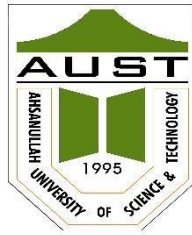


AHSANULLAH UNIVERSITY OF SCIENCE AND TECHNOLOGY
DHAKA-1208, BANGLADESH.



Department of Computer Science and Engineering
Spring 2019

Program: Bachelor of Science in Computer Science and Engineering

Course No: CSE 4108

Course Title: Artificial Intelligence Lab

Assignment No: 01

Date of Submission: 22/07/2019

Submitted to Dr. S.M. Abdullah Al-Mamun
Professor, Department of CSE, AUST.

Md. Siam Ansary
Adjunct Faculty, Department of CSE, AUST.

Submitted By

Name : Robiul Hasan Nowshad
Student ID : 16.01.04.061
Lab Group : B1

Q#1: Modify the demonstrated Python and Prolog codes to find the grandparents of somebody.

Solution:

Python:

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

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

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

```
i=0
```

```
while(i<=3):
```

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

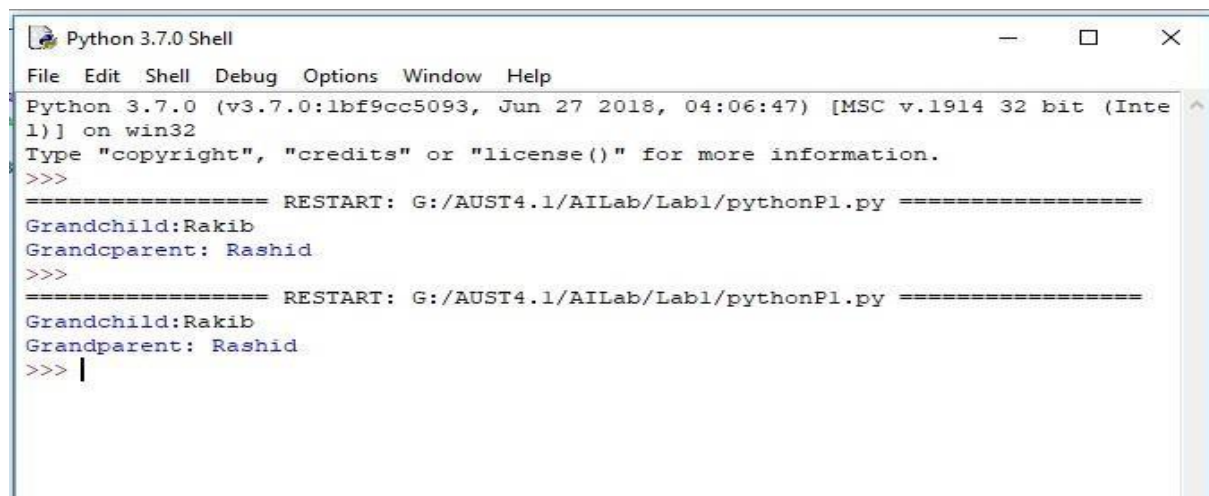
```
        for j in range(4):
```

```
            if ((tupleList1[j][0] == 'parent') & (
```

```
tupleList1[i][1] == tupleList1[j][2])):
```

```
                print(tupleList1[j][1], end=' ')
```

```
        i=i+1
```



```
Python 3.7.0 Shell
File Edit Shell Debug Options Window Help
Python 3.7.0 (v3.7.0:1bf9cc5093, Jun 27 2018, 04:06:47) [MSC v.1914 32 bit (Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: G:/AUST4.1/AILab/Lab1/pythonP1.py =====
Grandchild:Rakib
Grandparent: Rashid
>>>
===== RESTART: G:/AUST4.1/AILab/Lab1/pythonP1.py =====
Grandchild:Rakib
Grandparent: Rashid
>>> |
```

