# **AI ASSIGNMENT 4.3 NAME:2503A52L04**

# **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)**

- 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 1: Zero-shot Prompting**

### **Prompt**

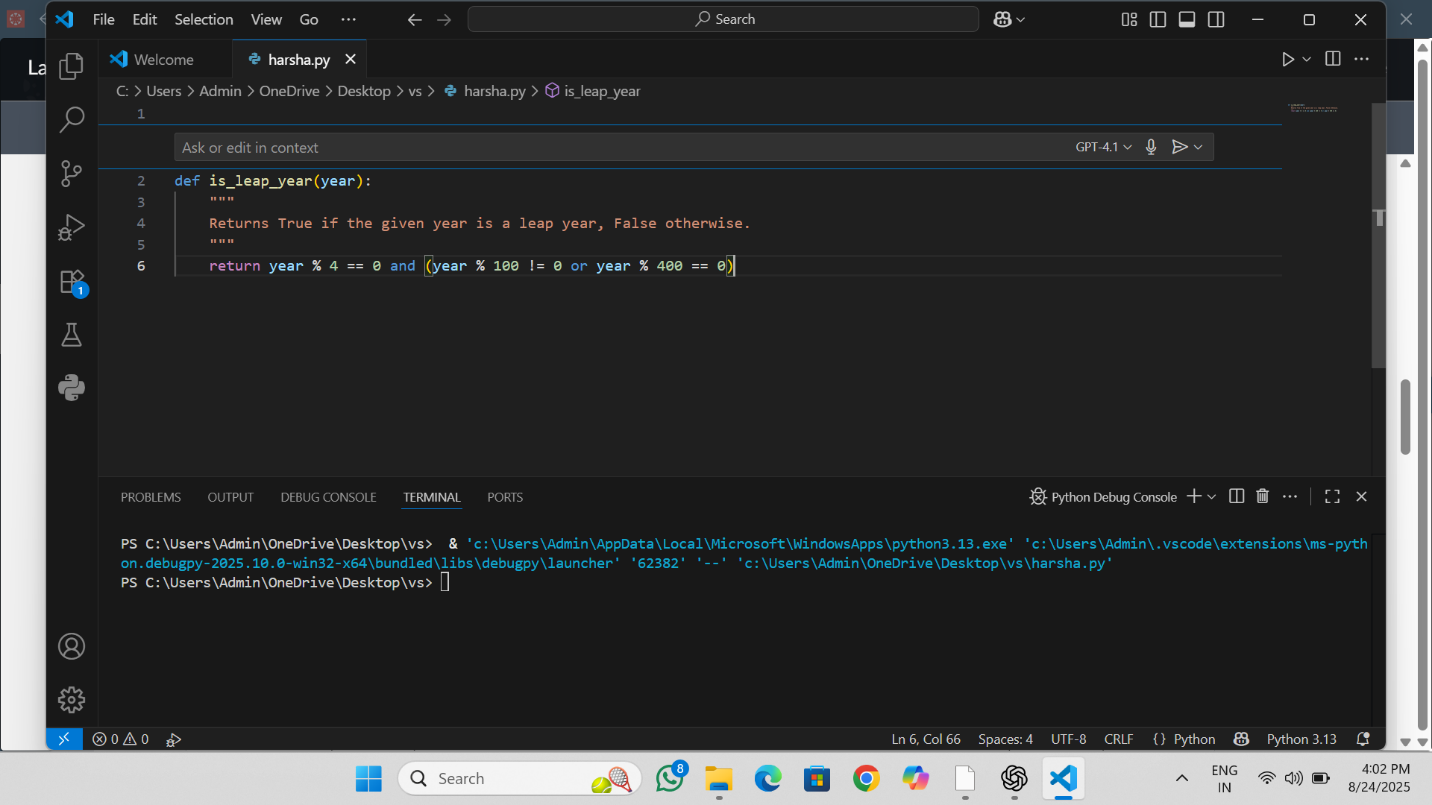
Write a Python function that checks whether a given year is a leap year.

### **AI-Generated Code**

def is\_leap\_year(year):  
 return (year % 4 == 0 and year % 100 != 0) or (year % 400 == 0)  
  
# Example usage  
print(is\_leap\_year(2024)) # True  
print(is\_leap\_year(2023)) # False

### **Explanation**

Zero-shot prompting provides only instruction without examples. The function implements the correct leap-year logic: years divisible by 4 are leap years, except century years unless divisible by 400.



