Explanation for the code:

**Overview of logic.py**

Purpose: Provides a framework for symbolic logic, enabling the representation and evaluation of logical sentences.

Core Components: Sentence Class: Abstract base class for all logical constructs.

Symbol: Represents individual logical variables

Not: Represents the negation of a sentence.

And/Or/Implication/Biconditional: Represent their respective logical operations.

Evaluation: Each class implements an evaluate method to determine truth values based on a model (mapping of symbols to truth values).

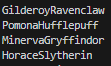
Model Checking: The model\_check function recursively checks if a set of statements (knowledge base) entails a query.

**Overview of Other Files**

1. clue.py:Simulates deduction similar to the game Clue, identifying valid combinations of characters, rooms, and weapons based on a knowledge base.Implements logical constraints and evaluates possible solutions using the check\_knowledge function.



1. harry.py:Implements a logical inference system to determine if a query is entailed by a knowledge base.Sets up a knowledge base involving characters and checks logical entailment for specific queries.
2. Other Files:Solve logic puzzles, such as Mastermind configurations and character assignments to houses, using constraints defined in the knowledge base.



**Key Takeaways**The structure allows for flexible representation of logical statements and facilitates reasoning in various logical puzzles and inference tasks.By utilizing logical operators and recursive checking, the code simulates deduction processes effectively, making it applicable for various reasoning scenarios in artificial intelligence.