

Prolog viva

1. What is prolog?

Prolog stands for **Programming in Logic**.

It is a high-level **logic programming language** associated primarily with **artificial intelligence** and **computational linguistics**. Prolog is based on formal logic and allows expressing facts and rules about problems within a system using a form of symbolic reasoning.

2. Name one language which can replace Prolog in AI applications.

LISP

3. Which version of Prolog have you used?

GNU Prolog is a free Prolog compiler with constraint solving over finite domains developed by Daniel Diaz.

4. Write features of prolog

- Prolog is a **logic programming language** based on **first-order predicate logic**.
- It uses **facts, rules, and queries** to represent and solve problems.
- Backtracking** and **pattern matching** are core mechanisms for execution.
- It is well-suited for applications in **AI, natural language processing, and expert systems**.
- Prolog supports **recursion** and **non-deterministic programming** naturally.

5. What are the main constructs in Prolog?

Main Constructs in Prolog (point-wise):

- Facts** – Declare known information (e.g., `parent(john, mary).`).
- Rules** – Define relationships using conditions (e.g., `grandparent(X, Y) :- parent(X, Z), parent(Z, Y).`).
- Queries** – Ask questions to the system (e.g., `?- parent(john, mary).`).
- Variables** – Represent unknowns, written with capital letters (e.g., `X, Y`).
- Predicates** – Functions that define relationships between arguments.

6. Give one example of predicate.

- `father(john, mary).` – *john is the father of mary*
- `likes(alice, pizza).` – *alice likes pizza*
- `sibling(X, Y) :- parent(Z, X), parent(Z, Y).` – *X and Y are siblings if they share a parent Z*
- `greater_than(X, Y) :- X > Y.` – *X is greater than Y*
- `is_even(N) :- 0 is N mod 2.` – *N is even*

7. What is list in Prolog. Give Example.

- A **list** is an ordered collection of elements.
- It is written using **square brackets**, e.g., `[a, b, c]`.
- Elements can be **atoms, numbers, variables**, or even other lists.
- The **head** is the first element; the **tail** is the rest of the list.
- Example: `[apple, banana, mango]` or `[H|T]` for head-tail pattern matching.
`[apple, banana, mango] = [H|T]`.

Result:

- `H = apple` (Head)
- `T = [banana, mango]` (Tail)

8. what are the list operations can be performed in prolog. Give examples

- Concatenation** – Join two lists.
 Example: `append([1,2], [3,4], X). % X = [1,2,3,4]`
- Membership** – Check if an element is in a list.
 Example: `member(3, [1,2,3,4]). % true`
- Length** – Find the number of elements.
 Example: `length([a,b,c], L). % L = 3`
- Head and Tail** – Extract first element and rest.
 Example: `[H|T] = [x,y,z]. % H = x, T = [y,z]`
- Delete** – Remove an element from list.
 Example: `select(b, [a,b,c], X). % X = [a,c]`

9. State the problem statement of the Travelling Salesman Problem (TSP).

Find the shortest possible route that visits each city once and returns to the starting city.

Example: Given cities A, B, C, D with distances, find the minimum-cost tour covering all.

10. State the Water Jug Problem.

Measure a specific quantity of water using two jugs of different capacities.

Example: Using a 4-liter and a 3-liter jug, measure exactly 2 liters.

11. State the Monkey and Banana Problem.

A monkey wants to reach bananas hanging from the ceiling using tools in the room.

Example: Monkey must move a box under bananas and climb to grab them.

12. State the 4-Queens Problem.

A: Place 4 queens on a 4×4 chessboard so that no two queens attack each other.

Example: Ensure no two queens share the same row, column, or diagonal.

13. How recursion is implemented in Prolog? Give one example.

- Prolog uses recursion** to repeat operations, especially in list processing and mathematical problems.
- A **base case** stops recursion.
- A **recursive rule** calls itself with a smaller or simpler input.

```
factorial(0, 1).                % Base case
factorial(N, F) :-
    N > 0,
    N1 is N - 1,
    factorial(N1, F1),
    F is N * F1.                % Recursive step
```

Query:

?- factorial(3, F). → F = 6

14. Do we have loops in Prolog? What can be useful in that case? write in crisp and point-wise

- a. **Prolog does not have traditional loops** like `for` or `while`.
- b. Instead, **recursion** is used to perform repeated actions.

15. What are the conditional statements are available in Prolog? Give one example.

- a) **If-Then (->)** – Executes the "then" part if the condition is true.
Example: `X > 0 -> write('Positive').`
- b) **If-Then-Else (-> ;)** – Adds an "else" part.
Example: `X > 0 -> write('Positive') ; write('Non-positive').`
- c) **Cut (!) with conditions** – used to control backtracking in Prolog.

Types of Cuts:

1. Green Cut

- Used for **efficiency** only; doesn't change logical meaning.
- Just **prevents unnecessary backtracking**.
- ☐ Safe to use.

Example

```
max(X, Y, X) :- X >= Y, !.      % Green cut
max(_, Y, Y).
```

2. Red Cut

- Changes the **logical meaning** of the program.
- Used to **force a decision** and prune alternatives.
- ☐ Can lead to incorrect logic if used carelessly.

Example

```
grade(Marks, pass) :- Marks >= 40, !.
grade(_, fail).      % Red cut forces decision at 40
```

