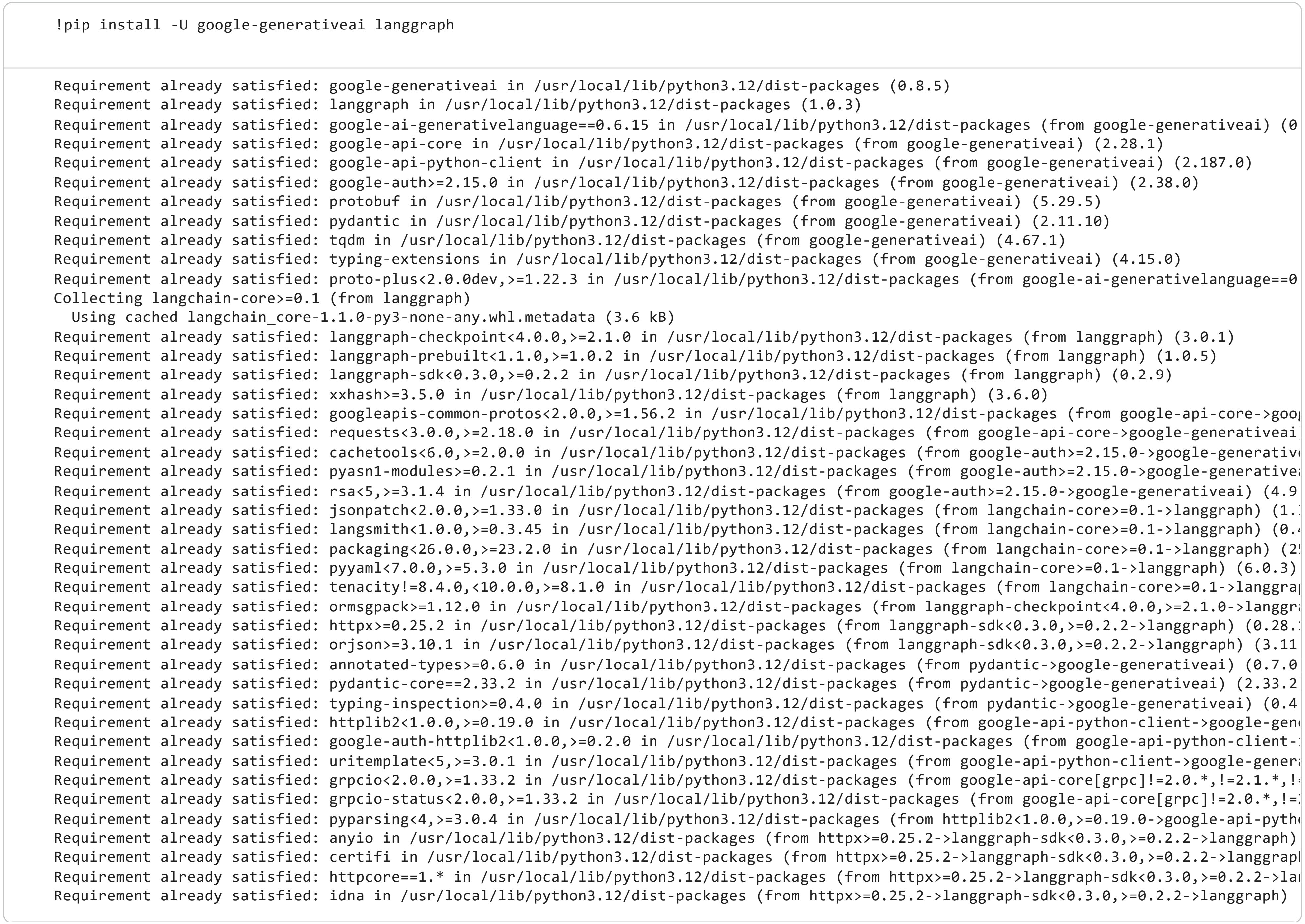

```
!pip install -U google-generativeai langgraph
```

```
Requirement already satisfied: google-generativeai in /usr/local/lib/python3.12/dist-packages (0.8.5)
Requirement already satisfied: langgraph in /usr/local/lib/python3.12/dist-packages (1.0.3)
Requirement already satisfied: google-ai-generativelanguage==0.6.15 in /usr/local/lib/python3.12/dist-packages (from google-generativeai) (0
Requirement already satisfied: google-api-core in /usr/local/lib/python3.12/dist-packages (from google-generativeai) (2.28.1)
Requirement already satisfied: google-api-python-client in /usr/local/lib/python3.12/dist-packages (from google-generativeai) (2.187.0)
Requirement already satisfied: google-auth>=2.15.0 in /usr/local/lib/python3.12/dist-packages (from google-generativeai) (2.38.0)
Requirement already satisfied: protobuf in /usr/local/lib/python3.12/dist-packages (from google-generativeai) (5.29.5)
Requirement already satisfied: pydantic in /usr/local/lib/python3.12/dist-packages (from google-generativeai) (2.11.10)
Requirement already satisfied: tqdm in /usr/local/lib/python3.12/dist-packages (from google-generativeai) (4.67.1)
Requirement already satisfied: typing-extensions in /usr/local/lib/python3.12/dist-packages (from google-generativeai) (4.15.0)
Requirement already satisfied: proto-plus<2.0.0dev,>=1.22.3 in /usr/local/lib/python3.12/dist-packages (from google-ai-generativelanguage==0
Collecting langchain-core>=0.1 (from langgraph)
  Using cached langchain_core-1.1.0-py3-none-any.whl.metadata (3.6 kB)
Requirement already satisfied: langgraph-checkpoint<4.0.0,>=2.1.0 in /usr/local/lib/python3.12/dist-packages (from langgraph) (3.0.1)
Requirement already satisfied: langgraph-prebuilt<1.1.0,>=1.0.2 in /usr/local/lib/python3.12/dist-packages (from langgraph) (1.0.5)
