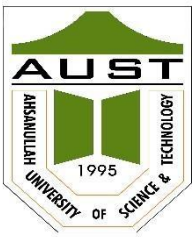
AHSANULLAH UNIVERSITY OF SCIENCE AND TECHNOLOGY
DHAKA-1208, BANGLADESH.



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

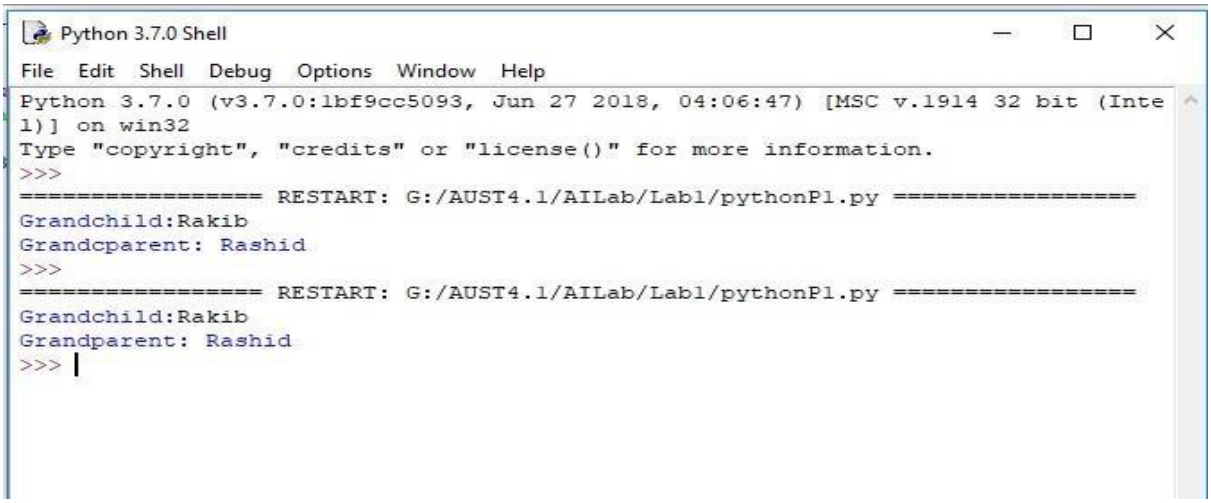
Name : Robiul Hasan Nowshad
Student ID : 16.01.04.061
Lab Group : B1

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

Solution:

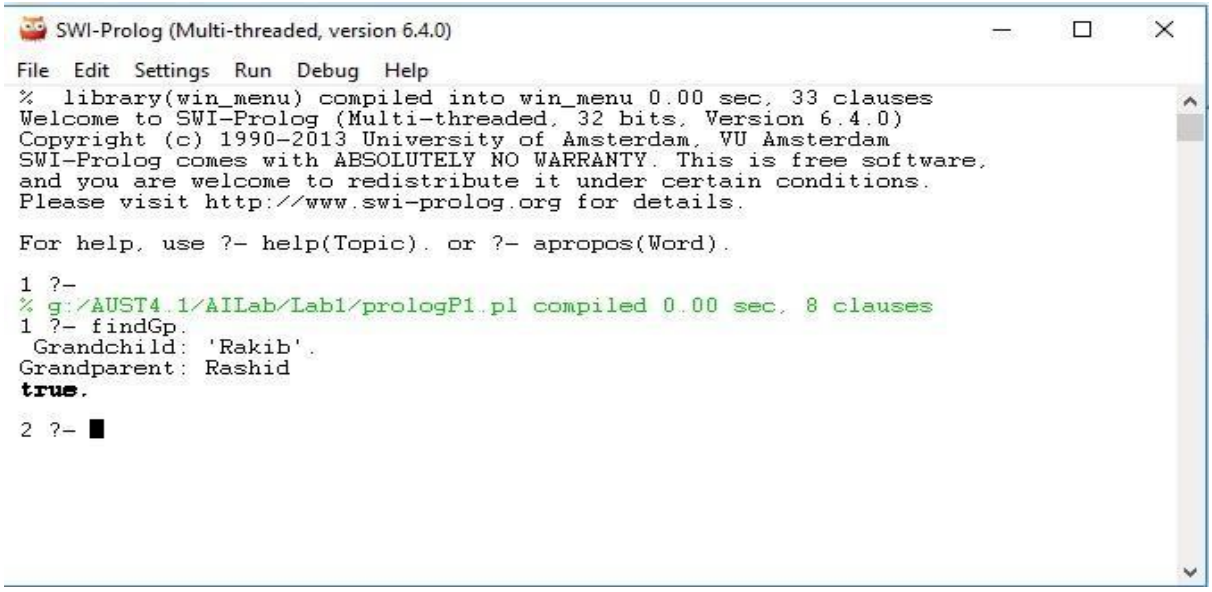
Python:

```
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
```



