AI(2180703)

Tutorial-8

Name : Yogesh Bavishi

Enrollment No.: 170200107003

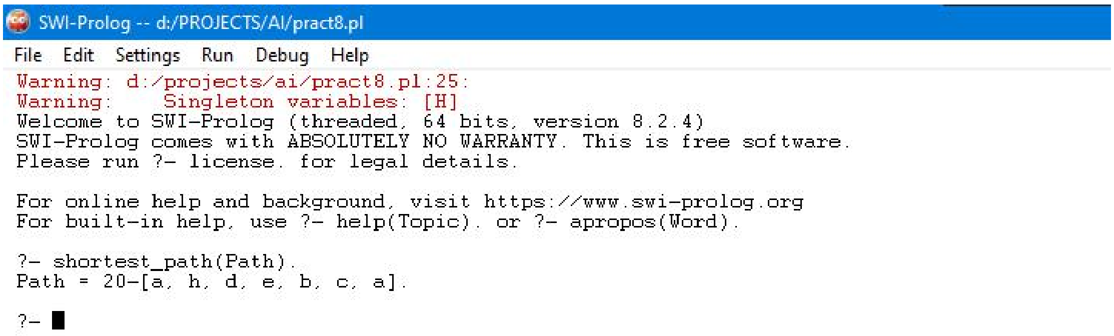
Division/Batch: E/E1

Q: Write a program to solve travelling salesman problem using Prolog.

Code(pract8.pl):

```
edge(a, b, 3).
edge(a, c, 4).
edge(a, d, 2).
edge(a, e, 7).
edge(b, c, 4).
edge(b, d, 6).
edge(b, e, 3).
edge(c, d, 5).
edge(c, e, 8).
edge(d, e, 6).
edge(b, a, 3).
edge(c, a, 4).
edge(d, a, 2).
edge(e, a, 7).
edge(c, b, 4).
edge(d, b, 6).
edge(e, b, 3).
edge(d, c, 5).
edge(e, c, 8).
edge(e, d, 6).
edge(a, h, 2).
edge(h, d, 1).

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, \+ me
mber(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\=7 -> Total is 100000; Total is Costn + Distance).

shortest_path(Path):-setof(Cost-Path, best_path(Path,Cost), Holder),pic
k(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).
```

Output :


SWI-Prolog -- d:/PROJECTS/AI/pract8.pl
File Edit Settings Run Debug Help

```
Warning: d:/projects/ai/pract8.pl:25:
Warning:    Singleton variables: [H]
Welcome to SWI-Prolog (threaded, 64 bits, version 8.2.4)
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).

?- shortest_path(Path).
Path = 20-[a, h, d, e, b, c, a].

?- 
```