Requirement already satisfied: langgraph-sdk<0.3.0,>=0.2.2 in /usr/local/lib/python3.12/dist-packages (from langgraph) (0.2.9)
Requirement already satisfied: xxhash>=3.5.0 in /usr/local/lib/python3.12/dist-packages (from langgraph) (3.6.0)
Requirement already satisfied: googleapis-common-protos<2.0.0,>=1.56.2 in /usr/local/lib/python3.12/dist-packages (from google-api-core->goo
Requirement already satisfied: requests<3.0.0,>=2.18.0 in /usr/local/lib/python3.12/dist-packages (from google-api-core->google-generativeai
Requirement already satisfied: cachetools<6.0,>=2.0.0 in /usr/local/lib/python3.12/dist-packages (from google-auth>=2.15.0->google-generativ
Requirement already satisfied: pyasn1-modules>=0.2.1 in /usr/local/lib/python3.12/dist-packages (from google-auth>=2.15.0->google-generative
Requirement already satisfied: rsa<5,>=3.1.4 in /usr/local/lib/python3.12/dist-packages (from google-auth>=2.15.0->google-generativeai) (4.9
Requirement already satisfied: jsonpatch<2.0.0,>=1.33.0 in /usr/local/lib/python3.12/dist-packages (from langchain-core>=0.1->langgraph) (1.
Requirement already satisfied: langsmith<1.0.0,>=0.3.45 in /usr/local/lib/python3.12/dist-packages (from langchain-core>=0.1->langgraph) (0.
Requirement already satisfied: packaging<26.0.0,>=23.2.0 in /usr/local/lib/python3.12/dist-packages (from langchain-core>=0.1->langgraph) (2
Requirement already satisfied: pyyaml<7.0.0,>=5.3.0 in /usr/local/lib/python3.12/dist-packages (from langchain-core>=0.1->langgraph) (6.0.3)
Requirement already satisfied: tenacity!=8.4.0,<10.0.0,>=8.1.0 in /usr/local/lib/python3.12/dist-packages (from langchain-core>=0.1->langgra
Requirement already satisfied: ormsgpack>=1.12.0 in /usr/local/lib/python3.12/dist-packages (from langgraph-checkpoint<4.0.0,>=2.1.0->langgr
Requirement already satisfied: httpx>=0.25.2 in /usr/local/lib/python3.12/dist-packages (from langgraph-sdk<0.3.0,>=0.2.2->langgraph) (0.28.