## **Task 2: One-shot Prompting**

### **Prompt**

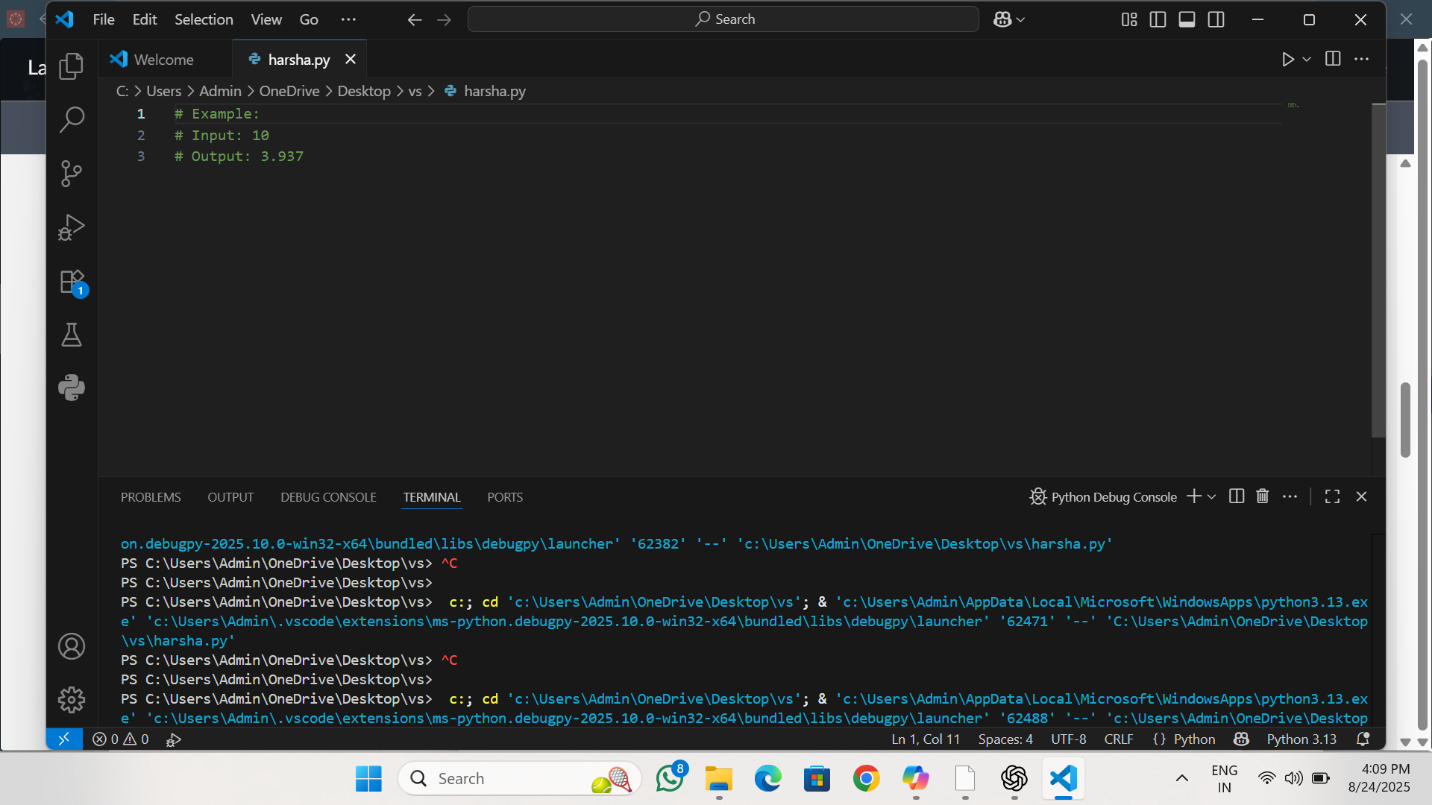
Example: Input = 10 cm → Output = 3.94 inches. Now write a function that converts centimeters to inches.

### **AI-Generated Code**

def cm\_to\_inches(cm):  
 # 1 inch = 2.54 centimeters  
 return cm / 2.54  
  
# Example usage  
print(cm\_to\_inches(10)) # 3.937007874015748  
print(cm\_to\_inches(25)) # 9.84251968503937

### **Explanation**

With a single example, the AI aligns to the expected conversion and output style. The function divides by 2.54 to convert centimeters to inches.



