

Write a Prolog Program for PLANETS DB.

AIM:

To write a Prolog program that stores and retrieves information about planets, including their distance from the Sun, number of moons, and type.

ALGORITHM:

1. Start the program.
2. Define facts in the form `planet(Name, DistanceFromSun, Moons, Type)`.
3. Load the program into the Prolog interpreter.
4. Query the database to retrieve details based on planet name, type, or conditions (e.g., `moons > 10`).
5. Prolog will return the matching results.
6. Stop.

File Edit View

```
% Facts: planet(Name, DistanceFromSun, Moons, Type).  
  
planet(mercury, 57.9, 0, terrestrial).  
planet(venus, 108.2, 0, terrestrial).  
planet(earth, 149.6, 1, terrestrial).  
planet(mars, 227.9, 2, terrestrial).  
planet(jupiter, 778.5, 79, gas_giant).  
planet(saturn, 1434, 83, gas_giant).  
planet(uranus, 2871, 27, ice_giant).  
planet(neptune, 4495, 14, ice_giant).
```

File Edit Settings Run Debug Help

```
?-
% c:/Users/gayathri/Downloads/planet.pl compiled 0.00 sec, 8 clauses
?- planet(Name, Distance, Moons, Type).
Name = mercury,
Distance = 57.9,
Moons = 0,
Type = terrestrial ;
Name = venus,
Distance = 108.2,
Moons = 0,
Type = terrestrial ;
Name = earth,
Distance = 149.6,
Moons = 1,
Type = terrestrial ;
Name = mars,
Distance = 227.9,
Moons = 2,
Type = terrestrial ;
Name = jupiter,
Distance = 778.5,
Moons = 79,
Type = gas_giant ;
Name = saturn,
Distance = 1434,
Moons = 83,
Type = gas_giant ;
Name = uranus,
Distance = 2871,
Moons = 27,
Type = ice_giant ;
Name = neptune,
Distance = 4495,
Moons = 14,
Type = ice_giant.

?- planet(Name, _, Moons, _), Moons > 10.
Name = jupiter,
Moons = 79 ;
Name = saturn,
Moons = 83 ;
Name = uranus,
Moons = 27 ;
Name = neptune,
Moons = 14.

?- planet(Name, _, _, terrestrial).
Name = mercury ;
Name = venus ;
Name = earth ;
Name = mars.

?
```

RESULT:

The program successfully stores and retrieves planetary information such as name, distance, moons, and type from the knowledge base.