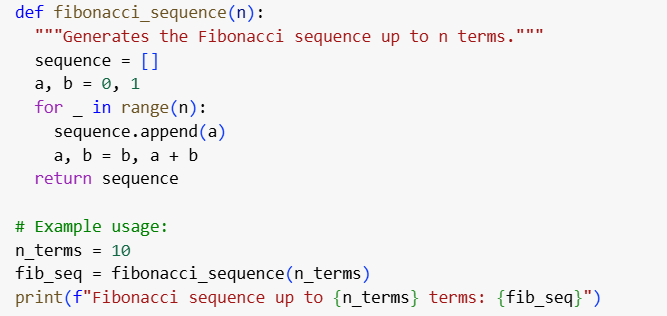
|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **SCHOOL OF COMPUTER SCIENCE AND ARTIFICIAL INTELLIGENCE** | | | | | **DEPARTMENT OF COMPUTER SCIENCE ENGINEERING** | | | | |
| **ProgramName:**B. Tech | | | | **Assignment Type: Lab** | | | **AcademicYear:**2025-2026 | | |
| **CourseCoordinatorName** | | | | Venkataramana Veeramsetty | | | | | |
| **Instructor(s)Name** | | | | |  | | --- | | Dr. V. Venkataramana (Co-ordinator) | | Dr. T. Sampath Kumar | | Dr. Pramoda Patro | | Dr. Brij Kishor Tiwari | | Dr.J.Ravichander | | Dr. Mohammand Ali Shaik | | Dr. Anirodh Kumar | | Mr. S.Naresh Kumar | | Dr. RAJESH VELPULA | | Mr. Kundhan Kumar | | Ms. Ch.Rajitha | | Mr. M Prakash | | Mr. B.Raju | | Intern 1 (Dharma teja) | | Intern 2 (Sai Prasad) | | Intern 3 (Sowmya) | | NS\_2 ( Mounika) | | | | | | |
| **CourseCode** | | | 24CS002PC215 | **CourseTitle** | | AI Assisted Coding | | | |
| **Year/Sem** | | | II/I | **Regulation** | | R24 | | | |
| **Date and Day**  **of Assignment** | | | Week2 - Tuesday | **Time(s)** | |  | | | |
| **Duration** | | | 2 Hours | **Applicableto**  **Batches** | |  | | | |
| **AssignmentNumber:4.2**(Present assignment number)/**24**(Total number of assignments) | | | | | | | | | |
|  | | | | | | | | | |
|  | **Q.No.** | **Question** | | | | | | ***ExpectedTime***  ***to complete*** |  |
|  | 1 | Lab 4: Advanced Prompt Engineering – Zero-shot, One-shot, and Few-shot Techniques  **Lab Objectives:**   * To explore and apply different levels of prompt examples in AI-assisted code generation. * To understand how zero-shot, one-shot, and few-shot prompting affect AI output quality. * To evaluate the impact of context richness and example quantity on AI performance. * To build awareness of prompt strategy effectiveness for different problem types.   **Lab Outcomes (LOs):**  After completing this lab, students will be able to:   * Use zero-shot prompting to instruct AI with minimal context. * Use one-shot prompting with a single example to guide AI code generation. * Apply few-shot prompting using multiple examples to improve AI responses. * Compare AI outputs across the three prompting strategies.   **Task Description#1**   * **Zero-shot:** Prompt AI with only the instruction — Write a Python function to generate the Fibonacci sequence up to n terms   **Expected Output#1**   * A working function without using any sample inputs/outputs.   **Task Description#2**   * One-shot: Provide one example: Input: 100, Output: 37.78 to help AI generate a function that converts Fahrenheit to Celsius.   **Expected Output#2**   * A correct conversion function guided by the single example.   **Task Description#3**   * **Few-shot:** Give 2–3 examples to create a function that extracts the domain name from an email address.   **Expected Output#3**   * Accurate function that returns only the domain portion of an email (e.g., @gmail.com).   **Task Description#4**   * Compare zero-shot vs few-shot prompting for generating a function that checks whether a word is a palindrome, ignoring punctuation and case.   **Expected Output#4**   * Output comparison + student explanation on how examples helped the model.   **Task Description#5**   * Use few-shot prompting with 3 sample inputs to generate a function that determines the maximum of three numbers without using the built-in max() function.   **Expected Output#5**   * A function that handles all cases with correct logic based on example patterns.   **Note: Report should be submitted a word document for all tasks in a single document with prompts, comments & code explanation, and output and if required, screenshots**  **Evaluation Criteria:**   | **Criteria** | **Max Marks** | | --- | --- | | Zero Shot (Task #1) | 0.5 | | One Shot (Task#2) | 0.5 | | Few Shot (Task#3 & Task #5) | 1.0 | | Comparison (Task#4) | 0.5 | | **Total** | **2.5 Marks** | | | | | | | Week2 - Wednesday |  |

