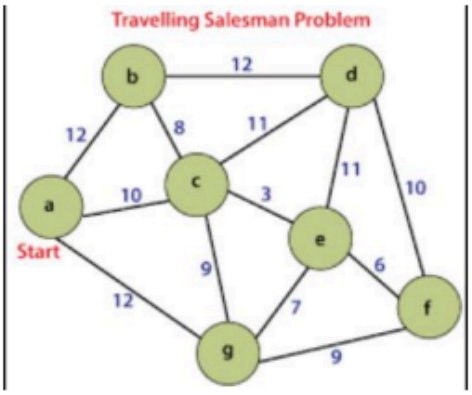
Lab Assignment 3

Garvit Shah
U21CS089

1. Implement Traveling Salesman problem in prolog.

The travelling salesman problem is a graph computational problem where the salesman needs to visit all cities (represented using nodes in a graph) in a list just once and the distances (represented using edges in the graph) between all these cities are known. The solution that is needed to be found for this problem is the shortest possible route in which the salesman visits all the cities and returns to the origin city.




```
edge(a,b,12).
edge(a,c,10).
edge(a,g,12).
edge(b,c,8).
edge(c,g,9).
edge(b,d,12).
edge(c,d,11).
edge(c,e,3).
edge(g,e,7).
edge(e,d,11).
edge(g,f,9).
edge(e,f,6).
edge(d,f,10).
```

```
edge(b,a,12).
edge(c,a,10).
edge(g,a,12).
edge(c,b,8).
edge(g,c,9).
edge(d,b,12).
```

```
edge(d,c,11).
edge(e,c,3).
edge(e,g,7).
edge(d,e,11).
edge(f,g,9).
edge(f,e,6).
edge(f,d,10).
len([], 0).
len([H|T], N):- len(T, X), N is X+1 .

best_path(Visited, Total):- path(a, a, Visited, Total).

path(Start, Fin, Visited, Total) :- path(Start, Fin, [Start],
Visited, 0, Total).

path(Start, Fin, CurrentLoc, Visited, Costn, Total) :-
    edge(Start, StopLoc, Distance), NewCostn is Costn + Distance,
\+ member(StopLoc, CurrentLoc),
    path(StopLoc, Fin, [StopLoc|CurrentLoc], Visited, NewCostn,
Total).

path(Start, Fin, CurrentLoc, Visited, Costn, Total) :-
    edge(Start, Fin, Distance), reverse([Fin|CurrentLoc],
Visited), len(Visited, Q),
    (Q\=8 -> Total is 100000; Total is Costn + Distance).

shortest_path(Path):-setof(Cost-Path, best_path(Path,Cost),
Holder),pick(Holder,Path).

best(Cost-Holder,Bcost-_,Cost-Holder):- Cost<Bcost,!.
best(_,X,X).

pick([Cost-Holder|R],X):- pick(R,Bcost-Bholder),best(Cost-
Holder,Bcost-Bholder,X),!.
pick([X],X).
```

```
~/D/C/V/A/L/Assign3 swipl
Welcome to SWI-Prolog (threaded, 64 bits, version 9.0.4)
SWI-Prolog comes with ABSOLUTELY NO WARRANTY. This is free software.
Please run ?- license. for legal details.

For online help and background, visit https://www.swi-prolog.org
For built-in help, use ?- help(Topic). or ?- apropos(Word).

?- consult('q1.pl').
Warning: /Users/garvitshah/Desktop/College/VI/AI/Lab/Assign3/q1.pl:29:
Warning:    Singleton variables: [H]
true.

?- shortest_path(Path).
Path = 63-[a, c, e, g, f, d, b, a].
```