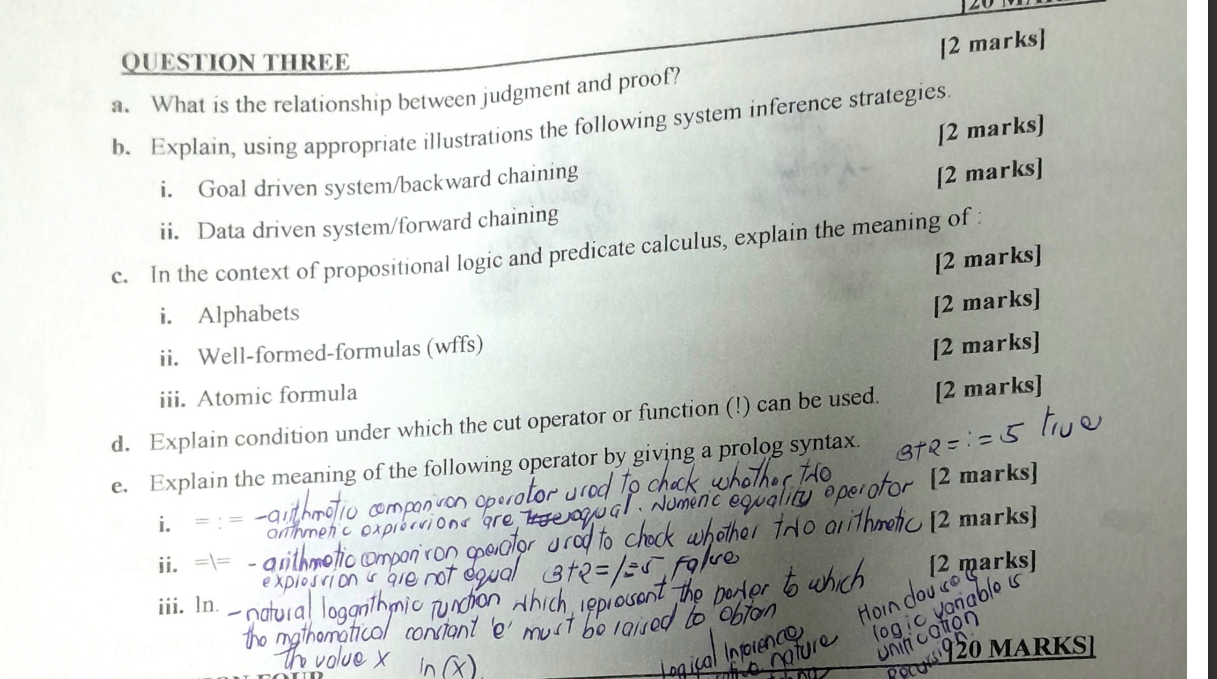
* **Backtracking** refers to the process of exploring alternative solutions to a problem when the initial solution fails.
* This backtracking process continues until a solution is found or all possibilities have been exhausted. The order in which goals are evaluated can affect the efficiency and effectiveness of the search process.
* **Unification** is the process of finding substitutions for variables to make predicates true, and evaluation order refers to the order in which goals or predicates are evaluated.
* The **evaluation** order of predicates such as checking if a route exists, checking for obstacles, and determining the shortest path can significantly impact the time taken to find a solution.
* backtracking, unification, and evaluation order are fundamental concepts in logic programming that play a crucial role in the search for solutions to problems.

**Question THREE**

****

**a. The relationship between judgment and pronoun lies in syntax and grammar, where pronouns replace nouns, and judgment involves forming opinions or decisions.**

**b.**

**i. Goal-driven system/backward chaining involves starting with the goal and working backward to find the necessary conditions. Illustration: Solving a puzzle by identifying the final arrangement first.**

**ii. Data-driven system/forward chaining starts with available data, making inferences step by step. Illustration: Decision-making process based on real-time information.**