## **Task 3: Few-shot Prompting**

### **Prompt**

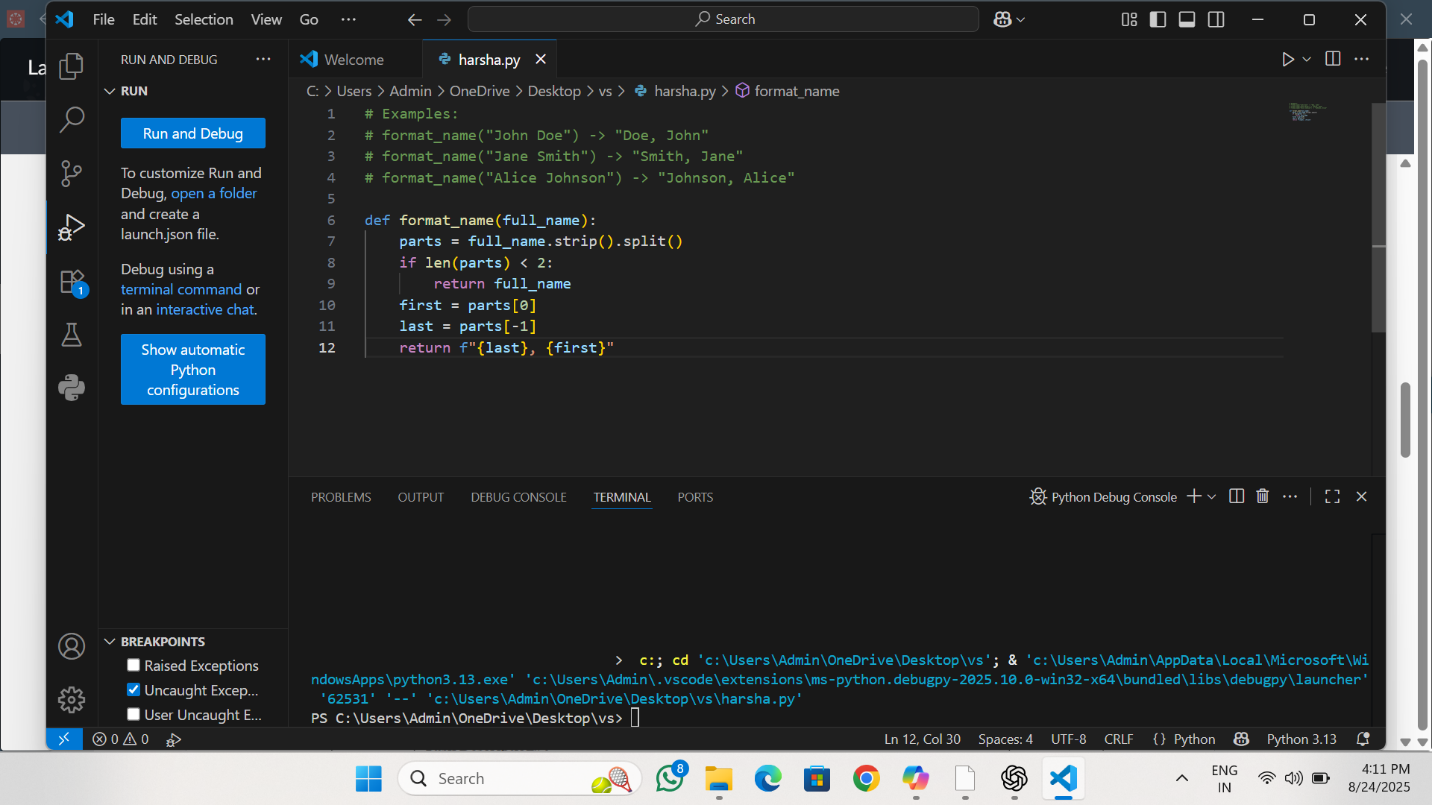
Examples:  
Input: "John Doe" → Output: "Doe, John"  
Input: "Alice Johnson" → Output: "Johnson, Alice"  
Input: "Ravi Kumar" → Output: "Kumar, Ravi"  
  
Now write a function that formats full names as "Last, First".

