Ahsanullah University of Science & Technology

Department of Computer Science & Engineering

Course No. : CSE 4108

Course Name : Artificial Intelligence Lab

Assignment No. : 01

Submitted By:

Name : Umme Habiba
ID No. : 170104004
Session : Spring - 2020
Section : A1

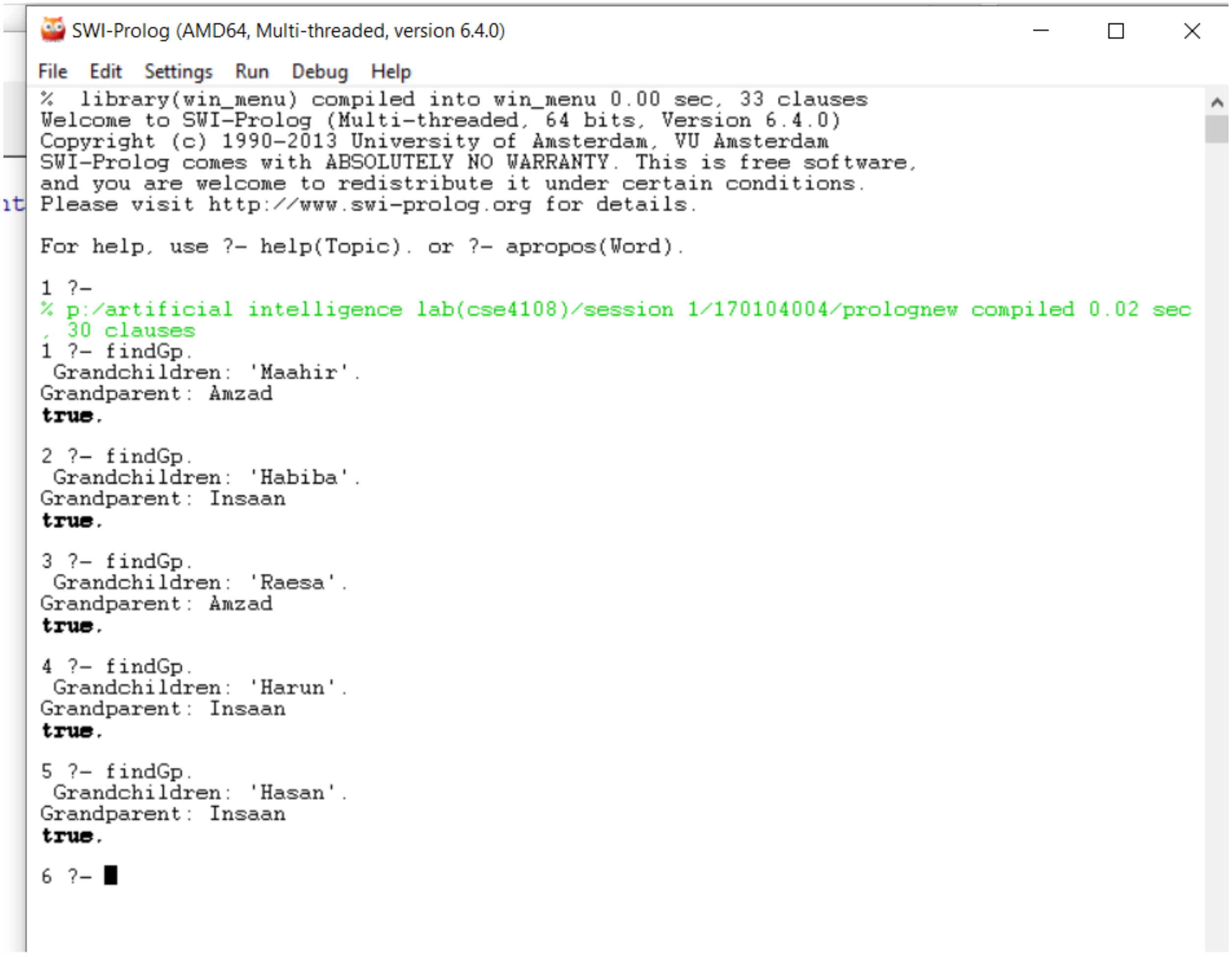
Question-03: Modify the Python and Prolog codes demonstrated above to find the grandparents of somebody.