Prolog:

```
parent('Hasib', 'Rakib'). parent('Rakib', 'Sohel'). parent('Rakib', 'Rebeka'). parent('Rashid'  
, 'Hasib'). grandparent(X, Z) :- parent(X, Y), parent(Y, Z).
```

```
findGp :- write(' Grandchild: '), read(Z), write('Grandparent: '),
```

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

```
findGp.
```

```
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 ?-
% g:/AUST4_1/AI/Lab/Lab1/prologP1.pl compiled 0.00 sec, 8 clauses
1 ?- findGp.
Grandchild: 'Rakib'.
Grandparent: Rashid
true.
2 ?- █
```

Q#2: Enrich the demonstrated knowledge base with 'brother', 'sister', 'uncle' and 'aunt' rules in Python and Prolog.

Solution:

Prolog:

```
parent('Rashid', 'Hasib'). parent('Hasib', 'Rakib'). parent('Hasib', 'Selina'). parent('Hasib', 'Jubayer').
parent('Rakib', 'Sohel'). parent('Selina', 'Trina'). parent('Jubayer', 'Shahi'). parent('Jubayer',
'Barsha'). male('Rashid'). male('Hasib'). male('Rakib'). male('Jubayer'). male('Sohel').
male('Shahi').
```

```
brother(X,Y):-parent(Z,X), parent(Z,Y), male(Y), not(X=Y). sister(X,Y):-parent(Z,X),
parent(Z,Y), not(male(Y)), not(X=Y).
```

```
uncle(X,Y):-parent(Z,X), parent(W,Z), parent(W,Y), male(Y), not(X=Y).
```

```
aunt(X,Y):-parent(Z,X), parent(W,Z), parent(W,Y), not(male(Y)), not(X=Y).
```

```
findBr :- write(' Person: '), read(X), write(' Brother: '),
          brother(X,Br), write(Br), tab(5), fail.
```

```
findBr.
```

```
findSs :- write(' Person: '), read(X), write(' Sister: '),
          sister(X,Ss), write(Ss), tab(5), fail.
```

