

Ahsanullah University of Science And Technology

Department of Computer Science & Engineering



CSE4108

Artificial Intelligence

Submitted By:

Mayisha Farzana

16.02.04.028

Date of Submission: **24 February, 2020**

Q. Enrich the KB demonstrated above with 'brother', 'sister', 'uncle' and 'aunt' rules in Python and Prolog.

Python code:

```
tupleList1=[('parent', 'Habib', 'Rana'),
             ('parent', 'Habib', 'Panna'),
             ('parent', 'Habib', 'Shova'),
             ('parent', 'Habib', 'Ratna'),
             ('parent', 'Habib', 'Shovon'),
             ('parent', 'Panna', 'Zahin'),
             ('parent', 'Panna', 'Labiba'),
             ('parent', 'Rana', 'Mashrur'),
             ('parent', 'Rana', 'Mayisha')]

Female=['Shova', 'Ratna', 'Labiba', 'Mayisha']
#find Sister
X=str(input("Name:"))
print('Sister:', end=' ')
flag=0
i=0
while(i<=8):
    if ((tupleList1[i][0] == 'parent')&(tupleList1[i][2] == X)):
        for j in range(9):
            if ((tupleList1[j][0] == 'parent')&(tupleList1[i][1] == tupleList1[j][1]) &( tupleList1[i][2] !=
tupleList1[j][2])):
                for k in Female:
                    if(tupleList1[j][2] == k):
                        print(tupleList1[j][2], end=' ')

        i=i+1

#find Brother
tupleList1=[('parent', 'Habib', 'Rana'),
             ('parent', 'Habib', 'Panna'),
             ('parent', 'Habib', 'Shova'),
             ('parent', 'Habib', 'Ratna'),
             ('parent', 'Habib', 'Shovon'),
             ('parent', 'Panna', 'Zahin'),
             ('parent', 'Panna', 'Labiba'),
             ('parent', 'Rana', 'Mashrur'),
             ('parent', 'Rana', 'Mayisha')]

Female=['Shova', 'Ratna', 'Labiba', 'Mayisha']
X=str(input("Name:"))
print('Brother:', end=' ')
i=0
flag = 0
while(i<=8):
    if ((tupleList1[i][0] == 'parent')&(tupleList1[i][2] == X)):
        for j in range(9):
            if ((tupleList1[j][0] == 'parent')&(tupleList1[i][1] == tupleList1[j][1]) &(tupleList1[i][2] !=
tupleList1[j][2])):
                flag = 0
                for k in Female:
```

```

        if(tupleList1[j][2] == k):
            flag=1
        if(flag==0):
            print(tupleList1[j][2], end=' ')

    i=i+1

#find Uncle

tupleList1= [('parent', 'Habib', 'Rana'),
              ('parent', 'Habib', 'Panna'),
              ('parent', 'Habib', 'Shova'),
              ('parent', 'Habib', 'Ratna'),
              ('parent', 'Habib', 'Shovon'),
              ('parent', 'Panna', 'Zahin'),
              ('parent', 'Panna', 'Labiba'),
              ('parent', 'Rana', 'Mashrur'),
              ('parent', 'Rana', 'Mayisha')]

Female=['Shova', 'Ratna', 'Labiba', 'Mayisha']
X=str(input("Name:"))
print('Uncle:', end=' ')
flag=0
i=0
Z = ""
while(i<=8):
    if ((tupleList1[i][0] == 'parent')&(tupleList1[i][2] == X)):
        Y = tupleList1[i][1]
        for j in range(8):
            if ((tupleList1[j][0] == 'parent')&(tupleList1[i][1] == tupleList1[j][2])):
                Z = tupleList1[j][1]
        i=i+1

m=0
while(m<=8):
    if ((tupleList1[m][0] == 'parent')&(tupleList1[m][1] == Z)):
        if(tupleList1[m][2] != Y):
            flag = 0
            for k in Female:
                if(tupleList1[m][2] == k):
                    flag=1
            if(flag==0):
                print(tupleList1[m][2], end=' ')

    m=m+1

#find Aunt
tupleList1= [('parent', 'Habib', 'Rana'),
              ('parent', 'Habib', 'Panna'),
              ('parent', 'Habib', 'Shova'),
              ('parent', 'Habib', 'Ratna'),
              ('parent', 'Habib', 'Shovon'),
              ('parent', 'Panna', 'Zahin'),
              ('parent', 'Panna', 'Labiba'),
              ('parent', 'Rana', 'Mashrur'),