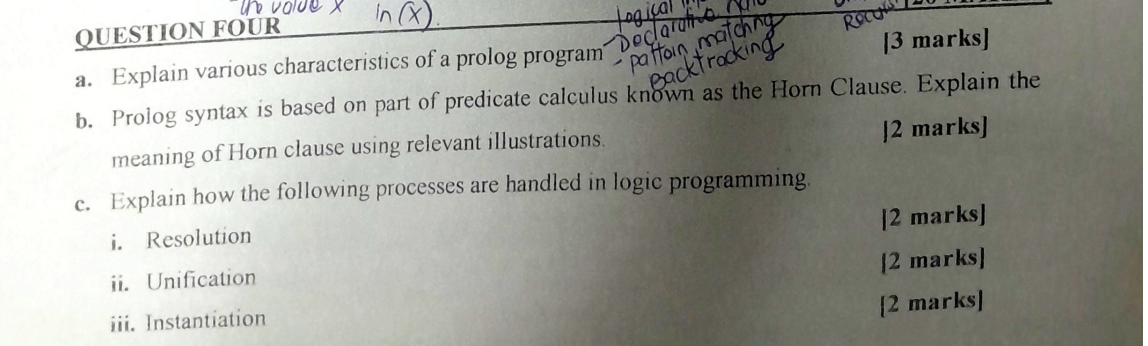
**c.**

**i. Alphabets in propositional logic and predicate calculus refer to variables representing statements. Example: P, Q, R.**

**ii. Well-formed formulas (wffs) are syntactically correct statements. Example: (P ∧ Q) → R.**

**iii. Atomic formula is a basic, indivisible statement. Example: P.d. The cut operator (!) can be used when a Prolog program needs to prevent backtracking beyond a certain point, ensuring a more efficient search space.**

**Question FOUR**

****

**a. Various characteristics of a Prolog program include declarative nature, rule-based programming, use of facts and rules, backward chaining for goal resolution, and the ability to handle non-deterministic situations.**

**b. The Horn clause in Prolog syntax is a logical implication in the form of "Head :- Body," where the Head is a goal to be achieved, and the Body contains conditions for achieving the goal. Illustration: `parent(X, Y) :- father(X, Y), mother(X, Y)`.**

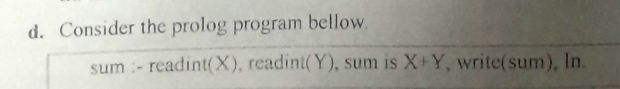
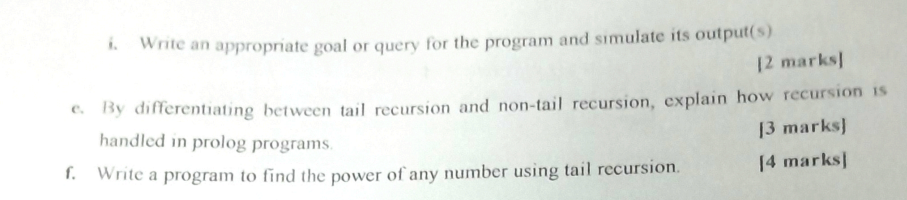
**c.**

**i. Resolution in logic programming involves resolving conflicting goals or clauses. Illustration: Resolving a query by combining relevant rules.**

**ii. Unification is the process of making two terms identical by assigning values to variables. Illustration: Unifying `parent(X, Y)` and `parent(John, Mary)` results in `X = John` and `Y = Mary`.**