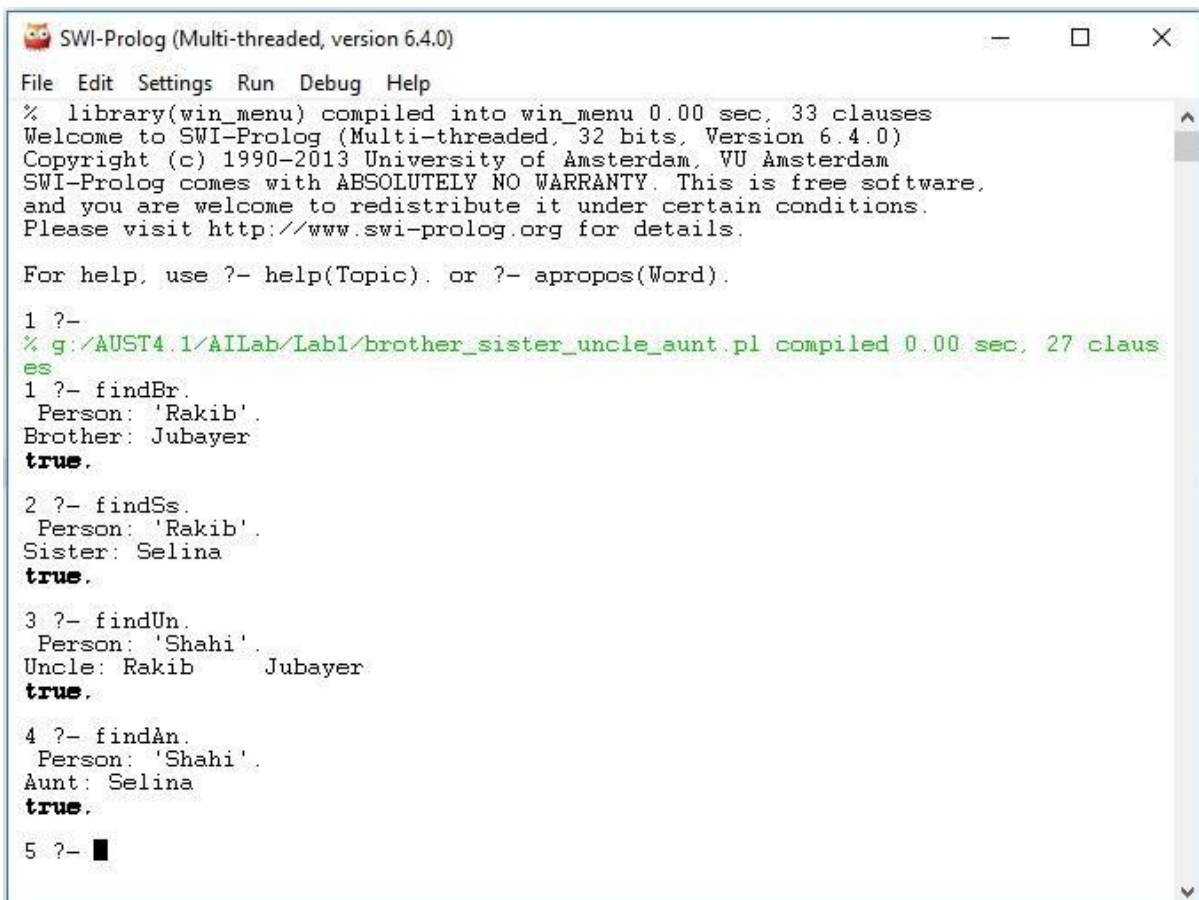
```
findSs.
```

```
findUn :- write(' Person: '), read(X), write('Uncle: '),  
         uncle(X, Un), write(Un), tab(5), fail.
```

```
findUn.
```

```
findAn :- write(' Person: '), read(X), write('Aunt: '),  
         aunt(X, An), write(An), tab(5), fail.
```

```
findAn.
```



```
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 ?-  
% g:/AUST4.1/AILab/Lab1/brother_sister_uncle_aunt.pl compiled 0.00 sec, 27 clauses  
1 ?- findBr.  
   Person: 'Rakib'.  
   Brother: Jubayer  
   true.  
  
2 ?- findSs.  
   Person: 'Rakib'.  
   Sister: Selina  
   true.  
  
3 ?- findUn.  
   Person: 'Shahi'.  
   Uncle: Rakib    Jubayer  
   true.  
  
4 ?- findAn.  
   Person: 'Shahi'.  
   Aunt: Selina  
   true.  
  
5 ?- █
```

Python:

```
a) Find Brother- tupleList1 = [('parent','Rashid','Hasib'), ('parent','Hasib','Rakib'),  
                               ('parent','Rakib','Sohel'),  
                               ('parent','Rakib','Rebeka')]  
tupleList2 = [('Rashid','M'),('Hasib','M'),('Rakib','M'),('Sohel','M'),('Rebeka','F')] found  
= 0;  
X = str(input('Person:'))  
print("Brother:",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][2]!=X)):
for k in range(5):          if((tupleList2[k][0] == tupleList1[j][2]) & (tupleList2[k][1] == 'M')):
print(tupleList1[j][2])
found = 1

```

```

i=i+1  if(found
== 0):
print('No brother')

```

b) Find Sister- tupleList1 = [('parent','Rashid','Hasib'), ('parent','Hasib','Rakib'), ('parent','Rakib','Sohel'), ('parent','Rakib','Rebeka')]

```

tupleList2 = [('Rashid','M'),('Hasib','M'),('Rakib','M'),('Sohel','M'),('Rebeka','F')] found
= 0;
X = str(input('Person:')) print("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][2]!=X)):
for k in range(5):          if((tupleList2[k][0] == tupleList1[j][2]) & (tupleList2[k][1] == 'F')):
print(tupleList1[j][2])
found = 1

i=i+1  if(found
== 0):  print('No
Sister')