Answer: For Prolog

Code:

```
parent('Insaan' , 'Amzad').
parent('Amzad' , 'Harun').
parent('Amzad' , 'Hasan').
parent('Amzad' , 'Habiba').
parent('Harun','Maahir').
parent('Harun','Raesa').

m('Insaan').
m('Amzad').
m('Harun').
m('Hasan').
m('Maahir').
f('Habiba').
f('Raesa').

grandparent(X, Z):- parent(X, Y), parent(Y, Z).
sibling(X,Z):- parent(Y,X), parent(Y,Z),not(X=Z).

brother(X,Z):-sibling(X,Z),m(X).
sister(X,Z):- sibling(X,Z),f(X).

uncle(X,Z):- brother(X,Y),parent(Y,Z).

aunt(X,Z):- sister(X,Y),parent(Y,Z).

findGp :- write(' Grandchildren: '), read(X), write('Grandparent: '),
            grandparent(Gp, X), write(Gp), tab(5),fail.

findGp.
```

Output:


SWI-Prolog (AMD64, Multi-threaded, version 6.4.0)
File Edit Settings Run Debug Help

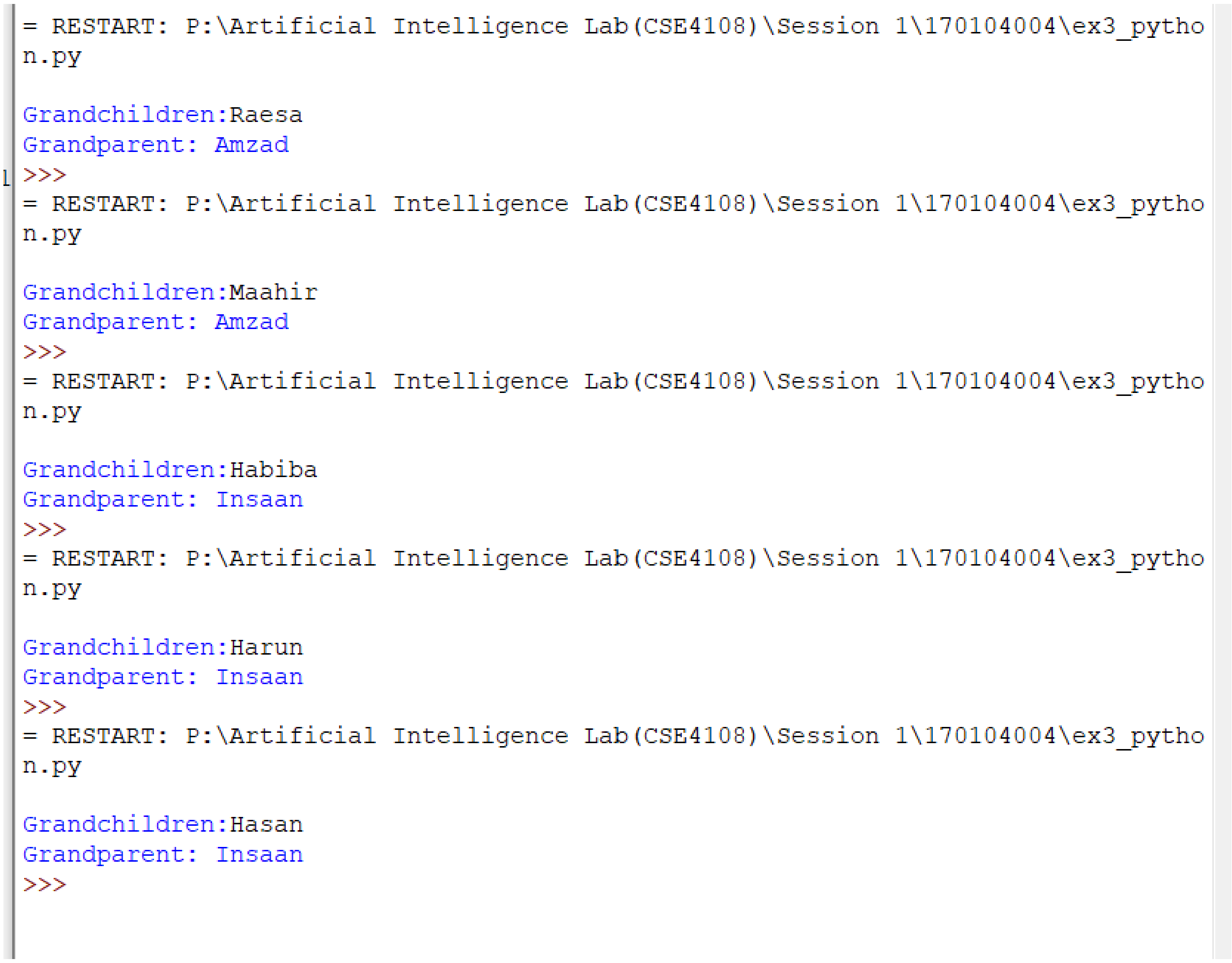
```
%  library(win_menu) compiled into win_menu 0.00 sec, 33 clauses
Welcome to SWI-Prolog (Multi-threaded, 64 bits, Version 6.4.0)
Copyright (c) 1990-2013 University of Amsterdam, VU Amsterdam
SWI-Prolog comes with ABSOLUTELY NO WARRANTY. This is free software,
and you are welcome to redistribute it under certain conditions.
Please visit http://www.swi-prolog.org for details.

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

1 ?-
% p:/artificial intelligence lab(cse4108)/session 1/170104004/prolognew compiled 0.02 sec
, 30 clauses
1 ?- findGp.
 Grandchildren: 'Maahir'.
Grandparent: Amzad
true.

2 ?- findGp.
 Grandchildren: 'Habiba'.
Grandparent: Insaan
true.

3 ?- findGp.
 Grandchildren: 'Raesa'.
Grandparent: Amzad
true.

4 ?- findGp.
 Grandchildren: 'Harun'.
Grandparent: Insaan
true.

5 ?- findGp.
 Grandchildren: 'Hasan'.
Grandparent: Insaan
true.

6 ?- █
```

