
# Ahsanullah University of Science and Technology

*Department of Computer Science & Engineering*

| | |
|---|---|
| Course No. | CSE 4108 |
| Course Name | Artificial Intelligence Lab |
| Assignment No. | 01 |

## Submitted To:

| | |
|---|---|
| Md. Siam Ansary | Tonmoy Hossain |
| Department of CSE, AUST | Department of CSE, AUST |

## Submitted By:

| | |
|---|---|
| **Name** | Easinur Rashid Prantick |
| **ID No.** | 17.02.04.030 |
| **Session** | Fall - 2020 |
| **Section** | A (A2) |

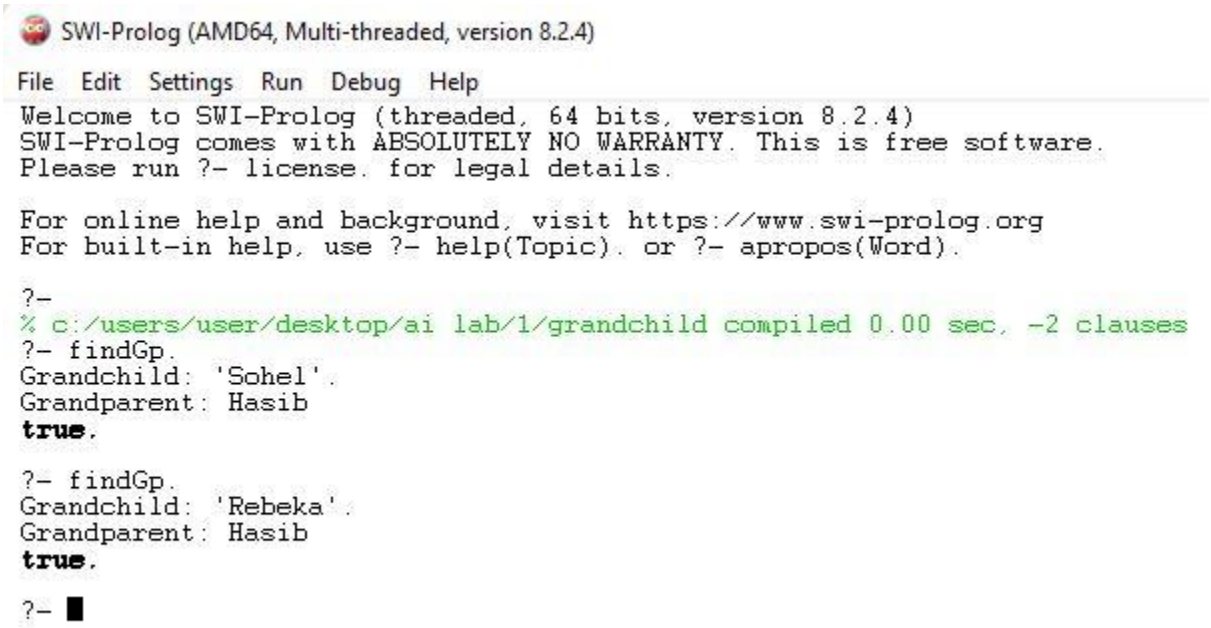
# Solution: Session 1

**Q03: Modify the Python and Prolog codes demonstrated above to find the grandparents of somebody.**

**Code in Prolog:**

```
parent('Hasib','Rakib').
parent('Rakib','Sohel').
parent('Rakib','Rebeka').
parent('Rashid','Hasib').

grandchild(Z, X):-
    parent(Y, X), parent(Z, Y).


findGp:-
    write('Grandchild: '), read(Gp), write('Grandparent: '),
    grandchild(X, Gp), write(X), tab(5), fail.
findGp.
```

