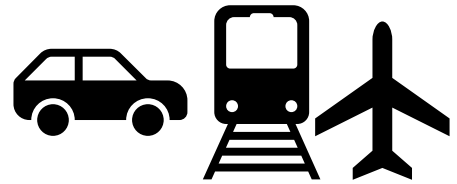

Planning a Travel Journey
19/12/2023

Planning a travel journey

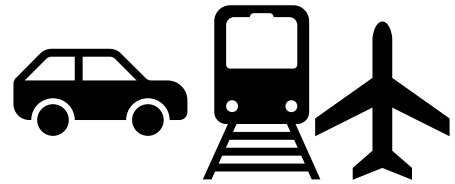


- The project is a program that helps in planning a travel journey from one place to another.

- It determines the transportation means required to travel from one place to another and the intermediate places which are needed to pass through when getting from one place to another

- The program’s knowledge base consists of a group of different places (10 places) with different dimensions, the means of transportation available to move between these places (3 transport means) , and the different paths between these places and each other.

The program's knowledge base



Places

- Cairo
- Alex
- Read sea
- Aswan
- Menofia
- Hurgada
- Matrouh
- Port said
- Sainai
- Luxor

Transports

- car
- train
- plane

```
2  % Knowledge base
3  place('Cairo').
4  place('Alex').
5  place('Read sea').
6  place('Aswan').
7  place('Menofia').
8  place('Hurgada').
9  place('Matrouh').
10 place('Port said').
11 place('Sainai').
12 place('Luxor').
13
14 transport(car).
15 transport(train).
16 transport(plane).
```

Connection

All paths to get to and from different places have been expressed in terms of the facts show start , the destination, and the means of transportation that will be relied upon in the rules used in the program.

```
17
18 connection('Cairo','Alex',car).
19 connection('Cairo','Read sea',plane).
20 connection('Cairo','Aswan',train).
21 connection('Cairo','Menofia',car).
22 connection('Cairo','Hurgada',plane).
23 connection('Cairo','Matrouh',train).
24 connection('Cairo','Port said',plane).
25 connection('Cairo','Sainai',plane).
26 connection('Cairo','Luxor',train).
27
28 connection('Alex','Cairo',car).
29 connection('Alex','Read sea',plane).
30 connection('Alex','Aswan',train).
31 connection('Alex','Menofia',car).
32 connection('Alex','Hurgada',plane).
33 connection('Alex','Matrouh',train).
34 connection('Alex','Port said',plane).
35 connection('Alex','Sainai',plane).
36 connection('Alex','Luxor',plane).
37
38 connection('Read sea','Cairo',plane).
39 connection('Read sea','Alex',plane).
40 connection('Read sea','Aswan',train).
41 connection('Read sea','Hurgada',car).
```

```
47 connection('Aswan','Cairo',train).
48 connection('Aswan','Alex',train).
49 connection('Aswan','Read sea',train).
50 connection('Aswan','Hurgada',train).
51 connection('Aswan','Matrouh',plane).
52 connection('Aswan','Port said',plane).
53 connection('Aswan','Sainai',plane).
54 connection('Aswan','Luxor',car).
55
56 connection('Menofia','Cairo',car).
57 connection('Menofia','Alex',car).
58
59 connection('Hurgada','Cairo',plane).
60 connection('Hurgada','Alex',plane).
61 connection('Hurgada','Read sea',car).
62 connection('Hurgada','Aswan',train).
63 connection('Hurgada','Port said',car).
64 connection('Hurgada','Sainai',car).
65 connection('Hurgada','Luxor',train).
66
67 connection('Matrouh','Cairo',train).
68 connection('Matrouh','Alex',train).
69 connection('Matrouh','Read sea',plane).
70 connection('Matrouh','Aswan',plane).
71 connection('Matrouh','Hurgada',plane).
```

Rules

The rules used to determine the journey path are divided into 3 rules :

1. **can_travel_directly(Start, End, Transport)**

This rule for direct travel determines the directness by searching the facts of the program's knowledge base.

It searches for the appropriate connection directly between the **start** and the **destination**.

```
% Rules for direct travel
can_travel_directly(Start, End, Transport) :-
    connection(Start, End, Transport).
```

2. can_travel_indirectly(Start, End, Transports)

this for indirect travel determined by searching for a connection between the **start** and an **intermediate** place via a specific means of transportation Then you call the first direct rule to search for a connection between the place, the **intermediate**, and the **destination** through another means

```
% Rules for indirect travel
can_travel_indirectly(Start, End, Transports) :-
    connection(Start, Intermediate, Transport1),
    can_travel_directly(Intermediate, End, Transport2),
    Transports = [Transport1,Intermediate, Transport2].
```

3. can_travel(Start, End, Transports)