Answer: For Python
Code:

```
tupleList1=[('parent', 'Insaan', 'Amzad'),('parent', 'Amzad', 'Harun')
        ,('parent', 'Amzad', 'Hasan'),('parent', 'Amzad', 'Habiba'),('parent', 'Harun',
'Raesa'),('parent', 'Harun', 'Maahir')]

X=str(input("\nGrandchildren:"))
print('Grandparent:', end=' ')
i=0
```

```
while(i<=5):
    if ((tupleList1[i][0] == 'parent')&( tupleList1[i][2] == X )):
        for j in range(6):
            if ((tupleList1[j][0] == 'parent') & ( tupleList1[i][1] == tupleList1[j][2])):

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

    i=i+1
```

Output:



```
= RESTART: P:\Artificial Intelligence Lab(CSE4108)\Session 1\170104004\ex3_pytho
n.py

Grandchildren:Raesa
Grandparent: Amzad
>>>
= RESTART: P:\Artificial Intelligence Lab(CSE4108)\Session 1\170104004\ex3_pytho
n.py

Grandchildren:Maahir
Grandparent: Amzad
>>>
= RESTART: P:\Artificial Intelligence Lab(CSE4108)\Session 1\170104004\ex3_pytho
n.py

Grandchildren:Habiba
Grandparent: Insaan
>>>
= RESTART: P:\Artificial Intelligence Lab(CSE4108)\Session 1\170104004\ex3_pytho
n.py

Grandchildren:Harun
Grandparent: Insaan
>>>
= RESTART: P:\Artificial Intelligence Lab(CSE4108)\Session 1\170104004\ex3_pytho
n.py

Grandchildren:Hasan
Grandparent: Insaan
>>>
```

Question-04: Enrich the KB demonstrated above with ‘brother’, ‘sister’, ‘uncle’ and ‘aunt’ rules in Python and Prolog.