**Input and Output:**




**Code in Python:**

```
tuplelist1 =  [
   ('parent', 'Hasib','Rakib'),
   ('parent', 'Rakib','Sohel'),
   ('parent', 'Rakib','Rebeka'),
   ('parent', 'Rashid','Hasib')
]
```

```
X = str(input("Grandchild : "))
print("Grandchildren : ", end=' ')
i,j = 0,0
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
```

**Output:**

```
Grandchild : Rebeka
Grandchildren :  Hasib
Process finished with exit code 0
```

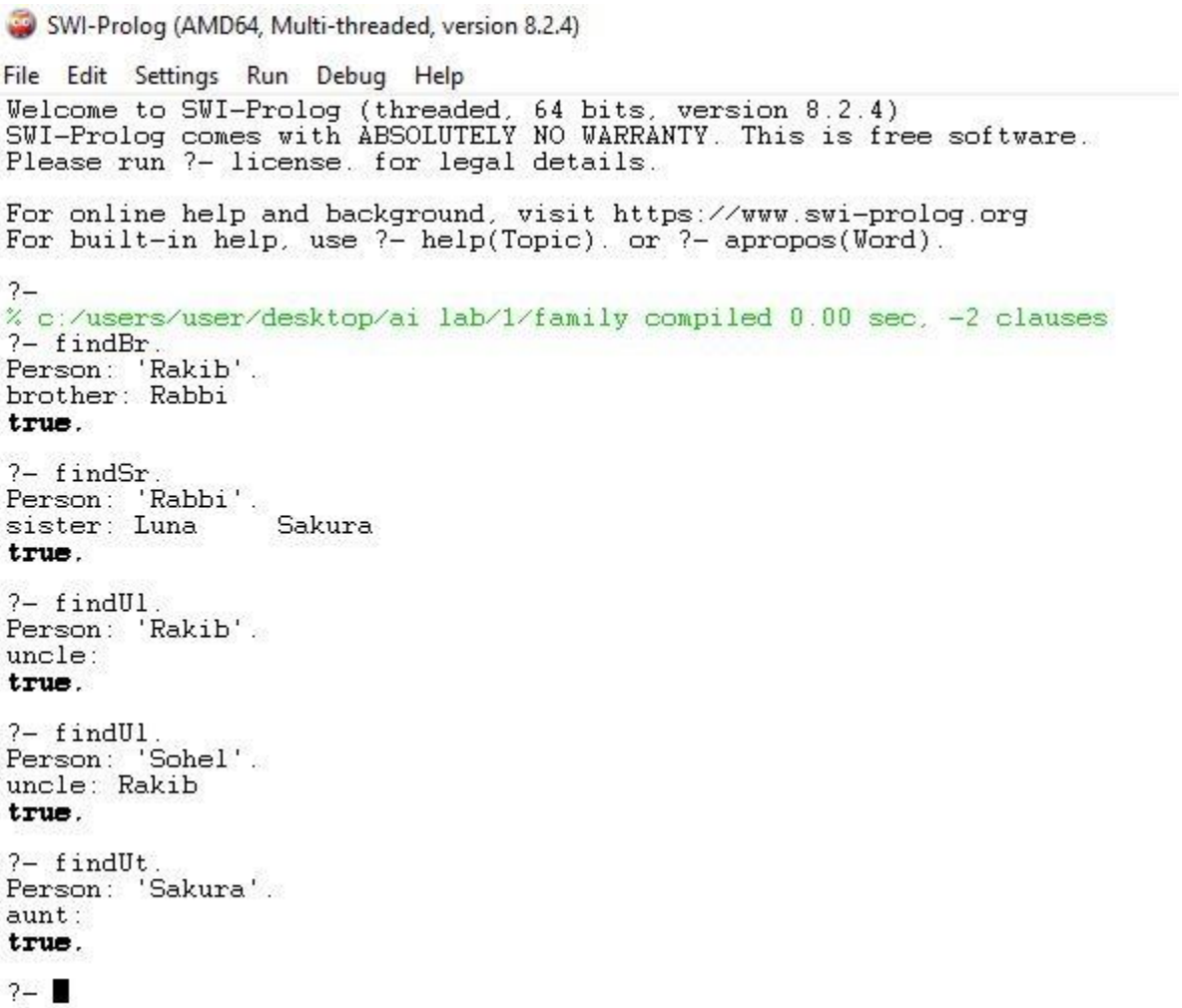
**Q 04: Enrich the KB demonstrated above with 'brother', 'sister', 'uncle' and 'aunt' rules in Python and Prolog.**