### **AI-Generated Code**

def format\_name(full\_name):  
 parts = [p for p in full\_name.split() if p]  
 if len(parts) < 2:  
 return full\_name # not enough parts to swap  
 first = parts[0]  
 last = parts[-1]  
 return f"{last}, {first}"  
  
# Example usage  
print(format\_name("John Doe")) # Doe, John  
print(format\_name("Alice Johnson")) # Johnson, Alice  
print(format\_name("Ravi Kumar")) # Kumar, Ravi

### **Explanation**

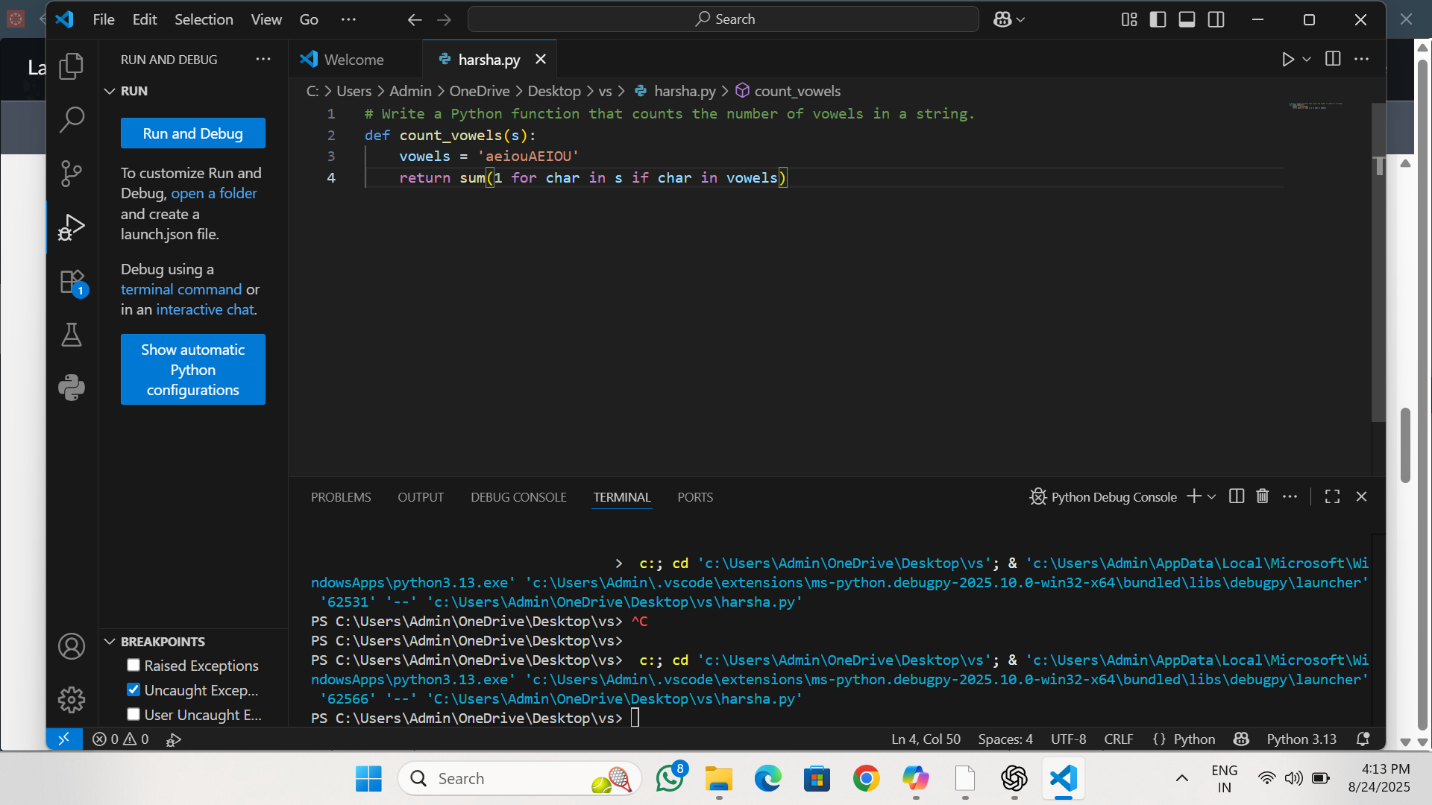
Multiple examples help the AI generalize the desired pattern. This version safely handles unexpected inputs (like single-word names).



