Lab Assignment – 3.1

Hall Ticket No.: 2303A510E6

Name – Anushka Boora

Batch – 29

Question 1: Zero-Shot Prompting (Palindrome Number Program)

Write a zero-shot prompt (without providing any examples) to generate

a Python function that checks whether a given number is a palindrome.

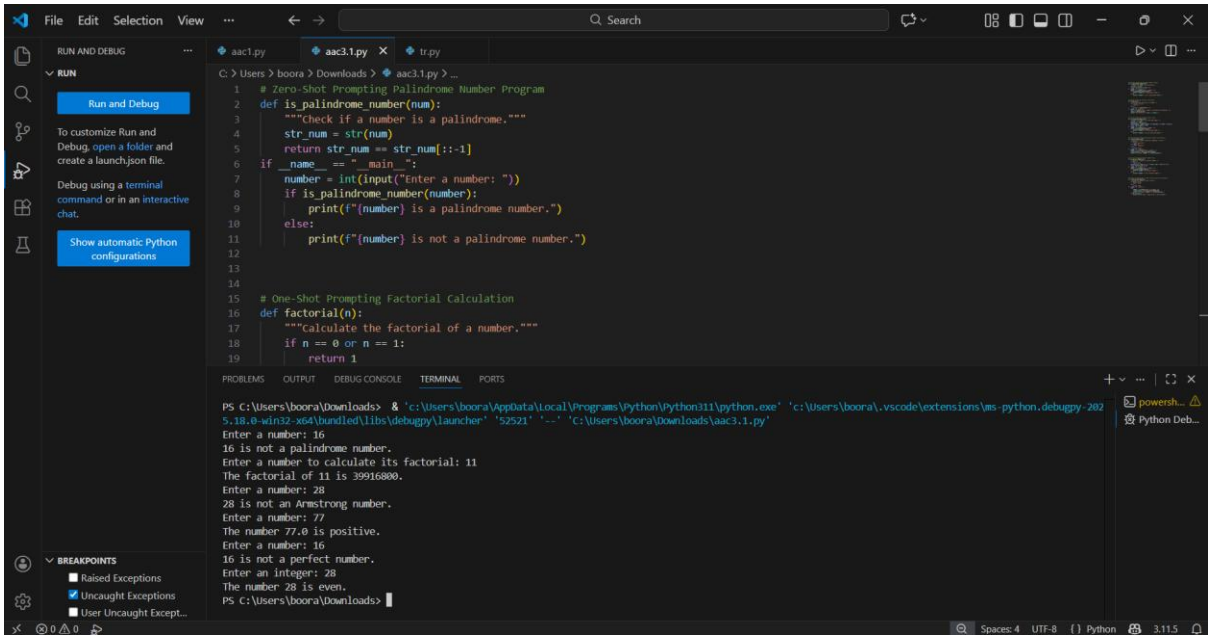
Task:

• Record the AI-generated code.

• Test the code with multiple inputs.

• Identify any logical errors or missing edge-case handling.

CODE & OUTPUT



Question 2: One-Shot Prompting (Factorial Calculation)

Write a one-shot prompt by providing one input-output example and

ask the AI to generate a Python function to compute the factorial of a

given number.

Example:

