

**Ahsanullah University of Science & Technology**

**Department of Computer Science & Engineering**

**Course No                    : CSE4108**

**Course   Title           : Artificial Intelligence Lab**

**Assignment   No : 01**

**Date of Submission    : 17/2/2020**

**Submitted To            : Dr. S.M.A. Al-Mamun**

**Tonmoy Hossain**

**Submitted By-**

**Name : Devopriya Tirtho**

**Id : 16.02.04.033**

**Section : A2**

**1. Modify the Python and Prolog codes demonstrated above to find the grandparents of somebody**

**Python Code:**

tupleList1=[('child','Rakib','Hasib'),('child','Sohel','Rakib'),('child','Rebeka','Rakib'),('child','Hasib','Rashid')]

X=str(input("Grandchild:"))

print('Grandparent:',end=' ')

i=0

while(i<=3):

if((tupleList1[i][0]=='child')&(tupleList1[i][1]==X)):

for j in range (4):

if((tupleList1[i][0]=='child')&(tupleList1[i][2]==tupleList1[j][1])):

print(tupleList1[j][2],end=' ')

i=i+1

**Prolog Code:**

parent('Hasib' , 'Rakib'). parent('Rakib' , 'Sohel'). parent('Rakib' , 'Rebeka').

parent('Rashid' , 'Hasib'). grandchild(Z,X) :- parent(X, Y), parent(Y, Z).

findGp :- write(' Grandchild: '), read(X), write('Grandparent: '),

grandparent(Z, Gp), write(Gp), tab(5), fail.

findGp.

**Lab Exercise: 2(a)**

**Enrich the KB demonstrated above with ‘brother’, ‘sister’ rules in Python and Prolog.**

**Object relationships as a KB:**

**Hasib is a parent of Rakib. Rakib is a parent of Sohel. Rakib is a parent of Rebeka. Hasib is a parent of Sakib.**

**Python Code:**

tupleList1=[('parent', 'Hasib', 'Rakib','male'),('parent', 'Hasib', 'Sakib','male'),('parent', 'Rakib', 'Rebeka','female'),('parent', 'Rakib', 'Sohel','male')]

X=str(input("Name:"))

print('Brother/Sister:', end=' ')

i=0

while(i<=3):

if ((tupleList1[i][0] == 'parent')&( tupleList1[i][2] == X)):

for j in range(4):

if ((tupleList1[j][0] == 'parent') & ( tupleList1[i][1] == tupleList1[j][1])&( tupleList1[j][3] == 'male')):

print(tupleList1[j][2], end=' Brother ')

elif ((tupleList1[j][0] == 'parent') & ( tupleList1[i][1] == tupleList1[j][1])&( tupleList1[j][3] == 'female')):

print(tupleList1[j][2], end=' Sister ')

i=i+1

**Prolog Code to Find Brother:**

parent('Hasib' , 'Rakib'). parent('Rakib' , 'Sohel'). parent('Rakib' , 'Rebeka'). parent('Rashid' , 'Hasib').

male('Hasib'). male('Rakib'). male('Sohel'). female('Rebeka'). brother(Y, Z) :- parent(X, Y), parent(X, Z),male(Z),not(Y=Z).

findBro :- write(' Name: '), read(Y), write('Brother: '),

brother(Y, Bro), write(Bro), tab(5), fail.

findBro.

**Prolog Code to Find Sister:**

parent('Hasib' , 'Rakib'). parent('Rakib' , 'Sohel'). parent('Rakib' , 'Rebeka'). parent('Rashid' , 'Hasib').

male('Hasib'). male('Rakib'). male('Sohel'). female('Rebeka'). sister(Y, Z) :- parent(X, Y), parent(X, Z),female(Z),not(Y=Z).

findSis :- write(' Name: '), read(Y), write('Sister: '),

brother(Y, Sis), write(Sis), tab(5), fail.

findSis.

**Lab Exercise: 2(b)**

**Enrich the KB demonstrated above with ‘Uncle’, ‘Aunt’ rules in Python and Prolog.**

**Object relationships as a KB:**

**Hasib is a parent of Rakib. Rakib is a parent of Sohel. Rakib is a parent of Rebeka. Hasib is a parent of Sakib.**

**Python Code:**

tupleList1=[('parent', 'Hasib', 'Rakib','male'),('parent', 'Hasib', 'Sakib','male'),('parent', 'Rakib', 'Rebeka','female'),('parent', 'Rakib', 'Sohel','male')

X=str(input("Name:"))

i=0

while(i<=3):

if ((tupleList1[i][0] == 'parent')&( tupleList1[i][2] == X)):

Y=tupleList1[i][1]

i=i+1

print('Uncle/Aunt:', end=' ')

m=0

while(m<=3):

if ((tupleList1[m][0] == 'parent')&( tupleList1[m][2] == Y)):

for j in range(4):

if ((tupleList1[j][0] == 'parent') & ( tupleList1[m][1] == tupleList1[j][1])&( tupleList1[m][3] == 'male')):

print(tupleList1[j][2], end=' Uncle')

elif ((tupleList1[j][0] == 'parent') & ( tupleList1[m][1] == tupleList1[j][1])&( tupleList1[m][3] == 'female')):

print(tupleList1[j][2], end=' Aunt')

m=m+1

**Prolog Code for Uncle:**

parent('Hasib' , 'Rakib'). parent('Rakib' , 'Sohel'). parent('Rakib' , 'Rebeka').

parent('Hasib' , 'Sakib'). male('Hasib'). male('Rakib'). male('Sohel'). male('Sakib'). female('Rebeka'). uncle(M, Z) :-

parent(X, M), parent(Y, X), parent(Y, Z), male(Z), not(X=Z).

findUn :- write(' Name: '), read(M), write('Uncle: '),

uncle(M, Un), write(Un), tab(5), fail.

findUn.

**Prolog Code for Aunt:**

parent('Hasib' , 'Rakib'). parent('Rakib' , 'Sohel'). parent('Rakib' , 'Rebeka').

parent('Hasib' , 'Sakib'). male('Hasib'). male('Rakib'). male('Sohel'). male('Sakib'). female('Rebeka'). aunt(M, Z) :-

parent(X, M), parent(Y, X), parent(Y, Z), female(Z), not(X=Z).

findAn :- write(' Name: '), read(M), write('Aunt: '),

aunt(M, An), write(An), tab(5), fail.

findAn.