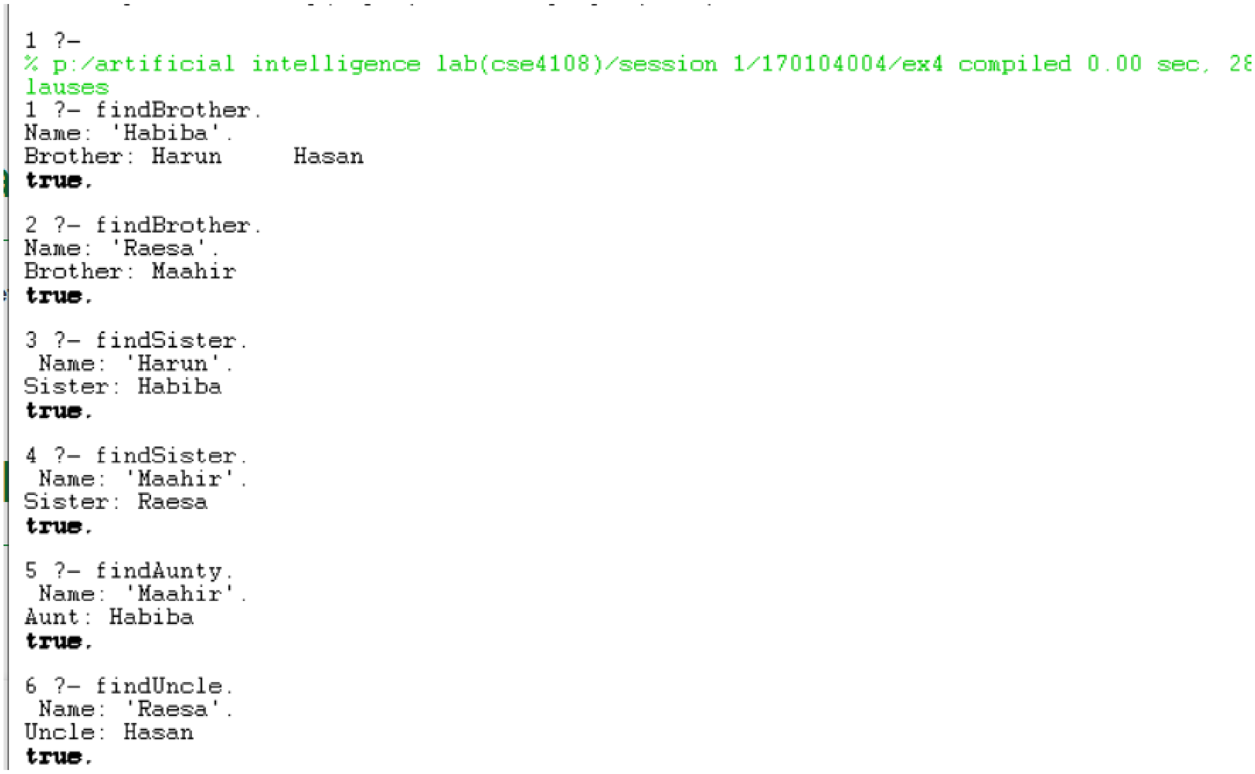
Answer: For Prolog

Code:

```
parent('Insaan' , 'Amzad').
parent('Amzad' , 'Harun').
parent('Amzad' , 'Hasan').
parent('Amzad' , 'Habiba').
parent('Harun','Maahir').
parent('Harun','Raesa').

m('Insaan').
m('Amzad').
m('Harun').
m('Hasan').
m('Maahir').
f('Habiba').
f('Raesa').

grandparent(X, Z):- parent(X, Y), parent(Y, Z).
sibling(X,Z):- parent(Y,X), parent(Y,Z),not(X=Z).

brother(X,Z):-sibling(X,Z),m(X).
sister(X,Z):- sibling(X,Z),f(X).

uncle(X,Z):- brother(X,Y),parent(Y,Z).

aunt(X,Z):- sister(X,Y),parent(Y,Z).

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


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


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

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

Output:



```
1 ?-
% p:/artificial intelligence lab(cse4108)/session 1/170104004/ex4 compiled 0.00 sec, 28
lauses
1 ?- findBrother.
Name: 'Habiba'.
Brother: Harun     Hasan
true.

2 ?- findBrother.
Name: 'Raesa'.
Brother: Maahir
true.

3 ?- findSister.
 Name: 'Harun'.
Sister: Habiba
true.

4 ?- findSister.
 Name: 'Maahir'.
Sister: Raesa
true.

5 ?- findAunty.
 Name: 'Maahir'.
Aunt: Habiba
true.

