**Part 1**

**4)**

**a)**

Find the fathers -> parent(X, Someone), male(X)

Find the sons -> parent(X, Someone), male(Someone)

Find the daughters -> parent(X, Someone), female(Someone)

**b)**

They both gave the same answers. The only difference is that the second one, ended with a false.

**c)**

Find the grandfathers -> parent(X, Someone), parent(GP, X), male(GP).

Find the grandmothers -> parent(X, Someone), parent(GP, X), female(GP).

Find the great-great grandparents -> parent(X, Someone), parent(GP, X), parent(GGP,GP), parent(GGGP,GGP).

**5)**

Fathers -> father(X, Someone) :- parent(X, Someone), male(X).

Sons -> sons(X, Someone) :- parent(X, Someone), male(Someone).

Daughters -> daughters(X, Someone) :- parent(X, Someone), female(Someone).

Grandfathers -> grandfathers(X, Someone, GP) :- parent(X, Someone), parent(GP,X), male(GP).

Grandmothers -> grandmothers(X, Someone, GP) :- parent(X, Someone), parent(GP,X), female(GP).

Great-Great Grandparents -> great-great-grandparents(X, Someone, GP) :- parent(X, Someone), parent(GP,X), parent(GGP,GP), parent(GGGP,GGP).

**6)**

Siblings -> sibling(X,Y):- parent(M,Y),female(M),parent(F,Y),male(F),parent(M,X),parent(F,X),X\==Y.

Brothers -> brother(X,Y) :- male(X), sibling(X,Y).

Sisters -> sister(X,Y) :- female(X), sibling(X,Y).

Uncles -> uncle(X,Y) :- male(X), sibling(X,Someone), parent(Someone, Y).

Aunts -> aunt(X,Y) :- female(X), sibling(X,Someone), parent(Someone, Y).

Cousins -> cousins(X,Y) :- parent(S1,X), parent(S2,Y), sibling(S1,S2).

**7)**

Blood Uncle -> blood\_uncle(X,Y) :- male(X), parent(Someone, Y), brother(X,Someone).

Blood Aunt -> blood\_aunt(X,Y) :- female(X), parent(Someone, Y), sister(X,Someone).

Marriage Uncle -> marriage\_uncle(X,Y) :- male(X), parent(Someone, Y), sister(W,Someone), married(X,W).

Marriage Aunt -> marriage\_aunt(X,Y) :- female(X), parent(Someone, Y), brother(W,Someone), married(X,W).

**8)**

grandparents(X,Y) :- parent(S1, X), parent(X, S2).

grandchild(X,Y) :- parent(S1, X), parent(Y, S2).

**9)**

ancestor(X,Y) :- parent(X,Z), ancestor(Z,Y).

**10)**

descendant(X,Y):- parent(Y,S1),descendant(X,S1).

**Part 2**

airport\_hop(A1, A2) :- hop(\_,A1,A2,\_,\_).

city\_hop(C1,C2):- hop(\_,S1,S2,\_,\_), city(S1,C1,\_),city(S2,C2,\_).

two\_hop(A1,A2) :- flight(\_,A1,S1,\_,\_), flight(\_,S1,A2,\_,\_), A1\==A2.

time\_diff(T1,T2,T) :- T1 = time(X,Y), T2 = time(W,Z), T = time(K,I), I is Z-Y, K is W-X.

duration(A1,A2,T,F):- airport\_hop(A1,A2), hop(S1,A1,A2,T1,T2), time\_diff(T1,T2,T).

no\_layover(A1,A2):- two\_hop(A1,A2), hop(\_,A1,S1,T1,\_), hop(\_,S1,A2,T2,\_), time\_diff(T1,T2,T), T = time(T3,\_), T3 < 2.

direct(A1,A2,F):- hop(F,A1,A2,\_,\_).

**Part 3**

name('Rui Carapinha').

name('Luis Carapinha').

name('Vadson Culanda').

name('Antonio Carapinha').

age(20, 'Rui Carapinha').

age(24, 'Luis Carapinha').

age(21, 'Vadson Culanda').

age(47, 'Antonio Carapinha').

profession('Student', 'Rui Carapinha').

profession('3D Modeler', 'Luis Carapinha').

profession('Student', 'Vadson Culanda').

profession('Maintenance', 'Antonio Carapinha').

country('Portugal', 'Rui Carapinha').

country('Portugal', 'Luis Carapinha').

country('Angola', 'Vadson Culanda').

country('Portugal', 'Antonio Carapinha').

city('Ovar', 'Rui Carapinha').

city('Ovar', 'Luis Carapinha').

city('Aveiro', 'Vadson Culanda').

city('Ovar', 'Antonio Carapinha').

home\_university('Universidade De Aveiro', 'Rui Carapinha').

home\_university('Universidade De Aveiro', 'Vadson Culanda').

home\_university('Universidade De Aveiro', 'Antonio Carapinha').

away\_university('Politechnika Wroclawska', 'Rui Carapinha').

away\_university('Politechnika Wroclawska', 'Vadson Culanda').

siblings(1, 'Rui Carapinha').

siblings(1, 'Luis Carapinha').

siblings(4, 'Antonio Carapinha').

siblings(5, 'Vadson Culanda').

current('Erasmus', 'Rui Carapinha').

current('Erasmus', 'Vadson Culanda').

current('Working', 'Luis Carapinha').

current('Working', 'Antonio Carapinha').

birthday('23', 'December', 'Rui Carapinha').

birthday('25', 'November', 'Vadson Culanda').

birthday('31', 'March', 'Luis Carapinha').

birthday('31', 'December', 'Antonio Carapinha').

**a)**

who(X,Y,Z):- name(X), age(Y, X), profession(Z, X).

**b)**

whoInErasmus(X) :- name(X), current(Y,X), Y == 'Erasmus'.

older(X) :- age(S1,X), not(( age(S2,\_), S2 > S1 )).

moreSiblings(X) :- siblings(S1,X), not(( siblings(S2,\_), S2 > S1 )).

liveInOvar(X) :- city(X,Y), X == 'Ovar'.

birthdayInDecember(X) :- birthday(\_,S1,X), S1 == 'December'.