

**CS431: Programming Languages Lab (2020)**  
**Department of Computer Science and Engineering**  
**Indian Institute of Technology Guwahati**

**Assignment Set II**

- ❖ Assignments will be evaluated by the TAs.
- ❖ You should submit report (for answering non-code related questions, if any), complete source codes and executable files (whichever applicable).
- ❖ All codes must be properly documented and good code writing practice should be followed (carry marks).
- ❖ Copying is strictly prohibited. Any case of copying will automatically result in F grade for the whole course, irrespective of your performance in the other parts of the lab.
- ❖ Submission deadline: **October 31, 2020.**
- ❖ Marks distribution: 20, 30, 50.

**1. Finding Relationship and Gender: (20 Marks)**

Consider a database that describes PARENT relationships as well as GENDER relationships. In this database, you can have predicate like `parent(jatin,avantika)` that interprets as "Jatin is a parent of Avantika". The predicate `male(jatin)` interprets as "Jatin is a man". Similarly, the predicate `female(avantika)` interprets as "Avantika is a woman". An example database of facts is:

```
parent(jatin,avantika).  
parent(jolly,jatin).  
parent(jolly,kattappa).  
parent(manisha,avantika).  
parent(manisha,shivkami).  
parent(bahubali,shivkami).  
male(kattappa).  
male(jolly).  
female(shivkami).  
female(avantika).  
male(bahubali).
```

Observe that few things are not specified in the database (for instance, `male(jatin)`).

**Question1: UNCLE (10 Marks)**

Your task is to write a Prolog program to determine UNCLE relationship using the above two types of facts only. You should not consider uncles "by marriage", i.e., for A to be B's uncle the two must have blood relationship. Your program should be able to generate the following outputs.

```
?- uncle(katappa,avantika).
```

Yes

```
?- uncle(avantika,manisha).
```

No

```
?- uncle(katappa,A).
```

A = avantika ;

No

```
?- uncle(jatin, avantika).
```

No

```
?- uncle(A, B).
```

A = katappa

B = avantika ;

No

### Question2: HALF SISTER (10 Marks)

In a similar way, write a Prolog program for HALFSISTER<sup>1</sup> relationship. Your program should be able to generate the following outputs.

```
?- halvesister(avantika, shivkami).
```

Yes

```
?- halvesister(A, shivkami).
```

A = avantika;

No

```
?- halvesister(A, B).
```

A = avantika

B = shivkami;

A = shivkami;

B = avantika;

No

## 2. Bus Travel Planner

(30 Marks)

Consider the Bus Service in any locality (you are free to choose any place). Maintain a knowledge base of buses along with their routes hop by hop.

The Bus details are stored in the following format:

*Bus (Number, Origin, Destination Place, Departure Time, Arrival Time, Distance, Cost)*

<sup>1</sup>Half Sister means a sister with whom one has only one parent in common

where,

Number→ Bus number

Origin→ Starting stoppage of the hop

Destination→ Ending stoppage of the hop

Departure time→ Starting time at the Origin

Arrival time→ Time of reaching the Destination

(Store the time in 24 hour format e.g. 7AM is stored as 7, 4PM is stored as 16, 1.30PM is stored as 13.5)

Distance→ Distance between Origin and Destination in kilometers

Cost→ Bus Fare between Origin and Destination in Rs. (The overall fare is the sum of fares between each hop)

Write a PROLOG program which would provide the optimized route between two places based on:

a) Distance

b) Time

c) Cost

[The program must return 3 results based on the above mentioned factors viz. a), b) and c)]

### **Sample Bus details:**

Bus(123,Amingaon,Jalukbari,14.5,15,10,10).

Bus(756,Panbazar, Chandmari, 16,16.5,7,8).

### **Sample Query and Result**

?-Route(Amingaon,Paltanbazar).

Optimum Distance:

Amingaon,123->Jalukbari,123->Maligaon,123->Paltanbazar,153

Distance=19.5 ,Time=2 ,Cost=20

Optimum Time:

Amingaon,123->Jalukbari,123->Lokhra,327->Panbazar,823

Distance=25 ,Time= 1, Cost=30

Optimum Cost:

Amingaon,123->Jalukbari,123->Maligaon,123->Paltanbazar,153

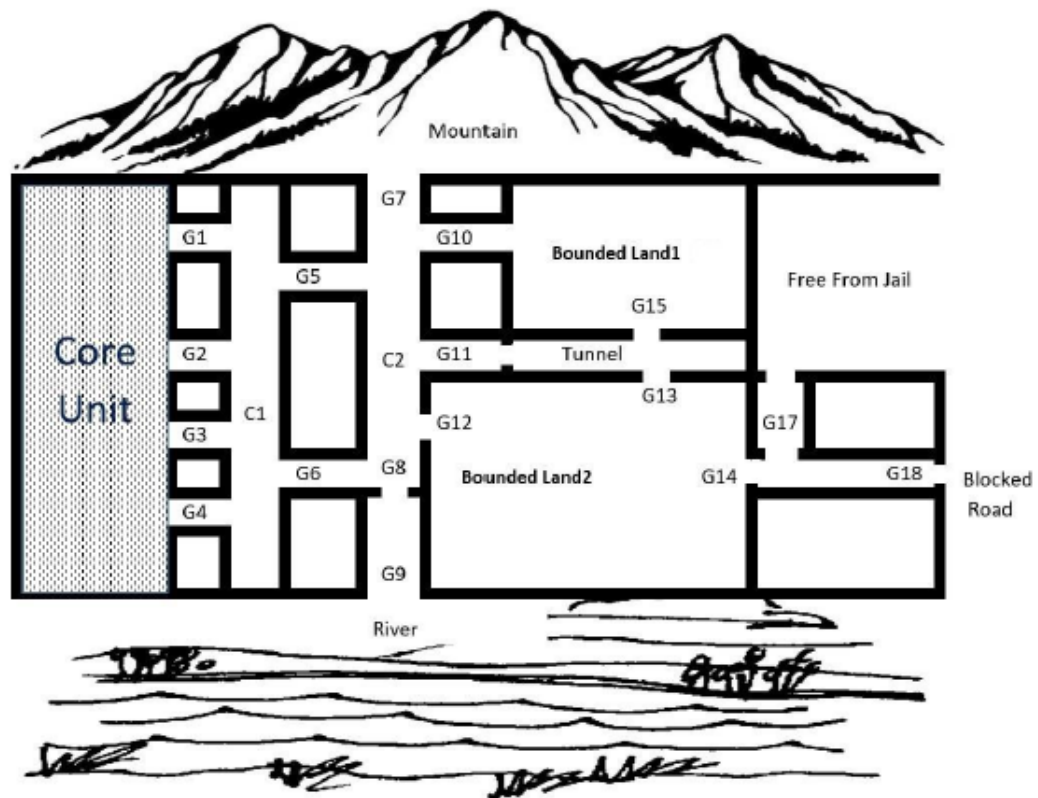
Distance=19.5 ,Time=2 ,Cost=2

**(50 Marks)**

On the New Year celebration, all the security personnel were busy in other tasks. So, in the meantime the prisoners made plans to escape from the jail.

There are two bounded land outwards of gate G10 and G12 and one tunnel which is connected to G11, G13 and G15. The tunnel is directly connected to the core unit of the jail. There is a blocked road ahead of G18. G17 is a gate which is open towards outside.

The prisoner may return to his previous position if required.



G1→G5 = 4ft, G2→G5=6ft, G3→G5=8ft, G4→G5=9ft, G1→G6=10ft, G2→G6=9ft, G3→G6=3ft, G4→G6=5ft, G5→G7=3ft, G5→G10=4ft, G5→G11=6ft, G5→G12=7ft, G5→G6=7ft, G5→G8=9ft, G6→G8=2ft, G6→G12=3ft, G6→G11=5ft, G6→G10=9ft, G6→G7=10ft, G7→G10=2ft, G7→G11=5ft,

G7→G12=7ft, G7→G8=10ft, G8→G9=3ft, G8→G12=3ft, G8→G11=4ft, G8→G10=8ft,  
G10→G15=5ft, G10→G11=2ft, G10→G12=5ft, G11→G15=4ft, G11→G13=5ft, G11→G12=4ft,  
G12→G13=7ft, G12→G14=8ft, G15→G13=3ft, G13→G14=4ft, G14→G17=5ft, G14→G18=4ft,  
G17→G18=8ft

When the distance is given between a pair of gates, it means the prisoner can move between the pair of gates in any order. For example, distance between G1→G5=4ft means the prisoner can move from G1 to G5 and vice-versa.

For the above scenario:

A) Write a PROLOG program to find all possible paths for a prisoner to escape from the jail.

B) Find the optimal path for the prisoner to escape. Optimal path is a path in which the prison has to cover minimum distance to escape from the jail.

C) Given a path, your program should be able to find if it is valid or invalid in terms of escaping the jail.

Sample Input/Output:

Input:

Optimal(X)

Output:

G3→G6→G12→G14→G17

Input:

Valid ([G1, G6, G8, G9, G8, G7, G10, G15, G13, G14, G18, G17]) [The argument is a list]

Output: True