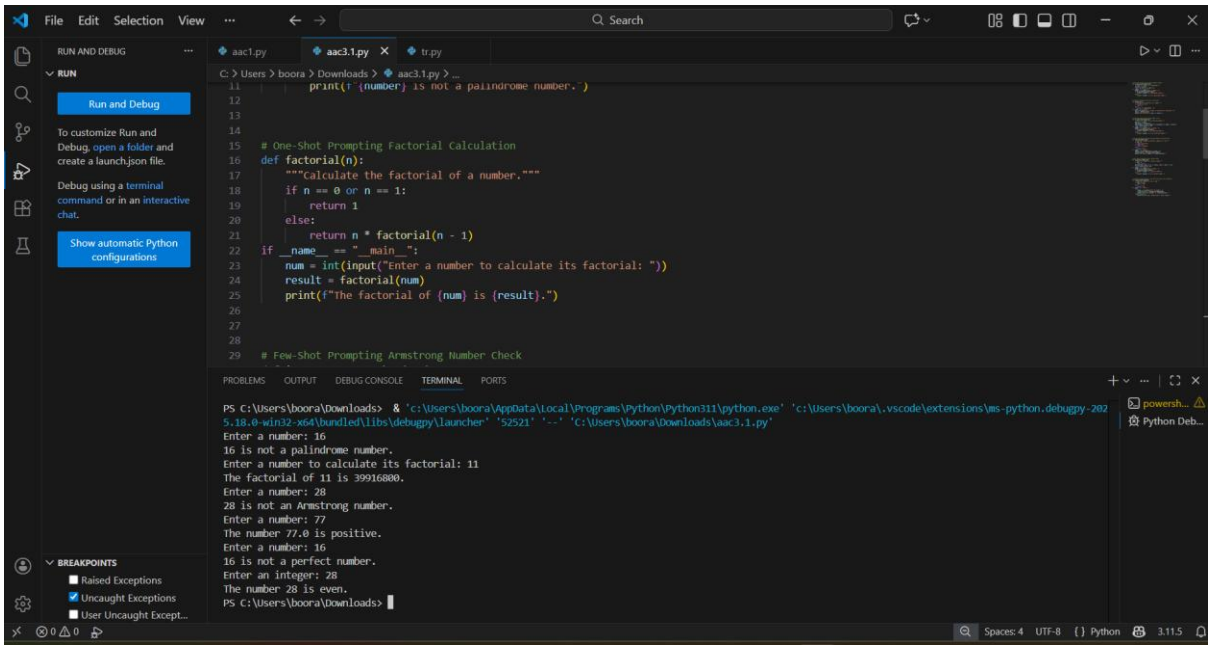
Input: 5 → Output: 120

Task:

• Compare the generated code with a zero-shot solution.

• Examine improvements in clarity and correctness.

CODE & OUTPUT



Question 3: Few-Shot Prompting (Armstrong Number Check)

Write a few-shot prompt by providing multiple input-output examples

to guide the AI in generating a Python function to check whether a

given number is an Armstrong number.

Examples:

• Input: 153 → Output: Armstrong Number

• Input: 370 → Output: Armstrong Number

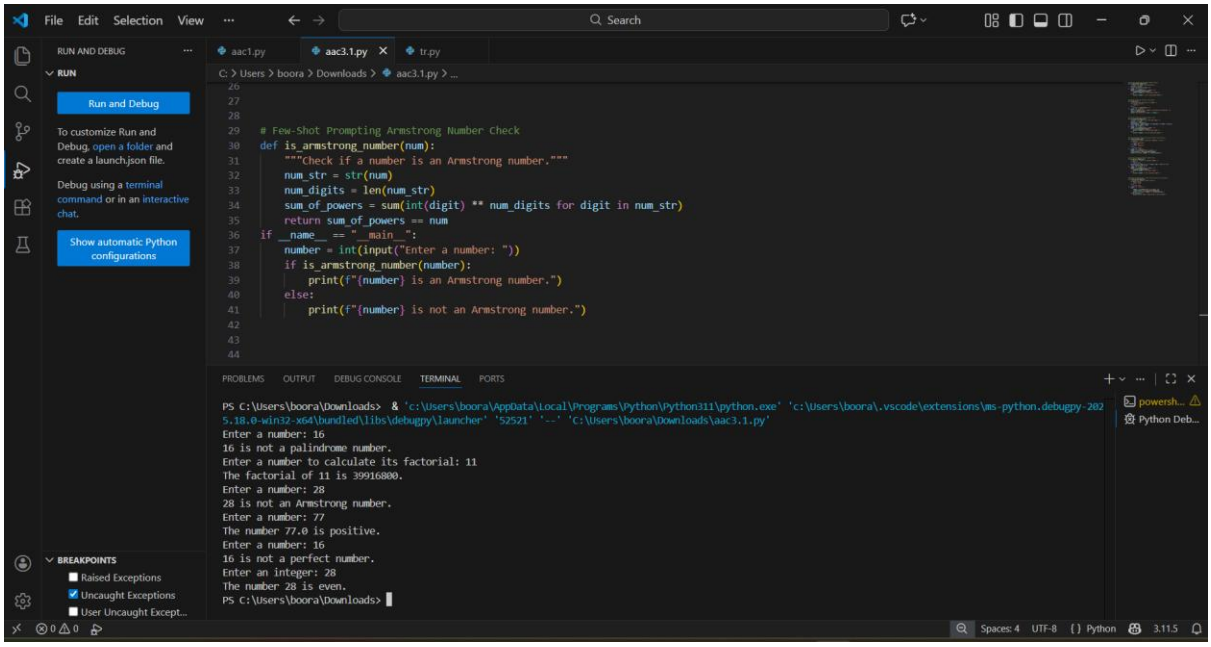
• Input: 123 → Output: Not an Armstrong Number

Task:

• Analyze how multiple examples influence code structure and accuracy.

• Test the function with boundary values and invalid inputs.

CODE & OUTPUT



Question 4: Context-Managed Prompting (Optimized Number Classification)

Design a context-managed prompt with clear instructions and constraints to generate an optimized Python program that classifies a number as prime, composite, or neither.

