The fifth element of list [1,2,3,4,5,6,7,8].
   b. The third before last element of the same list.

2. Implement a rule "**bookReader**".

   Someone is a "bookReader" if they read AT LEAST THREE books in a year.
   Assume that the only types of facts available are:

   "enjoys" facts of the form enjoys(p,a), meaning that person p enjoys object a.
   "objectKind" facts of the form objectKind(a,b), meaning that object named a is of type b.

   For example:

   enjoys(mary,mrToad) means that a person named "mary" enjoys an object named "mrToad".
   objectKind(mrToad,book) means that the object named "mrToad" is a book.
   objectKind(joconda,painting) means that the object named "joconda" is a painting.

   bookReader(mary) would be true if mary enjoys three different objects that were books.

3. Implement a rule "**crypto**" that solves the following cryptarithmetic addition problem:

   WAIT + ALL = GIFTS

   Each of the letters stands for a different digit.
   The 3 letters (W, A, G) cannot be 0.
   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(W,A,I,T,L,G,F,S)).
   ```

   Your solution should then produce all of the combinations of
   the digits that satisfy the addition problem above.

Make sure you never let W=A, or I=T, etc... all of the distinct letters must stand for distinct digits.

Don't be surprised if it takes your computer a while to solve this problem.

Use generate-and-test!

4. Consider the following set of Prolog rules for reverse:

```
( 1 )     reverse( X, Z )          :-     rev ( X, [ ] , Z ).
( 2 )     rev( [ ], Y, Y ).
( 3 )     rev( [ A | X ], Y, Z )   :-     rev ( X , [ A | Y], Z ).
```

Trace the order of execution for the following query: **reverse([5,6,3], W).**
Present your trace step by step as I did in the notes. For each call, indicate the appropriate subgoal that is attempting to be proven. Indicate where an answer is found. Make clear what you are doing.

Submit your solutions in a single file.