```

c) Find Uncle- tupleList1 = [('parent','Rashid','Hasib'), ('parent','Hasib','Rakib'), ('parent','Hasib','Jobayer'), ('parent','Rakib','Sohel'), ('parent','Rakib','Rebeka'), ('parent','Jobayer','Selina'), ('parent','Jobayer','Imtiyaz')]

```

tupleList2=[('Rashid','M'),('Hasib','M'),('Rakib','M'),('Jobayer','M'),('Sohel','M'),('Rebeka','F'),
('Selina','F'),('Imtiyaz','M')] def
find_uncle(X):

found = 0;
i = 0
while(i<=6):
if((tupleList1[i][0] == 'parent')&(tupleList1[i][2] == X)):
for j in range(7):
if((tupleList1[j][0]=='parent')&(tupleList1[i][1]==tupleList1[j][1])&(tupleList1[j][2]!=X)):
for k in range(8):
if((tupleList2[k][0] == tupleList1[j][2]) & (tupleList2[k][1] ==
'M')):

return tupleList1[j][2]
found = 1

```

```

    i=i+1    if(found ==
0):    return 'No
brother' found = 0

```

```

X = str(input('Person:'))

```

```

print("Uncle:",end="")

```

```

i = 0 while(i<=6):

```

```

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

```

```

if(find_uncle(tupleList1[i][1]) != 'No brother'):

```

```

    print(find_uncle(tupleList1[i][1]))

```

```

found = 1

```

```

    i = i+1    if(found

```

```

== 0):    print('No

```

```

Uncle')

```

d) Find Aunt- tupleList1 = [('parent','Rashid','Hasib'), ('parent','Hasib','Rebeka'), ('parent','Hasib','Selina'), ('parent','Rebeka','Sohel'), ('parent','Rebeka','Rakib'), ('parent','Selina','Jobayer'), ('parent','Selina','Imtiyaz')]
 tupleList2=[('Rashid','M'),('Hasib','M'),('Rakib','M'),('Jobayer','M'),('Sohel','M'),('Rebeka','F'), ('Selina','F'),('Imtiyaz','M')]

```

def find_aunty(X):

```

```

    found = 0;

```

```

    i = 0

```

```

while(i<=6):

```

```

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

```

```

        for j in range(7):

```

```

            if((tupleList1[j][0]=='parent')&(tupleList1[i][1]==tupleList1[j][1])&(tupleList1[j][2]!=X:

```

```

                for k in range(8):                    if((tupleList2[k][0] ==

```

```

tupleList1[j][2]) & (tupleList2[k][1] == 'F')):

```

```

                return tupleList1[j][2]

```

```

                found = 1

```

```

i=i+1    if(found == 0):

```

```

    return 'No Sister'

```

```

found = 0

```

```

X = str(input('Person:'))

```

```

print("Aunty:",end="") i = 0 while(i<=6):

```

```

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

```

```

if(find_aunty(tupleList1[i][1]) != 'No Sister'):

```

```
    print(find_aunty(tupleList1[i][1]))  
found = 1
```

```
    i = i+1    if(found  
== 0):  
    print('No Aunty')
```



```
Python 3.7.0 Shell  
File Edit Shell Debug Options Window Help  
Python 3.7.0 (v3.7.0:1bf9cc5093, Jun 27 2018, 04:06:47) [MSC v.1914 32 bit (Intel)] on win32  
Type "copyright", "credits" or "license()" for more information.  
>>>  
===== RESTART: G:\AUST4.1\AILab\Lab1\Find_Brother.py =====  
Person:Rebeka  
Brother:Sohel  
>>>  
===== RESTART: G:\AUST4.1\AILab\Lab1\Find_Sister.py =====  
Person:Sohel  
Sister:Rebeka  
>>>  
===== RESTART: G:\AUST4.1\AILab\Lab1\Find_Uncle.py =====  
Person:Imtiyaz  
Uncle:Rakib  
>>>  
===== RESTART: G:\AUST4.1\AILab\Lab1\Find_aunty.py =====  
Person:Sohel  
Aunty:Selina  
>>>
```