

Commands should be run from inside the folder 170101008\_annanya provided.

---

## 1) Finding Relationship and Gender

To load the prolog program run:

```
$ swipl -s first.pl
```

### Uncle Relationship

To run queries regarding uncle relationship use:

```
?- uncle(X, Y).
```

Uncle Query examples:

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

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

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

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

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

**Note:** The spelling of "kattappa" provided in the database is different from what I saw in the question queries. I haven considered the spelling "kattappa" which is given in the database.

### Half-Sister Relationship

To run queries regarding Half-Sister relationship use:

```
?- halvesister(X, Y).
```

Half-Sister Query examples:

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

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

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

---

## 2) Bus Travel Planner

To load the prolog program run:

```
$ swipl -s second.pl
```

To find optimal routes between two places run:

```
?- route(From, To).
```

This query will yield three shortest paths according to **time taken, cost and distance**. While calculating the time of travel I have taken the **overnight time** also into account: that is, if the departure time of a bus from a location is less than my arrival time of reaching that location than I consider that the bus travel happens the next day and the whole time spent in between is considered in time taken. This is the reason why you might see time taken to be more than 24 hours for certain paths. I have printed the bus number also with the path. It represents the bus that you took to reach that location. It is true for every point except the starting point for whom I have printed the bus number corresponding to the bus that left the starting point. Locations that can be use to find routes are ['Amingaon', 'Jalukbari', 'Paltanbazaar', 'Fancybazaar', 'Maligaon', 'Lokhra', 'Panbazaar', 'Chandmari']. You can look into the code for the possible bus routes. I have provided some sample queries:

Example Queries are:

```
?- route('Amingaon', 'Chandmari').  
?- route('Amingaon', 'Panbazaar').
```

---

### 3) Prisoner Escape Problem

To load the prolog program run:.

```
$ swipl -s third.pl
```

#### All Paths:

```
?- findPaths.
```

This query prints the total number of valid paths. To see each path use seePath. which prints each path stepwise after you press semicolon(;):

```
?- seePath.
```

#### Optimal Path:

To find and print the optimal Path use

```
?- optimal(X).
```

#### Validity:

To check if a path is valid. Here **X** is a list of nodes which symbolise a path

```
?- valid(X).
```

example Query:

```
?- valid(['G1', 'G6', 'G8', 'G9', 'G8', 'G7', 'G10', 'G15', 'G13', 'G14', 'G18', 'G17']).
```

**Note:** The nodes here are represented by strings as we can't use capital letters to represent them.