Assignment 1:
“Prolog”

Mohamed Ali Ramadan

7688825

CSI 2120 – Programming Paradigms

Dr. Lang

Date of Submission: February 26th, 2018

University of Ottawa

Assignment 1: Prolog

Question 1

a) findall(Employee,(employee(Employee,Company), company(Company,City), not(person(Employee,City))),L).

b) findall(X,(company(X,Y),city(Y,ontario)),L).

c) findall(X,(person(X,City),not(employee(X,Company))),L).

d) findall(X,(employee(X,C),company(C,ottawa)),L).

e) setof(X,Y^(employee(X,Y),company(Y,ottawa)),L).

The previous result list of part (d) contains the name of Marie twice because the findall/3 predicate does not eliminate duplicates like the setof/3 predicate, and Marie is a duplicate in this query because there are two instances of the name Marie who are employed and are both employed by two companies in Ottawa.

Question 2

Please open separate image: "*A1 Question 2.png*".

Question 3

Please view file *Q3.pl* for testing:

```
distance(Lat1Deg,Lon1Deg,Lat2Deg,Lon2Deg,D) :-
        Lat1Rad is (pi*(Lat1Deg/180)),
        Lon1Rad is (pi*(Lon1Deg/180)),
        Lat2Rad is (pi*(Lat2Deg/180)),
        Lon2Rad is (pi*(Lon2Deg/180)),
        A is ((sin((Lat1Rad-Lat2Rad)/2))**2),
        C is ((cos(Lat1Rad))*(cos(Lat2Rad))),
        Dx is ((sin((Lon1Rad-Lon2Rad)/2))**2),
        B is (C*Dx),
        DRad is (2*asin(sqrt(A+B))),
        D is (DRad*6371).
```

Question 4 (View *Q4A.pl* and *Q4B.pl*)

A)

```
absDiffA(A,B,Result) :- absDiffA(A,B,[],Result). %Helper

absDiffA([],[],Result,Reverse) :- mirrorAcc(Result,Reverse),!. %Boundary case

absDiffA([AH|AT],[BH|BT],SoFar,Result) :-
        Temp is (abs(AH-BH)),
        SoFar2 = [Temp|SoFar],
        absDiffA(AT,BT,SoFar2,Result).

absDiffA([AH|AT],[],SoFar,Result) :-
        Temp is (abs(AH)),
        SoFar2 = [Temp|SoFar],
        absDiffA(AT,[],SoFar2,Result).

absDiffA([],[BH|BT],SoFar,Result) :-
        Temp is (abs(BH)),
        SoFar2 = [Temp|SoFar],
        absDiffA([],BT,SoFar2,Result).

reverseList([],L,L) :- !.
reverseList([H|T],L,R) :- reverseList(T,[H|L],R).
mirrorAcc(L,R) :- reverseList(L,[],R).
```

B)

```
absDiffB(A,B,Result) :- absDiffB(A,B,[],Result). %Helper

absDiffB([],[],Result,Reverse) :- mirrorAcc(Result,Reverse),!. %Boundary case

absDiffB([AH|AT],[BH|BT],SoFar,Result) :-
        Temp is (abs(AH-BH)),
        SoFar2 = [Temp|SoFar],
        absDiffB(AT,BT,SoFar2,Result).

absDiffB([AH|AT],[],SoFar,Result) :-
        absDiffB([],[],SoFar,Result), !.

absDiffB([],[BH|BT],SoFar,Result) :-
        absDiffB([],[],SoFar,Result), !.

reverseList([],L,L) :- !.
reverseList([H|T],L,R) :- reverseList(T,[H|L],R).
mirrorAcc(L,R) :- reverseList(L,[],R).
```

Question 5

Please view file *Q5.pl* for testing:

```
bouquet(L) :-
setof(([F1,C1],[F2,C2],[F3,C3]),(flower(F1,C1),flower(F2,C2),flower(F3,C3),F1\==F2,F1\==F3,
F2\==F3,C1==red, C2==red),L), !.
```

```
bouquet(L) :-
setof(([F1,C1],[F2,C2],[F3,C3]),(flower(F1,C1),flower(F2,C2),flower(F3,C3),F1\==F2,F1\==F3,
F2\==F3,C1\==C2,C1\==C3,C2\==C3),L).
```