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.
```



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').

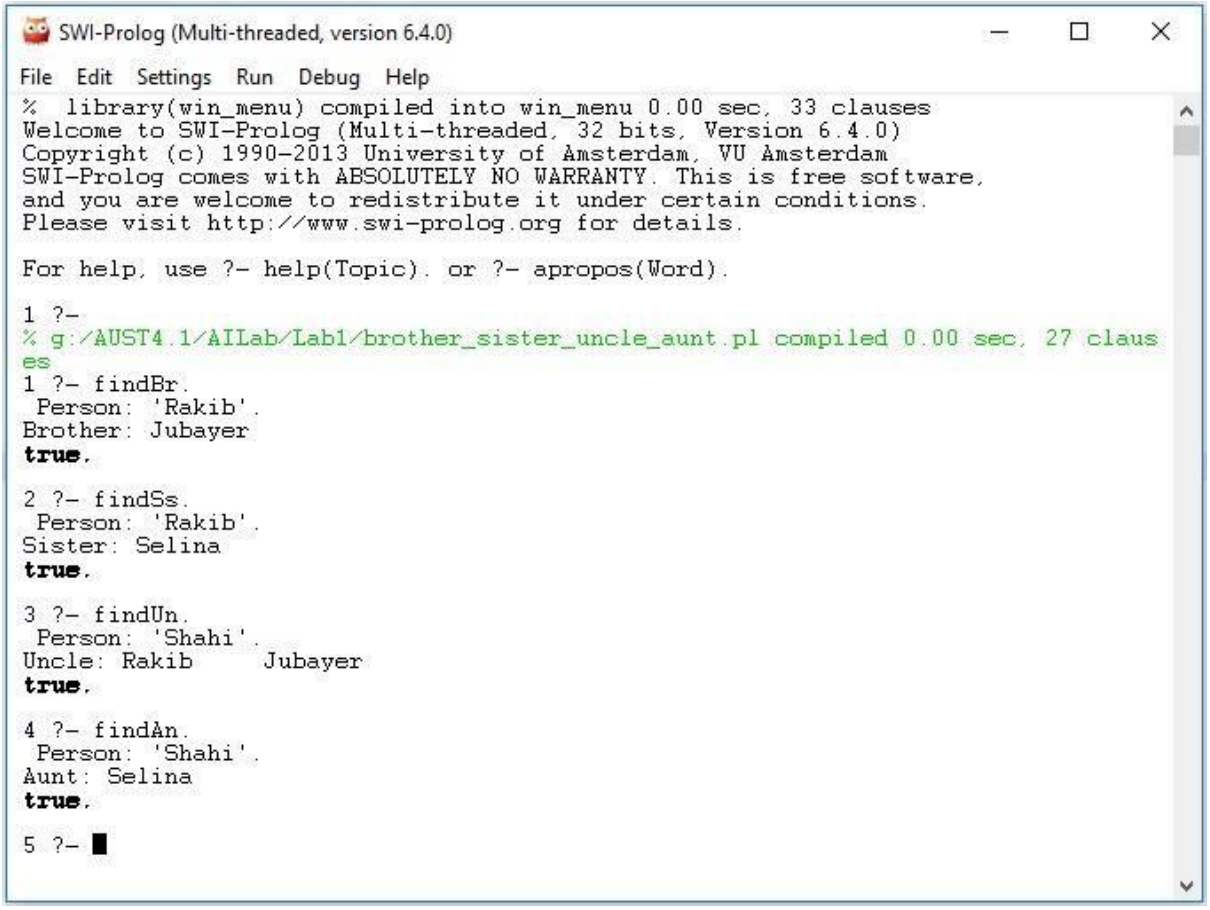
brother(X,Y):-parent(Z,X), parent(Z,Y), male(Y), not(X=Y). sister(X,Y):-parent(Z,X), parent(Z,Y), not(male(Y)), not(X=Y).

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.
```



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')
```