Section 1:

Advanced Customer Support Bot with Multi-Turn Conversation Handling and Enhanced Function Execution

Objective

Build an advanced customer support chatbot leveraging OpenAI models, LangChain, and vector databases. The bot should handle multi-turn conversations, infer user intent dynamically, and execute multiple function calls based on contextual understanding.

Feature 1: Build the Customer Support Bot

1. Data Ingestion and Retrieval

- Use the provided knowledge base document.
- Implement a hybrid retrieval system combining dense embeddings (FAISS, ChromaDB, Pinecone) with keyword-based retrieval.
- Optimize retrieval by using metadata filtering or re-ranking techniques.

2. Multi-Turn Question Answering System

- Use OpenAI models with LangChain to create an intelligent chatbot.
- Implement Retrieval-Augmented Generation (RAG) with history-aware responses.
- Utilize LangChain memory components to retain context across multiple interactions.

Task 2: Advanced Function Calling for Support Requests

1. Define Multiple Custom Functions

- Implement at least three functions:
    1. create_support_ticket(details): Creates a support ticket.
    2. fetch_order_status(order_id): Retrieves order-related information.

2. Intelligent Function Execution

- Use LangChain tools or OpenAI's function calling to trigger multiple functions dynamically based on user requests.
- Ensure the chatbot can handle conflicting user intents and ask clarifying questions before function execution.

3. Example Scenarios

- If the user says, “I need help with my order, and I also want to schedule a call,” the bot should identify and execute both fetch_order_status functions appropriately.
- Simulate a response confirming the ticket creation, order status retrieval with relevant details.

Expectations:

- Well-structured, scalable, and modular code.
- Use of LangChain tools, agents, and structured output functions.
- Ability to manage multi-turn conversations with memory.
- Efficient and optimized retrieval system.

Section 2:

Python Coding Questions

1.Problem: Write a Python function to find the median of two sorted arrays of different lengths without merging them.

Input Example: `[1, 3, 8]`, `[7, 9, 10, 11]`

Output Example: `8.0`

2. Problem: Implement a Python function to find the smallest window in a string that contains all the characters of another string.

Input Example: `"ADOBECODEBANC"`, `"ABC"`

Output Example: `"BANC"`

3. Problem: Write a Python function to implement the "Word Break" problem. Given a string and a dictionary of words, return `True` if the string can be segmented into space-separated words from the dictionary.

Input Example: `"leetcode"`, `["leet", "code"]`

Output Example: `True`

4. Problem: Implement a Python function to find the longest substring without repeating characters.

Input Example: `"abrkaabcdefghijjxxx"`

Output Example: `10` (substring: `"abcdefghij"`)

5. Problem: Write a Python function to solve the "N-Queens" problem. The function should return all possible solutions for placing N queens on an N×N chessboard so that no two queens threaten each other.

Input Example: `4`

Output Example: `[[1, 3, 0, 2], [2, 0, 3, 1]]`