Assignment: Build a RAG-Powered Multi-Agent Q&A Assistant

Objective

Design and implement a simple “knowledge assistant” that:

1. **Retrieves** relevant information from a small document collection (RAG)
2. **Generates** natural-language answers via an LLM
3. **Orchestrates** the retrieval + generation steps with a basic agentic workflow

---

Scope & Deliverables

1. **Data Ingestion**
   - Select or prepare 3–5 short text documents (e.g. company FAQs, product specs).
   - Ingest and chunk them for vector indexing.
2. **Vector Store & Retrieval**
   - Build a vector index (FAISS, Pinecone, Chroma, etc.).
   - Implement a retrieval function that, given a user query, returns the top 3 relevant chunks.
3. **LLM Integration**
   - Use any LLM
   - Return the LLM’s answer.

4. **Agentic Workflow**
   - Using a simple agent framework (e.g. LangChain’s OpenAI agent), build logic that:
     - If the query contains keywords like “calculate” or “define,” route to a tool (e.g. a calculator or dictionary API
     - Otherwise, do the RAG → LLM pipeline
   - Log each “decision” step.

5. **Demo Interface**
   - Expose a minimal CLI or web UI (Flask/Streamlit) where you can type questions and see:
     - Which tool/agent branch was used
     - The retrieved context snippets
     - The final answer

Short README explaining your architecture, key design choices and how to run your code.