· **Family Tree Relationship Finder (1-8)**

* Implement a simple family tree where you can define relationships (like parent, child, sibling, etc.).
* Example Queries: "Who is John's father?", "Who are the siblings of Mary?"

· **Route Finder (9-16)**

* Given a set of connected cities, Prolog can be used to find a route between two cities.
* Example Queries: "Is there a route between City A and City B?", "What is the shortest path from City A to City B?"

· **Expert System for Medical Diagnosis (17-24)**

* You can create an expert system that suggests possible diagnoses based on symptoms provided by the user.
* Example Queries: "What could be the disease if a person has fever and headache?"

· **Simple Puzzle Solver (like Sudoku or N-Queens) (25-32)**

* Implement logic to solve simple puzzles like Sudoku or the N-Queens problem, where you need to place N queens on a chessboard so that no two queens attack each other.

· **Library Book Recommendation System (33-40)**

* You can create a system to recommend books based on user preferences (genre, author, etc.) or history of borrowing.
* Example Queries: "What are good science fiction books?", "Suggest a book by Agatha Christie."

· **Animal Identification Game (41-48)**

* Create a small game where Prolog asks questions to identify an animal based on user responses about characteristics.
* Example Query: "Is the animal a mammal?", followed by further narrowing questions.

· **Scheduling System (49-56)**

* Implement a simple scheduling system where constraints are used to decide if a set of tasks or meetings can fit into a day without conflicts.
* Example Queries: "Can meeting A and meeting B happen on the same day?"

· **Travel Recommendation System (57-64)**

* Based on preferences like climate, distance, and budget, you can recommend travel destinations.
* Example Query: "Suggest a travel destination with a warm climate and within budget X."

· **Food Recommendation Based on Ingredients (65-72)**

* Create a system that suggests recipes based on the ingredients a user has.
* Example Queries: "What can I cook with tomatoes and pasta?"

· **Simple Employee Management System (73-80)**

* A system to store employee details and query about them, such as finding employees with certain skills or their managers.
* Example Queries: "Who is the manager of Alice?", "Who are the employees with Python skills?"