**iii. Instantiation is the assignment of specific values to variables during program**

**execution. Illustration: Instantiating a variable in a rule when a query is satisfied.**

**** ****

**?- sum.**

**% This will prompt the user to enter two integers, and then print the sum of those integers.**

**?- sum, sum.**

**% This will prompt the user to enter two integers twice, and print the sum of each pair.**

**?- sum, X is sum.**

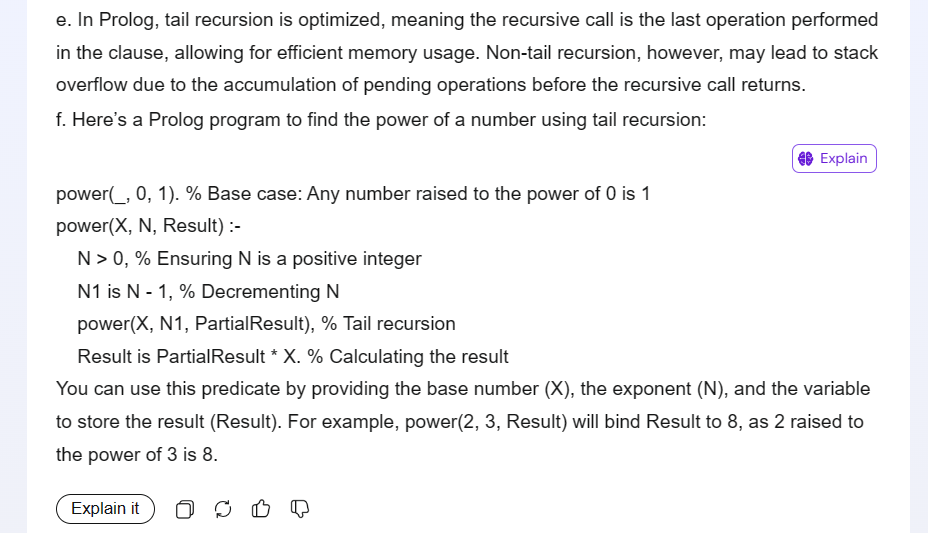
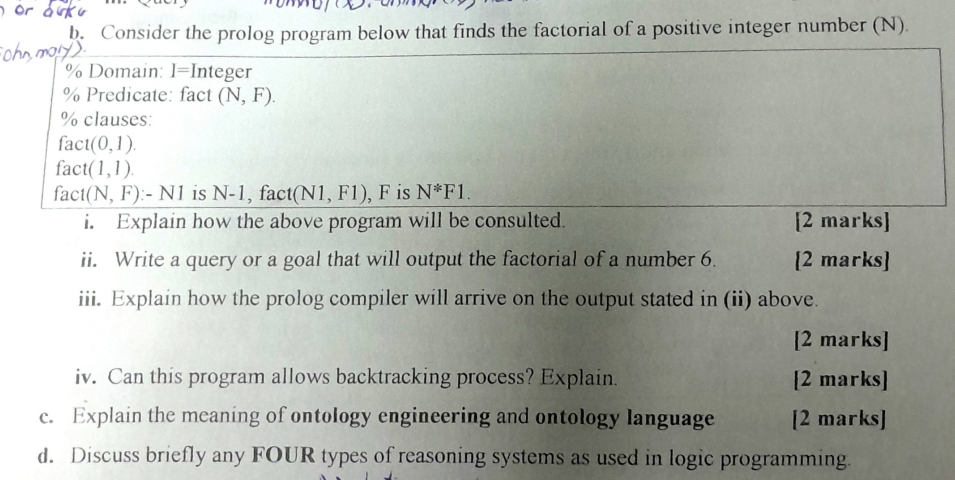
**% This will prompt the user to enter two integers, and then unify the variable X with their sum.**

**?- sum, Y = 5 + X, X is sum.**

**% This will prompt the user to enter two integers, add them together to get the sum, add 5 to the sum to get X, and then unify X with the term 5 + Y.**

**?- sum, Y = 5 + X, X is sum, Z is Y \* 2.**

**% This will prompt the user to enter two integers, add them together to get the sum, add 5 to the sum to get X, unify X with the term 5 + Y, multiply Y by 2 to get Z.**

**** **** b.

i. The Prolog program is consulting a fact `fact(0,1)` and a rule `fact(N, F) :- N > 0, N1 is N - 1, fact(N1, F1), F is N \* F1`. It calculates the factorial of a number `N`.

ii. Query: `fact(6, F).`

iii. The Prolog compiler will arrive at the output by unifying the query `fact(6, F)` with the rule `fact(N, F) :- N > 0, N1 is N - 1, fact(N1, F1), F is N \* F1`, then recursively applying the rule until it reaches the base case.

iv. Yes, this program allows backtracking. Prolog's backtracking mechanism allows for exploring alternative solutions, so if the user requests more solutions, Prolog will backtrack to find them.

c. Ontology engineering involves creating explicit formal specifications of the concepts, relationships, and entities within a particular domain. An ontology language is a formal language used to define ontologies, providing syntax and semantics for expressing knowledge about a domain.

d. 1. Deductive reasoning: Deriving logical conclusions from given premises using rules of inference.

2. Inductive reasoning: Inferring general rules or patterns from specific observations or examples.

3. Abductive reasoning: Generating hypotheses to explain observed phenomena or data.

4. Default reasoning: Making assumptions or drawing conclusions based on typical or default cases when complete information is unavailable.