**Code in Prolog:**

```
parent('Hasib','Rakib').
parent('Hasib','Rabbi').
parent('Hasib','Luna').
parent('Hasib','Sakura').
parent('Rabbi','Sohel').

male('Hasib').
male('Rakib').
male('Rabbi').
male('Sohel').

female('Luna').
female('Sakura').


brother(X,Z):-
    parent(Y,X),parent(Y,Z),male(X),dif(X,Z).
findBr :- write('Person: '), read(X), write('brother: '),
    brother(Br, X), write(Br), tab(5), fail.
findBr.

sister(X,Z):-
    parent(Y,X),parent(Y,Z),female(X),dif(X,Z).
findSr :- write('Person: '), read(X), write('sister: '),
    sister(Sr, X), write(Sr), tab(5), fail.
findSr.
```

```
uncle(X,Y):-parent(Z,Y), brother(X,Z).
findUl :- write('Person: '), read(X), write('uncle: '),
    uncle(Ul,X), write(Ul), tab(5), fail.
findUl.

aunt(X,Y):-parent(Z,Y), sister(X,Z).
findUt :- write('Person: '), read(X), write('aunt: '),
    uncle(Ut,X), write(Ut), tab(5), fail.
findUt.
```

**Input and Output:**




**Code in Python:**

```
parent = [('parent', 'Hasib', 'Rakib'), ('parent', 'Hasib', 'Rabbi'),
          ('parent', 'Hasib', 'Luna'), ('parent', 'Hasib', 'Sakura'),
          ('parent', 'Rabbi', 'Sohel')]
```

```
male = [('male', 'Hasib'), ('male', 'Rakib'),
        ('male', 'Rabbi'), ('male', 'Sohel')]

female = [('female', 'Luna'), ('female', 'Sakura')]

def findGp():
    X = str(input("Grandchildren:\n"))
    print('Grandparent:', end='\n')
    bl = True
    for i in range(len(parent)):
        if (parent[i][0] == 'parent') & (parent[i][2] == X):
            for j in range(len(parent)):
                if (parent[j][0] == 'parent') & (parent[i][1] == parent[j][2]):
                    print(parent[j][1])
                    bl = False
    if(bl):
        print('N/A')

def findGc():
    X = str(input("Grandparent:\n"))
    print('Grandchildren:', end='\n')
    bl = True
    for i in range(len(parent)):
        if (parent[i][0] == 'parent') & (parent[i][1] == X):
            for j in range(len(parent)):
                if (parent[j][0] == 'parent') & (parent[i][2] == parent[j][1]):
                    print(parent[j][2])
                    bl = False
    if(bl):
        print('N/A')

def findBr():
    X = str(input("Person:\n"))
    print('Brother:', end='\n')
    bl = True
    for i in range(len(parent)):
        if (parent[i][0] == 'parent') & (parent[i][2] == X):
            for j in range(len(parent)):
                if (parent[j][0] == 'parent') & (parent[j][1] == parent[i][1]) &
(parent[i][2]!=parent[j][2]):
                    for k in range(len(male)):
                        if(male[k][0]=='male') & (male[k][1]==parent[j][2]):
                            print(male[k][1])
                            bl = False
    if (bl):
        print('N\A')

def findBr2():
    X = str(input("Person:\n"))
    print('Brother:', end='\n')
    bl = True
    for i in range(len(parent)):
        if (parent[i][0] == 'parent') & (parent[i][2] == X):
            for j in range(len(parent)):
```

```
                if (parent[j][0] == 'parent') & (parent[j][1] == parent[i][1]) &
(parent[i][2]!=parent[j][2]):
                    for k in range(len(male)):
                        if(male[k][0]=='male') & (male[k][1]==parent[j][2]):
                            print(male[k][1])
