Ahsanullah University of Science and Technology (AUST)
Department of Computer Science and Engineering

Assignment 1

Course No.: CSE4108

Course Title: Artificial Intelligence Lab

Date of Submission-16.05.2023

Submitted To- Dr. S.M.A. Al-Mamun & Mr. Raihan Tanvir.

Submitted By-

MD Shihabul Islam Shovo
190204075
B
Year- 4th
Semester-1st
Department-CSE

1) **Question: 1.** Modify the Python and Prolog codes demonstrated above to find the grandparents of somebody.
2) **Question: 1.** Enrich the KB demonstrated above with ‘brother’, ‘sister’, ‘uncle’ and ‘aunt’ rules in Python and Prolog.

Answer:

Python Code:

```
#Assignment - 01
tupleList1=[('parent', 'john', 'mary'),('parent', 'john', 'peter'), ('parent', 'sue', 'mary'),
            ('parent', 'sue', 'peter'), ('parent', 'mary', 'tom'), 'parent', 'peter', 'lisa'),
           ('parent', 'peter', 'bob'), ('parent', 'john', 'tyler'), ('parent', 'sue', 'tyler')]
genderList = [('male','john'), ('male','peter'), ('male','tom'), ('male','bob'),
        ('male','tyler'), 'female','sue'), 'female','mary'), ('female','lisa'),]

ch = int(input("Enter your choice to find relation of \n1.Brother\n2.Sister\n3.Uncle\n4.Aunt\nChoice: "))
i,l = 0,0
if ch == 1 or ch == 2:
   X = str(input("Enter the name to find the siblings: "))
   if ch == 1:
      print("Brother: ", end=' ')
   else:
      print("Sister: ", end=' ')
   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[j][2] != X)):
               for k in range(10):
                  if ch == 1:
                     if ((genderList[k][0] == 'male') & (genderList[k][1] == tupleList1[j][2])):
```

```
                print(tupleList1[j][2], end=' ')
            else:
                if ((genderList[k][0] == 'female') & (genderList[k][1] == tupleList1[j][2])):
                    print(tupleList1[j][2], end=' ')
    i = i + 1
elif ch == 3 or ch == 4:
    X = str(input("Enter the name to find someone's uncle/aunt: "))
    if ch == 3:
        print("Uncle: ", end=' ')
    else:
        print("Aunt: ", end=' ')
    while(l<=8):
        if ((tupleList1[l][0] == 'parent') & (tupleList1[l][2] == X)) :
            while(i<=8):
                if ((tupleList1[i][0] == 'parent') & (tupleList1[i][2] == tupleList1[l][1])) :
                    for j in range(9):
                        if ((tupleList1[j][0] == 'parent') & (tupleList1[i][1] == tupleList1[j][1]) & (tupleList1[j][2] != tupleList1[l][1])):
                            for k in range(8):
                                if ch == 3:
                                    if ((genderList[k][0] == 'male') & (genderList[k][1] == tupleList1[j][2])):
                                        print(tupleList1[j][2], end=' ')
                                else:
                                    if ((genderList[k][0] == 'female') & (genderList[k][1] == tupleList1[j][2])):
                                        print(tupleList1[j][2], end=' ')
                i = i + 1
        l = l+1
elif ch == 5:
    # Procedure to find the grandparent of someone
```

```
X=str(input("Enter the name to find someone's grandparent:"))
print('Grandparent:', end=' ')
i=0
while(i<=9):
        if ((tupleList1[i][0] == 'parent')&( tupleList1[i][2] == X)):
                for j in range(9):
                if ((tupleList1[j][0] == 'parent') & ( tupleList1[i][1] == tupleList1[j][2])):
                        print(tupleList1[j][1], end=' ')
        i=i+1
```

Prolog Code:

```
% Facts about family relationships

parent('john', 'mary').
parent('john', 'peter').
parent('sue', 'mary').
parent('sue', 'peter').
parent('mary', 'tom').
parent('peter', 'lisa').
parent('peter', 'bob').
parent('john', 'tyler').
parent('sue', 'tyler').

male('john').
male('peter').
male('tom').
```

```
male('bob').
male('tyler').

female('sue').
female('mary').
female('lisa').

% Rules to define family relationships
sibling(X, Y) :- parent(Z, X), parent(Z, Y), X \= Y.
brother(X, Y) :- sibling(X, Y),male(X).
sister(X, Y) :- sibling(X, Y), female(X).
uncle(X, Y) :- parent(Z, Y), brother(X, Z).
aunt(X, Y) :- parent(Z, Y), sister(X, Z).
grandparent(Z, X) :- parent(Y, X), parent(Z, Y).

findGp :- write('Person name to find their grandparents: '),
        read(X),
                grandparent(Gc, X),
                  format('~w is a grndparent of ~w. ~n',[Gc, X]),
                    fail.
```