6 ?- findUncle.
 Name: 'Raesa'.
Uncle: Hasan
true.
```

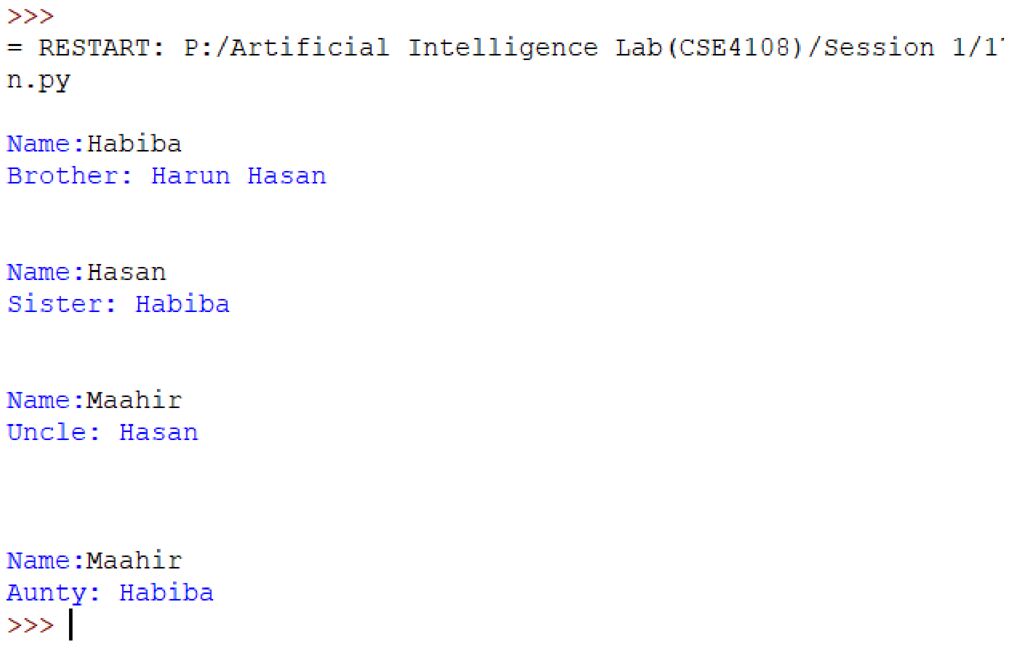
Answer: For Python

Code:

```
tupleList1=[('parent', 'Insaan', 'Amzad'),
           ('parent', 'Amzad', 'Harun'),
           ('parent', 'Amzad', 'Hasan'),
           ('parent', 'Amzad', 'Habiba'),
           ('parent', 'Harun', 'Maahir')]

X=str(input("\nName:"))
print('Brother:', end=' ')
i=0
while(i<=4):
          if (  (tupleList1[i][0] == 'parent')  & (tupleList1[i][2] == X) & (tupleList1[i][1] ==
'Amzad')):
                    for j in range(5):
```

```
            if( (tupleList1[j][0] == 'parent') &  (tupleList1[j][1] == 'Amzad') &
(tupleList1[j][2] != X) ):
    if(tupleList1[j][2]== 'Raesa' ):
              continue

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

    i=i+1

X=str(input("\n\n\nName:"))
print('Sister:', end=' ')
i=0

while(i<=4):
    if (  (tupleList1[i][0] == 'parent')  & (tupleList1[i][2] == X) & (tupleList1[i][1] ==
'Amzad')):
        for j in range(5):
            if( (tupleList1[j][0] == 'parent') &  (tupleList1[j][1] == 'Amzad') &
(tupleList1[j][2] != X) ):

                if((tupleList1[j][2]== 'Harun') or (tupleList1[j][2]== 'Hasan' )):
                    continue

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

    i=i+1

X=str(input("\n\n\nName:"))
print('Uncle:', end=' ')
i=0
while(i<=4):
    if (  (tupleList1[i][0] == 'parent')  & (tupleList1[i][2] == X) ):
        for j in range(5):
            if( (tupleList1[j][0] == 'parent') &  (tupleList1[j][1] == 'Amzad')  &
(tupleList1[j][2] != X) &  (tupleList1[i][1] == 'Harun')):
                if(tupleList1[j][2]== 'Habiba') or (tupleList1[j][2] == 'Harun' ):
                    continue

                print(tupleList1[j][2], end=' ')
    i=i+1

print()

X=str(input("\n\n\nName:"))
print('Aunty:', end=' ')
i=0
while(i<=4):
    if (  (tupleList1[i][0] == 'parent')  & (tupleList1[i][2] == X) ):
        for j in range(5):
            if( (tupleList1[j][0] == 'parent') &  (tupleList1[j][1] == 'Amzad')  &
(tupleList1[j][2] != X) &  (tupleList1[i][1] == 'Harun')):

                if((tupleList1[j][2]== 'Hasan') or (tupleList1[j][2]== 'Harun' )):
                    continue

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

    i=i+1
```

Output:



```
>>>
= RESTART: P:/Artificial Intelligence Lab(CSE4108)/Session 1/1
n.py

Name:Habiba
Brother: Harun Hasan


Name:Hasan
Sister: Habiba


Name:Maahir
Uncle: Hasan



Name:Maahir
Aunty: Habiba
>>> 
```