                            bl = False
    if (bl):
        print('N\A')

def findSr():
    X = str(input("Person:\n"))
    print('Sister:', end='\n')
    bl = True
    for i in range(len(parent)):
        if (parent[i][0] == 'parent') & (parent[i][2] == X):
            for j in range(len(parent)):
                if (parent[j][0] == 'parent') & (parent[j][1] == parent[i][1]) &
(parent[i][2]!=parent[j][2]):
                    for k in range(len(female)):
                        if(female[k][0]=='female') & (female[k][1]==parent[j][2]):
                            print(female[k][1])
                            bl = False
    if(bl):
        print('N\A')

def findUl():
    X = str(input("Person:\n"))
    print('Uncle:', end='\n')
    bl = True
    for i in range(len(parent)):
        if(parent[i][0]=='parent') & (parent[i][2]==X):#parent[i][1] parent
            for j in range(len(parent)):
                if (parent[j][0] == 'parent') &
(parent[j][2]==parent[i][1]):#parent[j][1] grandparent
                    for k in range(len(parent)):
                        if (parent[j][0] == 'parent') & (parent[j][1] ==
parent[k][1]) & (parent[i][1]!=parent[k][2]): #parent[k][2] parent's sibling
                            for l in range(len(male)):
                                if(male[l][0]=='male') & (male[l][1]==parent[k][2]):
#male[l][1] uncle
                                    print(male[l][1])
                                    bl = False
    if(bl):
        print('N/A')

def findUt():
    X = str(input("Person:\n"))
    print('Aunt:', end='\n')
    bl = True
    for i in range(len(parent)):
        if(parent[i][0]=='parent') & (parent[i][2]==X):#parent[i][1] parent
            for j in range(len(parent)):
                if (parent[j][0] == 'parent') &
(parent[j][2]==parent[i][1]):#parent[j][1] grandparent
                    for k in range(len(parent)):
```

```
                        if (parent[j][0] == 'parent') & (parent[j][1] ==
parent[k][1]) & (parent[i][1]!=parent[k][2]): #parent[k][2] parent's sibling
                            for l in range(len(female)):
                                if(female[l][0]=='female') &
(female[l][1]==parent[k][2]): #female[l][1] aunt
                                    print(female[l][1])
                                    bl = False
    if(bl):
        print('N/A')

def Print():
    print('1: Grandparent, 2: Grandchildren, 3: Brother, 4: Sister, 5: Uncle, 6:
Aunt')
    x = input("Choose Option:\n")
    return x

x = Print()
while (1):
    if x == '1':
        findGp()
    elif x == '2':
        findGc()
    elif x == '3':
        findBr()
    elif x == '4':
        findSr()
    elif x == '5':
        findUl()
    elif x == '6':
        findUt()
    else:
      break
    x = Print()
```

**Output:**

```
3
Person:
Rakib
Brother:
Rabbi
1: Grandparent, 2: Grandchildren, 3: Brother, 4: Sister, 5: Uncle, 6: Aunt
Choose Option:
4
Person:
Rabbi
Sister:
Luna
Sakura
1: Grandparent, 2: Grandchildren, 3: Brother, 4: Sister, 5: Uncle, 6: Aunt
Choose Option:
5
Person:
Sohel
Uncle:
Rakib
1: Grandparent, 2: Grandchildren, 3: Brother, 4: Sister, 5: Uncle, 6: Aunt
Choose Option:
6
Person:
Sohel
Aunt:
Luna
Sakura
```