Predicate to check if it is possible to travel between two places It calls the first rule to check whether there is a direct path between the start and the end
If verified, it prints (there is a direct path) and prints the means of transportation used
Otherwise, it calls the second rule to check if there is an indirect path and prints (there is an indirect path) along with finding the possible paths.
If the first and second rules are not met, the query is **invalid**

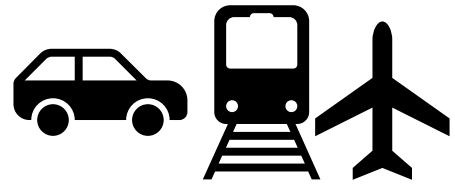
```
can_travel(Start, End, Transports) :-
    can_travel_directly(Start, End, Transports),
    writeln('Direct connection available!'),
    writeln(Transports).

can_travel(Start, End, Transports) :-
    can_travel_indirectly(Start, End, Transports),
    writeln('Indirect connection available!'),
    writeln( Transports).
```

4. is_compliant(Operation) :-

Predicate to check if an operation is compliant with the database

```
% Predicate to check if an operation is compliant with the database
is_compliant(Operation) :-
    call(Operation), !, write('Operation is compliant.'), nl.
is_compliant(_) :-
    write('Operation is not compliant.'), nl.
```

Tracing



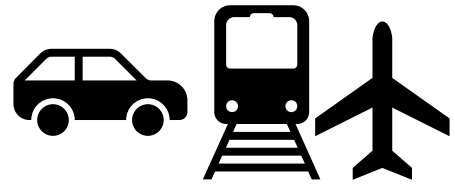
? can_travel('Cairo', 'Alex', car).

Tracing

```
?- can_travel('Cairo','Alex',car).
Direct connection available!
car

?- trace.

[trace]  ?- can_travel('Cairo','Alex',car).
   Call: (10) can_travel('Cairo', 'Alex', car) ? creep
   Call: (11) can_travel_directly('Cairo', 'Alex', car) ? creep
   Call: (12) connection('Cairo', 'Alex', car) ? creep
   Exit: (12) connection('Cairo', 'Alex', car) ? creep
   Exit: (11) can_travel_directly('Cairo', 'Alex', car) ? creep
   Call: (11) writeln('Direct connection available!') ? creep
Direct connection available!
   Exit: (11) writeln('Direct connection available!') ? creep
   Call: (11) writeln(car) ? creep
car
   Exit: (11) writeln(car) ? creep
   Exit: (10) can_travel('Cairo', 'Alex', car) ? creep
```

? can_travel('Cairo', 'Menofia', plane).

Tracing

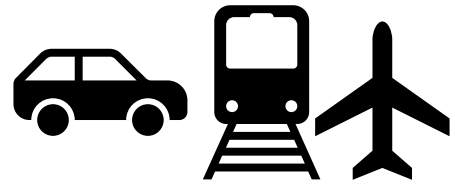


```
?- can_travel('Cairo', 'Menofia', plane).
false.

?- trace.

[trace]  ?- can_travel('Cairo', 'Menofia', plane).
   Call: (10) can_travel('Cairo', 'Menofia', plane) ? creep
   Call: (11) can_travel_directly('Cairo', 'Menofia', plane) ? creep
   Call: (12) connection('Cairo', 'Menofia', plane) ? creep
   Fail: (12) connection('Cairo', 'Menofia', plane) ? creep
   Fail: (11) can_travel_directly('Cairo', 'Menofia', plane) ? creep
   Redo: (10) can_travel('Cairo', 'Menofia', plane) ? creep
   Call: (11) can_travel_indirectly('Cairo', 'Menofia', plane) ? creep
   Call: (12) connection('Cairo', _15190, _15192) ? creep
   Exit: (12) connection('Cairo', 'Alex', car) ? creep
   Call: (12) can_travel_directly('Alex', 'Menofia', _15280) ? creep
   Call: (13) connection('Alex', 'Menofia', _15324) ? creep
   Exit: (13) connection('Alex', 'Menofia', car) ? creep
   Exit: (12) can_travel_directly('Alex', 'Menofia', car) ? creep
   Call: (12) plane=[car, 'Alex', car] ? creep
   Fail: (12) plane=[car, 'Alex', car] ? creep
   Redo: (12) connection('Cairo', _15560, _15562) ? creep
   Exit: (12) connection('Cairo', 'Read sea', plane) ? creep
   Call: (12) can_travel_directly('Read sea', 'Menofia', _15650) ? creep
```

```
   Fail: (12) plane=[car, 'Alex', car] ? creep
   Redo: (12) connection('Cairo', _15560, _15562) ? creep
   Exit: (12) connection('Cairo', 'Read sea', plane) ? creep
   Call: (12) can_travel_directly('Read sea', 'Menofia', _15650) ? creep
   Call: (13) connection('Read sea', 'Menofia', _15694) ? creep
   Fail: (13) connection('Read sea', 'Menofia', _15738) ? creep
   Fail: (12) can_travel_directly('Read sea', 'Menofia', _15782) ? creep
   Redo: (12) connection('Cairo', _15824, _15826) ? creep
   Exit: (12) connection('Cairo', 'Aswan', train) ? creep
   Call: (12) can_travel_directly('Aswan', 'Menofia', _15914) ? creep
   Call: (13) connection('Aswan', 'Menofia', _15958) ? creep
   Fail: (13) connection('Aswan', 'Menofia', _16002) ? creep
   Fail: (12) can_travel_directly('Aswan', 'Menofia', _16046) ? creep
   Redo: (12) connection('Cairo', _16088, _16090) ? creep
   Exit: (12) connection('Cairo', 'Menofia', car) ? creep
   Call: (12) can_travel_directly('Menofia', 'Menofia', _16178) ? creep
   Call: (13) connection('Menofia', 'Menofia', _16222) ? creep
   Fail: (13) connection('Menofia', 'Menofia', _16266) ? creep
   Fail: (12) can_travel_directly('Menofia', 'Menofia', _16310) ? creep
   Redo: (12) connection('Cairo', _16352, _16354) ? creep
   Exit: (12) connection('Cairo', 'Hurgada', plane) ? creep
   Call: (12) can_travel_directly('Hurgada', 'Menofia', _16442) ? creep
   Call: (13) connection('Hurgada', 'Menofia', _16486) ? creep
   Fail: (13) connection('Hurgada', 'Menofia', _16530) ? creep
   Fail: (12) can_travel_directly('Hurgada', 'Menofia', _16574) ? creep
```

