CSE341 – Programming Languages (Fall 2016)

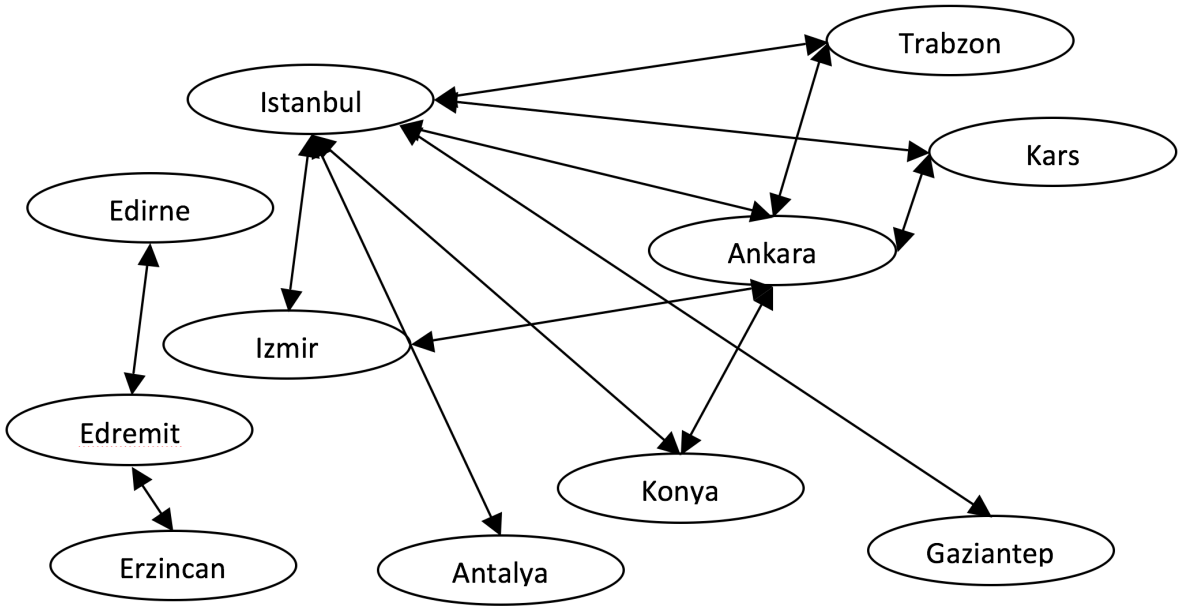
Homework #2

Handed out: 8:00am Monday November 28, 2016.

Due: 11:55pm Wednesday December 14, 2016.

Hand-in Policy: Source code and documentation should be submitted online as a single .zip or .rar file with naming convention studentid_lastname_firstname_hw2.zip via Moodle by the submission deadline. No late submissions will be accepted.
Collaboration Policy: No collaboration is permitted. Any cheating (copying someone elses work in any form) will result in a grade of -100 for the first offense and -200 for the subsequent attempts.
Grading: Each homework will be graded on the scale 100. Unless otherwise noted, the questions/parts will be weighed equal.

Part 1. In the graph below you see the possible flights between some of the cities in Turkey. Write the predicate "route(X,Y) – a route between X and Y exists" that checks if there is a route between any given two cities.



Your program should have all the facts and predicates/rules. See the following:

```
% knowledge base
…
flight(istanbul,antalya).  % the fact that Istanbul and Antalya has a flight.
…
% rules
…
```

```
route(X,Y) :- flight(X,Y). % a predicate indicating there exist a route between
                           % X and Y if there is flight between X and Y.
…
```

A single query to complete your program should check if there is a direct route between two given cities. Alternatively, it can list all the connected cities for a given city. See the following:

```
?- route(edirne,X).
X = erzincan ;
X = edremit ;
```

Part 2. Continuing with the previous problem, you are asked to write a program that checks if a route exists between two cities and if so provides the shortest route.

In the first step, you are to expand the knowledge by adding distances for the flights. E.g.,

```
% knowledge base
…
flight(istanbul, antalya).  % the fact that Istanbul and Antalya has a flight.
distance(istanbul, antalya, 481).  % flight distance – calculated using
                                   % https://www.distancecalculator.net
% complete all the flights and distances …
…
```

A single query to complete your program should check if there is a direct route between two given cities and the shortest distance between them. See the following:

```
?- sroute(edremit,erzincan,X).
X = 1044 ;
```

Part 3. You are given the following database about classes, classrooms and student enrollment.

Classes		
Class	Time	Room
102	10	z23
108	12	z11
341	14	z06
455	16	207
452	17	207

Enrollment	
Student	Class
a	102
a	108
b	102
c	108
d	341
e	455

Write the predicates “when(X,Y) – time of the course X is Y”, “where(X,Y) – place of the course X is Y”, and “enroll(X,Y) – student X is enrolled in course Y”. For example:

```
% facts..
when(102,10).
```

3.1. Define/write a predicate “schedule(S,P,T)” associates a student to a place and time of class. See the example query and its result.

```
?- schedule(a,P,T).
P = 102
T = 10 ;

P = 108
T = 12 ;
```

3.2. Define/write another predicate “usage(P,T)” that gives the usage times of a classroom. See the example query and its result.

```
?- usage(207,T).
T = 455 ;
T = 456 ;
```

3.3. Define/write another predicate “conflict(X,Y)” that gives true if X and Y conflicts due to classroom or time.

3.4. Define/write another predicate “meet(X,Y)” that gives true if student X and student Y are present in the same classroom at the same time.

Part 4. Write the following predicates operating on lists.

4.1. Define a Prolog predicate “add(L,S)” that adds all the elements of L binding the results to S.

4.2. Define a Prolog predicate “unique(L1,L2)” that removes duplicates in L1 binding the results to L2.

4.3. Define a Prolog predicate “flatten(L1,L2)” that flattens the list of lists (your can assume that elements of L1 are not nested lists) in L1 binding results to L2.