Requirement already satisfied: orjson>=3.10.1 in /usr/local/lib/python3.12/dist-packages (from langgraph-sdk<0.3.0,>=0.2.2->langgraph) (3.11
Requirement already satisfied: annotated-types>=0.6.0 in /usr/local/lib/python3.12/dist-packages (from pydantic->google-generativeai) (0.7.0
Requirement already satisfied: pydantic-core==2.33.2 in /usr/local/lib/python3.12/dist-packages (from pydantic->google-generativeai) (2.33.2
Requirement already satisfied: typing-inspection>=0.4.0 in /usr/local/lib/python3.12/dist-packages (from pydantic->google-generativeai) (0.4
Requirement already satisfied: httplib2<1.0.0,>=0.19.0 in /usr/local/lib/python3.12/dist-packages (from google-api-python-client->google-gen
Requirement already satisfied: google-auth-httplib2<1.0.0,>=0.2.0 in /usr/local/lib/python3.12/dist-packages (from google-api-python-client-
Requirement already satisfied: uritemplate<5,>=3.0.1 in /usr/local/lib/python3.12/dist-packages (from google-api-python-client->google-gener
Requirement already satisfied: grpcio<2.0.0,>=1.33.2 in /usr/local/lib/python3.12/dist-packages (from google-api-core[grpc]!=2.0.*,!=2.1.*,!
Requirement already satisfied: grpcio-status<2.0.0,>=1.33.2 in /usr/local/lib/python3.12/dist-packages (from google-api-core[grpc]!=2.0.*,!=
Requirement already satisfied: pyparsing<4,>=3.0.4 in /usr/local/lib/python3.12/dist-packages (from httplib2<1.0.0,>=0.19.0->google-api-pyth
Requirement already satisfied: anyio in /usr/local/lib/python3.12/dist-packages (from httpx>=0.25.2->langgraph-sdk<0.3.0,>=0.2.2->langgraph)
Requirement already satisfied: certifi in /usr/local/lib/python3.12/dist-packages (from httpx>=0.25.2->langgraph-sdk<0.3.0,>=0.2.2->langgrap
Requirement already satisfied: httpcore==1.* in /usr/local/lib/python3.12/dist-packages (from httpx>=0.25.2->langgraph-sdk<0.3.0,>=0.2.2->la
Requirement already satisfied: idna in /usr/local/lib/python3.12/dist-packages (from httpx>=0.25.2->langgraph-sdk<0.3.0,>=0.2.2->langgraph)
Requirement already satisfied: h11>=0.16 in /usr/local/lib/python3.12/dist-packages (from httpcore==1.*->httpx>=0.25.2->langgraph-sdk<0.3.0,
Requirement already satisfied: jsonpointer>=1.9 in /usr/local/lib/python3.12/dist-packages (from jsonpatch<2.0.0,>=1.33.0->langchain-core>=0
Requirement already satisfied: requests-toolbelt>=1.0.0 in /usr/local/lib/python3.12/dist-packages (from langsmith<1.0.0,>=0.3.45->langchain
Requirement already satisfied: zstandard>=0.23.0 in /usr/local/lib/python3.12/dist-packages (from langsmith<1.0.0,>=0.3.45->langchain-core>=
Requirement already satisfied: pyasn1<0.7.0,>=0.6.1 in /usr/local/lib/python3.12/dist-packages (from pyasn1-modules>=0.2.1->google-auth>=2.1
Requirement already satisfied: charset_normalizer<4,>=2 in /usr/local/lib/python3.12/dist-packages (from requests<3.0.0,>=2.18.0->google-api
Requirement already satisfied: urllib3<3,>=1.21.1 in /usr/local/lib/python3.12/dist-packages (from requests<3.0.0,>=2.18.0->google-api-core-
Requirement already satisfied: sniffio>=1.1 in /usr/local/lib/python3.12/dist-packages (from anyio->httpx>=0.25.2->langgraph-sdk<0.3.0,>=0.2
Using cached langchain_core-1.1.0-py3-none-any.whl (473 kB)
Installing collected packages: langchain-core
Successfully installed langchain-core-1.1.0
```

```
import google.generativeai as genai

GEMINI_API_KEY = "AIzaSyBGu_Ealjb1br4RRVKv3cmVZOkKU4ScgBo"

genai.configure(api_key=GEMINI_API_KEY)
```

```
def call_llm(prompt: str) -> str:
    model = genai.GenerativeModel("gemini-2.5-flash")

    response = model.generate_content(
        prompt,
        generation_config={"temperature": 0.4}
    )

    return response.text
```

```
def step_research(topic):
    prompt = f"Research this topic deeply and gather detailed, factual information:\n\nTopic: {topic}"
    return {"research": call_llm(prompt)}

def step_summarize(research_text):
    prompt = f"Summarize the following research in a clear and concise way:\n\n{research_text}"
    return {"summary": call_llm(prompt)}

def step_review(summary):
    prompt = f"Review this summary for clarity, correctness, and completeness:\n\n{summary}"