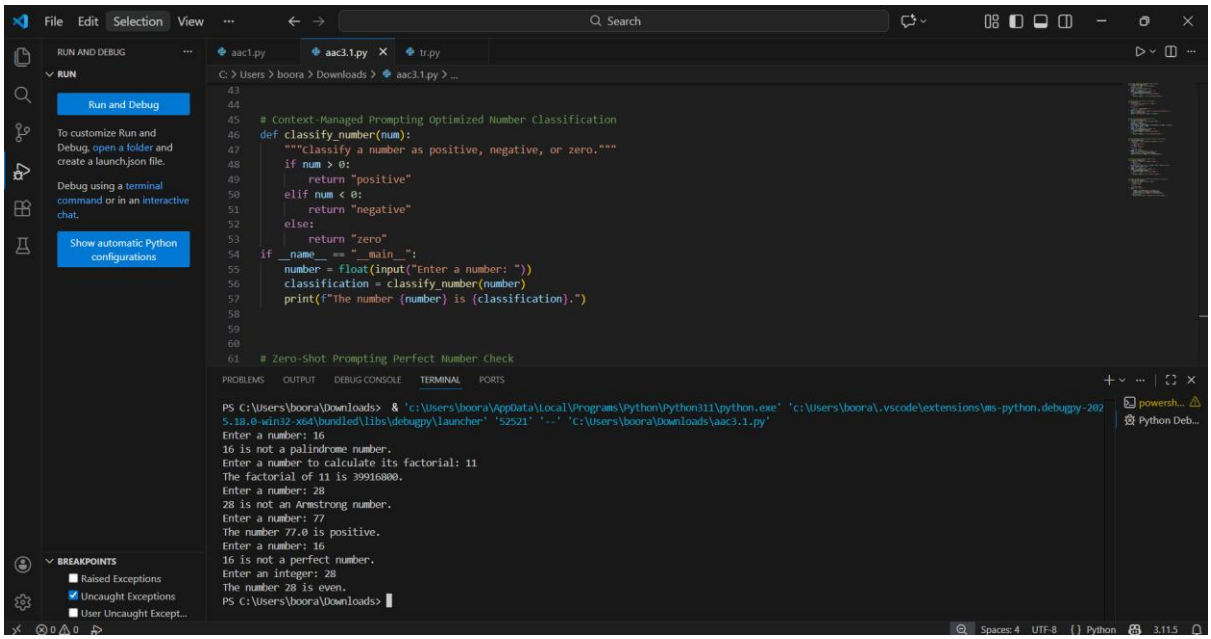
Task:

• Ensure proper input validation.

• Optimize the logic for efficiency.

• Compare the output with earlier prompting strategies.

CODE & OUTPUT



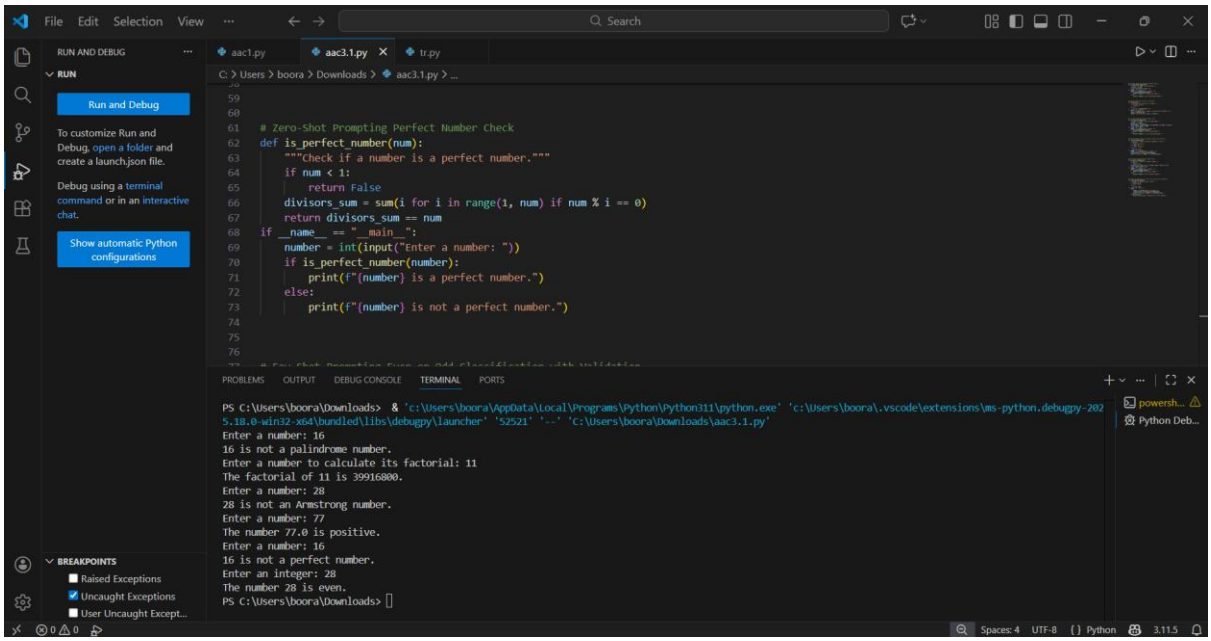
Question 5: Zero-Shot Prompting (Perfect Number Check)

Write a zero-shot prompt (without providing any examples) to generate a Python function that checks whether a given number is a perfect number.