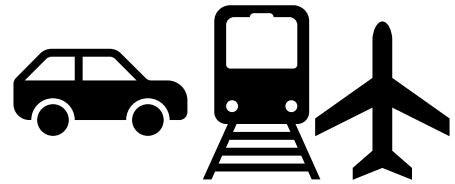
? can_travel('Menofia', 'Alex', Transport).

Tracing



```
?- can_travel('Menofia','Alex',Transport).
Direct connection available!
car
Transport = car ;
Indirect connection available!
[car,Cairo,car]
Transport = [car, 'Cairo', car] ;
false.

?- trace.

[trace]  ?- can_travel('Menofia','Alex',Transport).
   Call: (10) can_travel('Menofia', 'Alex', _15904) ? creep
   Call: (11) can_travel_directly('Menofia', 'Alex', _15904) ? creep
   Call: (12) connection('Menofia', 'Alex', _15904) ? creep
   Exit: (12) connection('Menofia', 'Alex', car) ? creep
   Exit: (11) can_travel_directly('Menofia', 'Alex', car) ? creep
   Call: (11) writeln('Direct connection available!') ? creep
Direct connection available!
   Exit: (11) writeln('Direct connection available!') ? creep
   Call: (11) writeln(car) ? creep
car
   Exit: (11) writeln(car) ? creep
   Exit: (10) can_travel('Menofia', 'Alex', car) ? creep
Transport = car ;
   Redo: (10) can_travel('Menofia', 'Alex', _15904) ? creep
   Call: (11) can_travel_indirectly('Menofia', 'Alex', _15904) ? creep
   Call: (12) connection('Menofia', _17658, _17660) ? creep
   Exit: (12) connection('Menofia', 'Cairo', car) ? creep
   Call: (12) can_travel_directly('Cairo', 'Alex', _17748) ? creep
   Call: (13) connection('Cairo', 'Alex', _17792) ? creep
   Exit: (13) connection('Cairo', 'Alex', car) ? creep
   Exit: (12) can_travel_directly('Cairo', 'Alex', car) ? creep
   Call: (12) _15904=[car, 'Cairo', car] ? creep
   Exit: (12) [car, 'Cairo', car]=[car, 'Cairo', car] ? creep
   Exit: (11) can_travel_indirectly('Menofia', 'Alex', [car, 'Cairo', car]) ? creep
   Call: (11) writeln('Indirect connection available!') ? creep
Indirect connection available!
   Exit: (11) writeln('Indirect connection available!') ? creep
   Call: (11) writeln([car, 'Cairo', car]) ? creep
[car,Cairo,car]
   Exit: (11) writeln([car, 'Cairo', car]) ? creep
   Exit: (10) can_travel('Menofia', 'Alex', [car, 'Cairo', car]) ? creep
Transport = [car, 'Cairo', car] ;
   Redo: (12) connection('Menofia', _19118, _19120) ? creep
   Exit: (12) connection('Menofia', 'Alex', car) ? creep
   Call: (12) can_travel_directly('Alex', 'Alex', _19208) ? creep
   Call: (13) connection('Alex', 'Alex', _19252) ? creep
   Fail: (13) connection('Alex', 'Alex', _19296) ? creep
   Fail: (12) can_travel_directly('Alex', 'Alex', _19340) ? creep
   Fail: (11) can_travel_indirectly('Menofia', 'Alex', _15904) ? creep
   Fail: (10) can_travel('Menofia', 'Alex', _15904) ? creep
false.
```