**ROLL NO:2403A51301 DATE:18-08-2025**

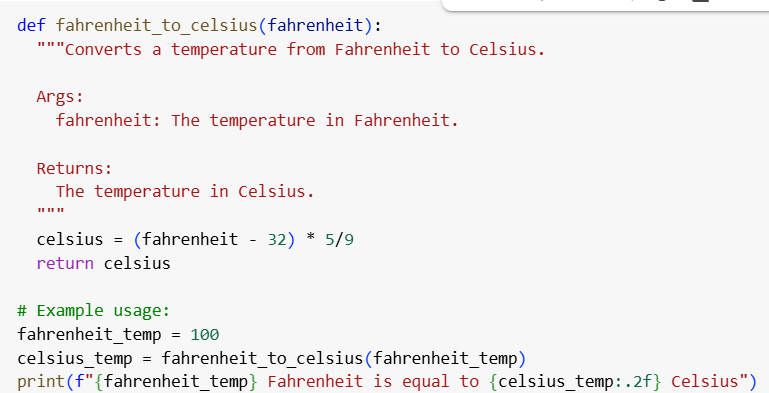
**NAME:JOGIPARTHY SAI GANESH**

**#TASK1  
ZERO-SHOT-PROMPT:** Write a Python function to generate the Fibonacci sequence up to n terms**.  
CODE:** **OUTPUT:** **OBSERVATION:** The single-shot prompt successfully generated a correct Python function that produces the Fibonacci sequence up to *n* terms. The output matches the expected sequence ,we didn’t mentioned for 10 terms but it takes automatically it may helpful.

**#TASK2  
ONE-SHOT-PROMPT:** Write a Python function that converts Fahrenheit to Celsius.

Example:

Input: 100

Output: 37.78  
**CODE:**

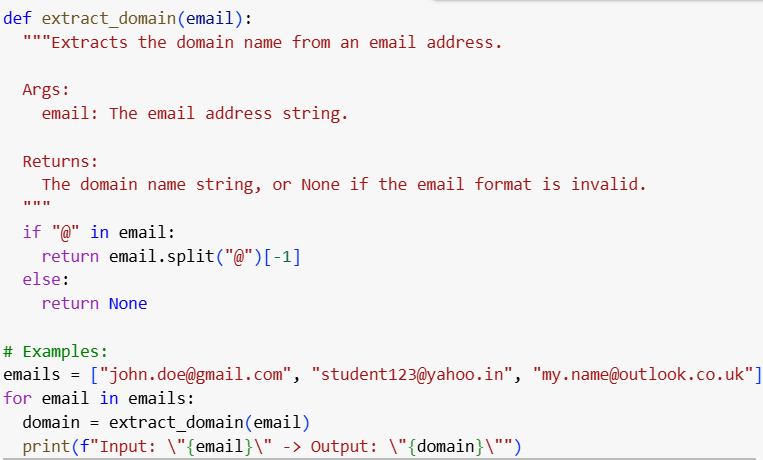
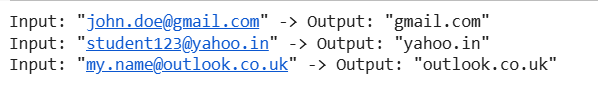
**OUTPUT:** **OBSERVATION:** The one-shot prompt with a given example correctly guided the AI to generate a Fahrenheit-to-Celsius conversion function. The output 100 → 37.78 matches the expected result, showing how a single example improves accuracy in function design.