    return {"review": call_llm(prompt)}
```

```
from dataclasses import dataclass
from langgraph.graph import StateGraph, END

@dataclass
class ResearchState:
    topic: str = ""
    research: str = ""
    summary: str = ""
    review: str = ""

def build_research_graph():
    graph = StateGraph(ResearchState)

    graph.add_node("research", lambda s: step_research(s.topic))
    graph.add_node("summarize", lambda s: step_summarize(s.research))
    graph.add_node("review", lambda s: step_review(s.summary))

    graph.set_entry_point("research")

    graph.add_edge("research", "summarize")
    graph.add_edge("summarize", "review")
    graph.add_edge("review", END)

    return graph.compile()
```

```
def run_research_agent(topic: str):
    state = ResearchState(topic=topic)
    result = research_agent.invoke(state)

    # result is a dict — access keys directly
    return result.get("review", "No review field returned.")
```

```
output = run_research_agent("How does LangGraph work?")
print(output)
```

```
This is an **excellent summary** of LangGraph. It is clear, correct, and remarkably complete for its length.

Here's a breakdown:

**Clarity:**
*   **Outstanding.** The language is precise and easy to understand, even for someone new to LangGraph.
*   Key terms like "stateful," "multi-actor," "explicit control flow," "graph," "nodes," "edges," and "state" are clearly defined and contex
*   The analogy to a "Finite State Machine" is very helpful for conceptual understanding.
*   Examples for nodes (LLM, tool, Python function, another graph) are perfect.
*   The explanation of Checkpointing is particularly clear and highlights its importance.

**Correctness:**
*   **Highly accurate.** All technical descriptions align perfectly with LangGraph's design and functionality.
*   The relationship to LangChain ("extension of LangChain") is correctly stated.
*   The emphasis on cycles for iterative processes is spot on.
*   The description of `StateGraph` and `TypedDict` for state is correct.
*   The benefits listed at the end are accurate reflections of LangGraph's strengths.

**Completeness:**
*   **Very good.** For a summary, it covers all the essential aspects without being overly verbose.
*   It introduces the core problem LangGraph solves (sophisticated, stateful, multi-actor apps).
*   It clearly defines the fundamental components (nodes, edges, state).
*   It highlights crucial differentiators (cycles, FSM-like behavior, explicit control).
*   It lists key features and mechanisms (`StateGraph`, Checkpointing).
*   It briefly touches on the developer workflow and the overall benefits.
*   It doesn't miss any critical high-level concepts.

**Minor Suggestions (really nitpicks, as it's already great):**
*   None that significantly improve the summary. It strikes an excellent balance between detail and conciseness.

**Overall:**
This is a **top-tier summary**. It effectively communicates the core concepts, purpose, and key features of LangGraph in a concise, accurate
```

```python
# List of sample research questions
questions = [
    "Explain the main differences between LangGraph and LangChain.",
    "How do autonomous AI agents work in industry applications?",
    "Summarize recent advancements in large language model memory techniques.",
    "Provide an overview of vector databases used for AI retrieval-augmented generation.",
    "Explain the ethical considerations of using AI agents for business decision-making.",
    "What are the key techniques in prompt engineering for AI agents?",
    "How can AI agents improve software development workflows?",
    "Explain multi-agent reinforcement learning in simple terms.",
    "What are the challenges in deploying LLM-based AI agents in production?",
    "Describe how RAG (retrieval-augmented generation) works in AI systems."
]
```

```
]

# Loop over the questions and print results
for q in questions:
    print(f"\n=== Question: {q} ===\n")
    output = run_research_agent(q)
    print(output[:1000])  # print first 1000 characters
    print("\n" + "-"*80)
```