Assignment #3
CS 3060 Programming Languages, Fall 2020
Instructor: S. Roy

Prolog #1

Due Date: Oct 3 @ 11.59 PM.
Total Points: 60 points

Directions: Using the source provided via Gitlab `https://gitlab.com/sanroy/fa20-cs3060-hw/`, complete the assignment below. The process for completing this assignment should be as follows:

1. You already forked the Repository “sanroy/fa20-cs3060-hw” to a repository “yourId/fa20-cs3060-hw” under your username. If not, do it now.
2. Get a copy of hw3 folder in “sanroy/fa20-cs3060-hw” repository as a hw3 folder in your repository “yourId/fa20-cs3060-hw”
3. Complete the assignment, committing changes to git. Each task code should be in a separate prolog file. As an example, task1.pl for Task 1.
4. Push all commits to your Gitlab repository
5. If you have done yet done so, add TA and Roy as a member (in ‘Developer’ mode) of your Gitlab repository

Tasks:

1. **Task #1:** (20 points) Create a knowledge base of your choosing. This knowledge base must include at least 12 items (including at least 8 facts and 4 rules) and you must come up with at least 6 querries (whereas at least two querries will involve facts, at least two querries will involve rules, and at least two queries will have answer NO from prolog). Run the querries and show the answers (in readme) you get from Prolog. `Writing readme carries 2 points.`

2. **Task #2:** (20 points) Write a rule that will find the largest element of a list of integers. Run at least 2 querries and show the answers (in readme) you get from Prolog. Your rule should work even if there are duplicate elements, postive integers, zero, and negatives integers. `Writing readme carries 2 points.`

3. **Task #3:)** (20 points) Consider the following knowledge base: Interpretation of *hasDirectFlight(x,y)* is that there is a one-way direct flight connection from airport x to y.

   ```
   hasDirectFlight(newOrleans, chicago).
   hasDirectFlight(philadelphia, newOrleans).
   hasDirectFlight(columbus, philadelphia).
   hasDirectFlight(sanFrancisco, columbus).
   hasDirectFlight(detroit, sanFrancisco).
   hasDirectFlight(toledo, detroit).
   hasDirectFlight(houston, sanFrancisco).
   ```

   Write a recursive rule `hasFlightRoute/2` that tells us whether there is a flight route from one town A to another town B. Run at least 3 querries (one with at least 2-hop route, one with one-hop route, one with no route) and show the answers (in readme) you get from Prolog. `Writing readme carries 2 point.`