```

```

('parent', 'Rana', 'Mayisha')]

Female=['Shova', 'Ratna', 'Labiba', 'Mayisha']
X=str(input("Name:"))
print('Aunt:', end=' ')
flag=0
i=0
Z = "";
while(i<=8):
    if ((tupleList1[i][0] == 'parent')&(tupleList1[i][2] == X)):
        Y = tupleList1[i][1]
        for j in range(8):
            if ((tupleList1[j][0] == 'parent')&(tupleList1[j][1] == tupleList1[i][2])):
                Z = tupleList1[j][1]
        i=i+1

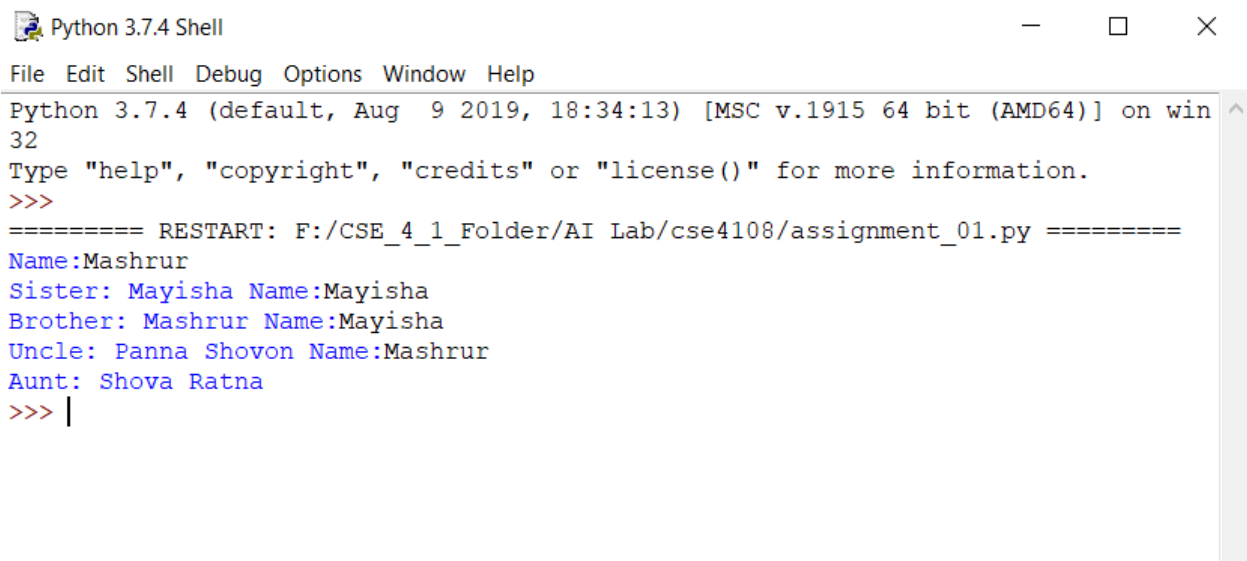
m=0
while(m<=8):

    if ((tupleList1[m][0] == 'parent')&(tupleList1[m][1] == Z)):
        if(tupleList1[m][2] != Y):
            for k in Female:
                if(tupleList1[m][2] == k):
                    print(tupleList1[m][2], end=' ')

    m=m+1

```

OUTPUT:



```

Python 3.7.4 Shell
File Edit Shell Debug Options Window Help
Python 3.7.4 (default, Aug  9 2019, 18:34:13) [MSC v.1915 64 bit (AMD64)] on win
32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: F:/CSE_4_1_Folder/AI Lab/cse4108/assignment_01.py =====
Name:Mashrur
Sister: Mayisha Name:Mayisha
Brother: Mashrur Name:Mayisha
Uncle: Panna Shovon Name:Mashrur
Aunt: Shova Ratna
>>> |

```

Prolog:

```
parent('Habib' , 'Rana').
parent('Habib' , 'Panna').
parent('Habib' , 'Shova').
parent('Habib' , 'Ratna').
parent('Habib' , 'Shovon').
parent('Panna' , 'Zahin').
parent('Panna' , 'Labiba').
parent('Rana' , 'Mashrur').
parent('Rana' , 'Mayisha').
female('Shova'). female('Ratna'). female('Labiba'). female('Mayisha').
sister(X, Y) :- parent(Z, X), parent(Z, Y), not(X=Y), female(X).
brother(X,Y):-parent(Z,X),parent(Z,Y), not(X=Y), not( female(X)).
uncle(P,M):-parent(R,M),parent(Z,R),parent(Z,P), not(R=P) ,not (female(P)).
aunt(L,M):-parent(R,M),parent(Z,L),parent(Z,R), not(L=R) ,female(L).
```

/* Procedure to find the SISTER of X*/

```
findS :- write(' Name: '), read(X), write('Sister: '), sister(S,X), write(S), tab(5), fail.
findS.
```

/* Procedure to find the Brother of X*/

```
findB:-write('Name: '),read(X),write('Brother: '),brother(B,X),write(B),tab(5),fail.
findB.
```


/*Procedure to find the Uncle of P*/

```
findU:-write('Name'),read(P),write('Uncle'),uncle(U,P),write(U),tab(5),fail.
findU.
```

/*Procedure to find the Aunt of L*/

```
findA:-write('Name '),read(L),write('Aunt '),aunt(A,L),write(A),tab(5),fail.
findA.
```

OUTPUT:

 SWI-Prolog (Multi-threaded, version 6.4.0)

File Edit Settings Run Debug Help

```
% library(win_menu) compiled into win_menu 0.00 sec, 33 clauses
Welcome to SWI-Prolog (Multi-threaded, 32 bits, Version 6.4.0)
Copyright (c) 1990-2013 University of Amsterdam, VU Amsterdam
SWI-Prolog comes with ABSOLUTELY NO WARRANTY. This is free software
and you are welcome to redistribute it under certain conditions.
Please visit http://www.swi-prolog.org for details.
```

For help, use `?- help(Topic).` or `?- apropos(Word).`

```
1 ?-
% f:/cse_4_1_folder/ai lab/cse4108/assignment_01 compiled 0.00 sec,
1 ?- findS.
    Name: 'Mashrur'.
    Sister: Mayisha
true.

2 ?- findB.
    Name: 'Mayisha'.
    Brother: Mashrur
true.

3 ?- findA.
    Name: 'Mayisha'.
    Aunt Shova      Ratna
true.

4 ?- findU.
    Name: 'Mashrur'.
    UnclePanna      Shovon
true.
```