# Assignment – 3: Retrieval-Augmented Generation (RAG) using LangChain Task 1: Short Answer Questions

1. 1. What is the motivation behind Retrieval-Augmented Generation (RAG)?

Retrieval-Augmented Generation (RAG) combines retrieval mechanisms with generative models to provide accurate and fact-based answers. It helps overcome the hallucination problem of LLMs by fetching relevant context from external documents.

1. 2. Explain the difference between RAG and standard LLM-based QA.

Standard LLM QA relies solely on internal model knowledge. In contrast, RAG retrieves relevant data from external sources to provide up-to-date, contextual responses.

1. 3. What is the role of a vector store in a RAG pipeline?

A vector store stores text embeddings in a format that allows fast similarity search. During RAG, it retrieves chunks of documents most similar to the user’s query.

1. 4. Compare “stuff”, “map\_reduce”, and “refine” document chain types in LangChain.

- 'stuff': loads all documents and puts them into one prompt.  
- 'map\_reduce': maps over docs separately, then reduces.  
- 'refine': uses an initial doc to generate output and refines it with others.

1. 5. What are the main components of a basic LangChain RAG pipeline?

- Document Loader  
- Text Splitter  
- Embeddings  
- Vector Store (e.g., FAISS/Chroma)  
- Retriever  
- LLM  
- RetrievalQA Chain