? - is_compliant(can_travel('Menofia','Alex',Transport)).



```
?- is_compliant(can_travel('Menofia','Alex',Transport)).
Direct connection available!
car
Operation is compliant.
Transport = car ;
false.
```

? - is_compliant(can_travel('Menofia','Alex',plane)).

```
?- is_compliant(can_travel('Menofia','Alex',plane)).
Operation is not compliant.

?-
```

Tracing

```
[trace]  ?- is_compliant(can_travel('Menofia','Alex',plane)).
   Call: (10) is_compliant(can_travel('Menofia', 'Alex', plane)) ? creep
   Call: (11) can_travel('Menofia', 'Alex', plane) ? creep
   Call: (12) can_travel_directly('Menofia', 'Alex', plane) ? creep
   Call: (13) connection('Menofia', 'Alex', plane) ? creep
   Fail: (13) connection('Menofia', 'Alex', plane) ? creep
   Fail: (12) can_travel_directly('Menofia', 'Alex', plane) ? creep
   Redo: (11) can_travel('Menofia', 'Alex', plane) ? creep
   Call: (12) can_travel_indirectly('Menofia', 'Alex', plane) ? creep
   Call: (13) connection('Menofia', _17472, _17474) ? creep
   Exit: (13) connection('Menofia', 'Cairo', car) ? creep
   Call: (13) can_travel_directly('Cairo', 'Alex', _17562) ? creep
   Call: (14) connection('Cairo', 'Alex', _17606) ? creep
   Exit: (14) connection('Cairo', 'Alex', car) ? creep
   Exit: (13) can_travel_directly('Cairo', 'Alex', car) ? creep
   Call: (13) plane=[car, 'Cairo', car] ? creep
   Fail: (13) plane=[car, 'Cairo', car] ? creep
   Redo: (13) connection('Menofia', _17842, _17844) ? creep
   Exit: (13) connection('Menofia', 'Alex', car) ? creep
   Call: (13) can_travel_directly('Alex', 'Alex', _17932) ? creep
   Call: (14) connection('Alex', 'Alex', _17976) ? creep
   Fail: (14) connection('Alex', 'Alex', _18020) ? creep
   Fail: (13) can_travel_directly('Alex', 'Alex', _18064) ? creep
   Fail: (12) can_travel_indirectly('Menofia', 'Alex', plane) ? creep
   Fail: (11) can_travel('Menofia', 'Alex', plane) ? creep
   Redo: (10) is_compliant(can_travel('Menofia', 'Alex', plane)) ? creep
   Call: (11) write('Operation is not compliant.') ? creep
```

```
   Fail: (12) can_travel_directly('Menofia', 'Alex', plane) ? creep
   Redo: (11) can_travel('Menofia', 'Alex', plane) ? creep
   Call: (12) can_travel_indirectly('Menofia', 'Alex', plane) ? creep
   Call: (13) connection('Menofia', _17472, _17474) ? creep
   Exit: (13) connection('Menofia', 'Cairo', car) ? creep
   Call: (13) can_travel_directly('Cairo', 'Alex', _17562) ? creep
   Call: (14) connection('Cairo', 'Alex', _17606) ? creep
   Exit: (14) connection('Cairo', 'Alex', car) ? creep
   Exit: (13) can_travel_directly('Cairo', 'Alex', car) ? creep
   Call: (13) plane=[car, 'Cairo', car] ? creep
   Fail: (13) plane=[car, 'Cairo', car] ? creep
   Redo: (13) connection('Menofia', _17842, _17844) ? creep
   Exit: (13) connection('Menofia', 'Alex', car) ? creep
   Call: (13) can_travel_directly('Alex', 'Alex', _17932) ? creep
   Call: (14) connection('Alex', 'Alex', _17976) ? creep
   Fail: (14) connection('Alex', 'Alex', _18020) ? creep
   Fail: (13) can_travel_directly('Alex', 'Alex', _18064) ? creep
   Fail: (12) can_travel_indirectly('Menofia', 'Alex', plane) ? creep
   Fail: (11) can_travel('Menofia', 'Alex', plane) ? creep
   Redo: (10) is_compliant(can_travel('Menofia', 'Alex', plane)) ? creep
   Call: (11) write('Operation is not compliant.') ? creep
Operation is not compliant.
   Exit: (11) write('Operation is not compliant.') ? creep
   Call: (11) nl ? creep

   Exit: (11) nl ? creep
   Exit: (10) is_compliant(can_travel('Menofia', 'Alex', plane)) ? creep
```

Thank you