🧠 Explain Mode — Easy Questions

Concepts (single-turn)

1. What is a variable in Python?
2. What is a list in Python?
3. What is a dictionary?
4. What is a function?
5. What is a loop?
6. What is recursion?
7. What is the difference between list and tuple?
8. What is OOP in simple terms?

Follow-up (tests memory)

Ask in order:

1. What is a function?
2. Give a simple example
3. Why is it useful?

Then:

4. Can you explain it again in one sentence?

This checks:

- conversation summary
- context reuse

Programming understanding

1. What does this code do in Python: `len("hello")`
2. Why do we use return in functions?
3. What is indentation in Python?

4. What is the difference between for and while?

💻 Generate Mode — Easy Code Tasks

Very basic

1. Write a Python function to add two numbers
2. Write a function to check if a number is even
3. Write a function to reverse a string
4. Write a function to find the maximum of a list
5. Write a function to count vowels in a string

Slightly structured

6. Write a Python function to check if a number is prime
7. Write a function to remove duplicates from a list
8. Write a function to sort a list
9. Write a function to compute factorial
10. Write a function to check palindrome

With constraints (tests RAG grounding)

11. Write a Python function with type hints to sum a list
12. Write a function with a docstring and example
13. Write a function that uses recursion to compute Fibonacci

🔀 Router Edge Cases (very important)

These test classification quality:

1. explain prime number
2. prime number python function
3. how to write a loop
4. loop example in python
5. what is sorting
6. write code for sorting

Expected:

- explain → conceptual
- generate → implementation

🧪 Mixed Conversation Test (real scenario)

Run in this exact order:

1. What is a list?
2. Give an example
3. Write a Python function to reverse a list
4. Explain how it works
5. Now write a function to remove duplicates

This validates:

- mode switching
- GPU unload/load
- memory isolation
- RAG reuse

⭐ Unknown Knowledge Test (for self-learning)

Use something **not in HumanEval**:

1. Write a function to validate an email format
2. Write a function to parse a CSV line manually

Should trigger:
low similarity → learning flow (if enabled).

🧩 Ultra-fast smoke test (before demo)

Use just these 6:

1. What is a variable?
2. Give an example
3. Write a Python function to add two numbers
4. Write a function to reverse a string
5. Explain recursion
6. Write a recursive factorial

If these pass → system is stable.

🏁 If you want a benchmark-style test set

I can give you:

- a scored evaluation checklist
- expected router label
- expected behavior per query

so you can demo the system like a product, not just a script.