## **Task 4: Zero-shot vs Few-shot – Count Vowels**

### **Zero-shot Prompt**

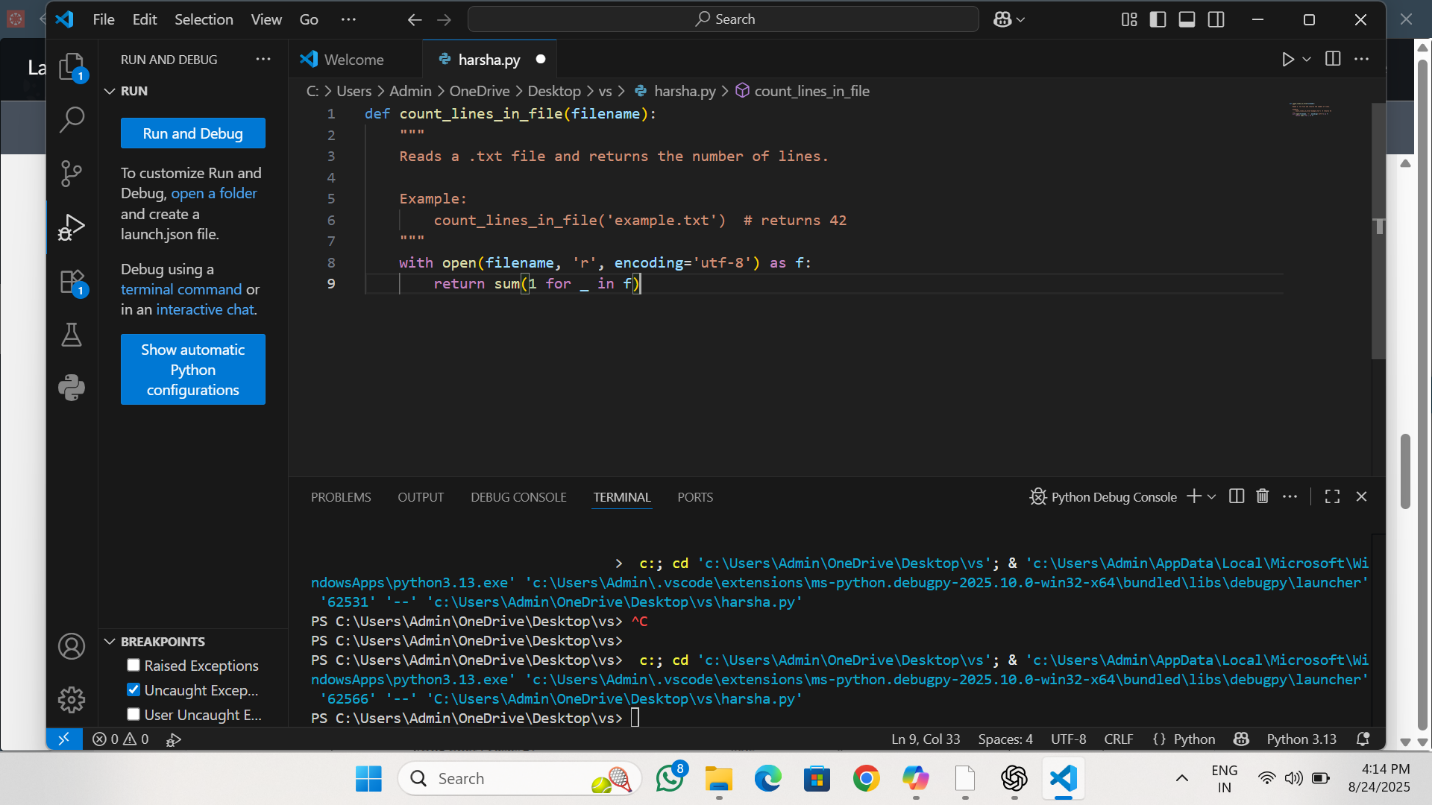
Write a function that counts the number of vowels in a string.



## Task 5: Few-shot – File Line Counter

### Prompt

Examples:  
File content:  
Line1  
Line2  
Line3  
Output: 3  
  
Now write a function that reads a .txt file and returns the number of lines.



### Explanation

Few-shot examples clarify the task. The function reads the file line by line (memory-friendly) and counts lines. Basic error handling is included for missing files.