**#TASK3  
FEW-SHOT-PROMPT**: Write a Python function that extracts the domain name from an email address.

Examples:

Input: "john.doe@gmail.com" → Output: "gmail.com"

Input: "student123@yahoo.in" → Output: "yahoo.in"

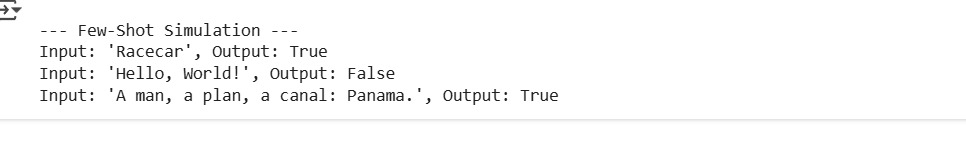
Input: "my.name@outlook.co.uk" → Output: "outlook.co.uk"  
**CODE:**   
**OUTPUT:**   
  
**OBSERVATION:** The few-shot prompt with multiple examples helped the AI generate a correct function that extracts only the domain part from email addresses. The outputs (gmail.com, yahoo.in, outlook.co.uk) match the expected results, showing improved reliability through guided examples.

**#TASK4  
ZERO-SHOT PROMPT**: "Write a Python function to check whether a given word is a palindrome, ignoring punctuation and case."  
**CODE:   
OUTPUT:** **FEW-SHOT-PROMPT**: "Write a Python function to check whether a given word is a palindrome, ignoring punctuation and case."  
 Examples:

Input: "Madam" → Output: True

Input: "RaceCar" → Output: True

Input: "Hello" → Output: False

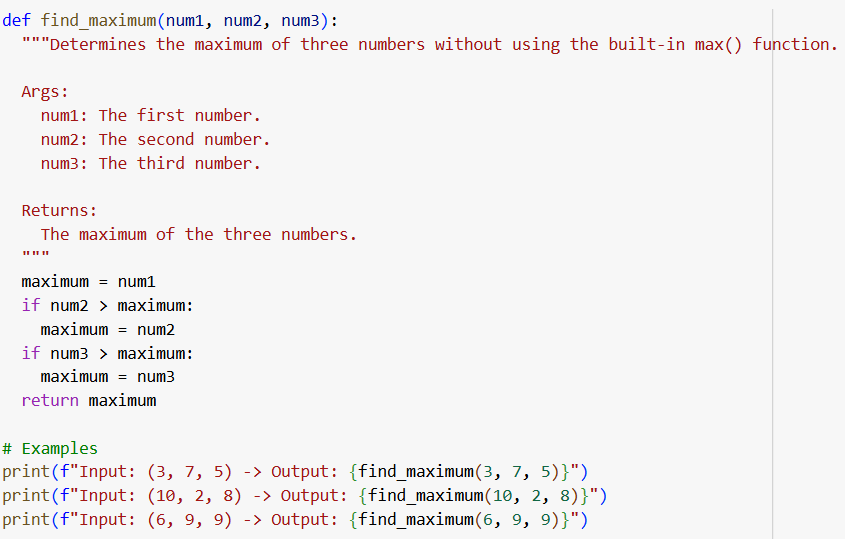
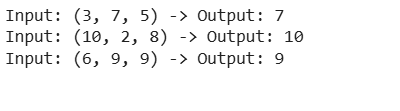
**CODE:   
  
OUT PUT:** **OBSERVATION(By comparing both zero-shot and few-shot prompting**): In zero-shot, the function worked correctly but without clarity on varied inputs.  
In few-shot, examples guided the model to handle different cases (mixed case, punctuation, spaces) more reliably.  
This shows that examples improve accuracy and adaptability of the function.

**#TASK5  
FEW-SHOT-PROMPT:** Write a Python function to determine the maximum of three numbers without using the built-in max() function.

Examples:

Input: (3, 7, 5) → Output: 7

Input: (10, 2, 8) → Output: 10

Input: (6, 9, 9) → Output: 9  
**CODE:** **OUTPUT:** **OBSERVATION:** The function compares numbers pairwise using conditional statements to determine the largest value.  
It correctly returns the maximum of three numbers without relying on the built-in max() function.

Top of Form

Bottom of Form