Task:

- Record the AI-generated code.
- Test the program with multiple inputs.
- Identify any missing conditions or inefficiencies in the logic.

CODE & OUTPUT



Question 6: Few-Shot Prompting (Even or Odd Classification with Validation)

Write a few-shot prompt by providing multiple input-output examples to guide the AI in generating a Python program that determines whether a given number is even or odd, including proper input validation.

Examples:

• Input: 8 → Output: Even

• Input: 15 → Output: Odd

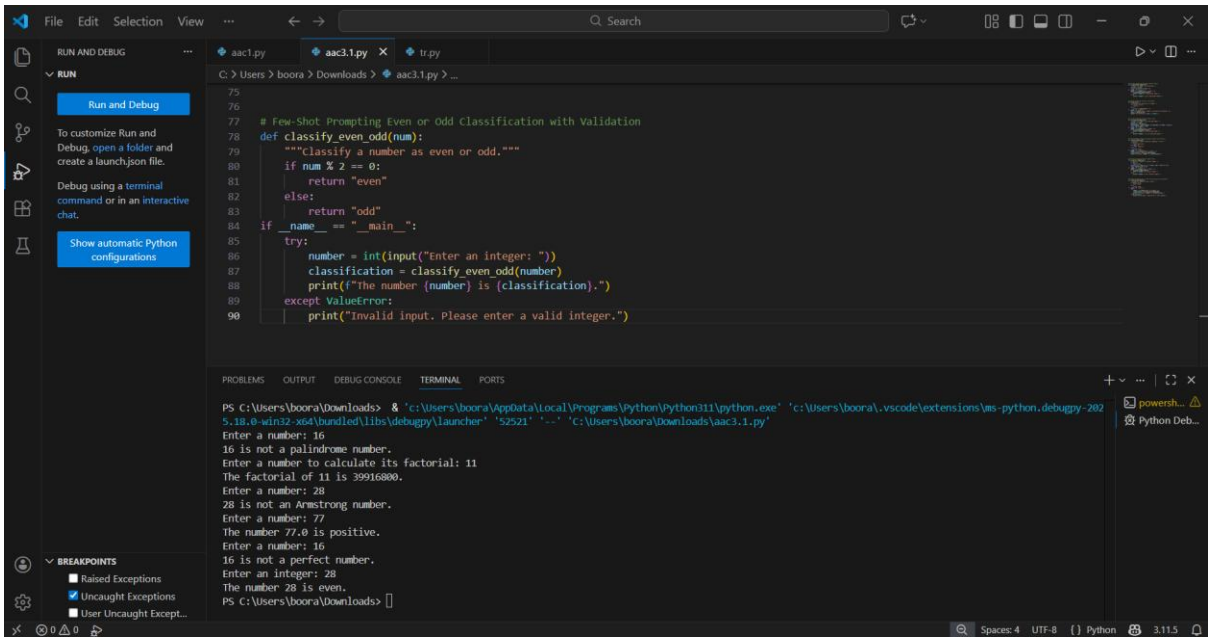
• Input: 0 → Output: Even

Task:

• Analyze how examples improve input handling and output clarity.

• Test the program with negative numbers and non-integer inputs.

CODE & OUTPUT


File Edit Selection View
Search
RUN AND DEBUG
RUN
Run and Debug
To customize Run and Debug, open a folder and create a launch.json file.
Debug using a terminal command or in an interactive chat.
Show automatic Python configurations
BREAKPOINTS
Raised Exceptions
Uncaught Exceptions
User Uncaught Except...
aac1.py
aac3.1.py
tr.py
C: > Users > boora > Downloads > aac3.1.py > ...
# Few-Shot Prompting Even or Odd Classification with Validation
def classify_even_odd(num):
"""classify a number as even or odd."""
if num % 2 == 0:
return "even"
else:
return "odd"
if __name__ == "__main__":
try:
number = int(input("Enter an integer: "))
classification = classify_even_odd(number)
print(f"The number {number} is {classification}.")
except ValueError:
print("Invalid input. Please enter a valid integer.")
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS C:\Users\boora\Downloads> & 'c:\Users\boora\AppData\Local\Programs\Python\Python311\python.exe' 'c:\Users\boora\.vscode\extensions\ms-python.debugpy-2025.18.0-win32-x64\bundled\libs\debugpy\launcher' '52521' '--' 'C:\Users\boora\Downloads\aac3.1.py'
Enter a number: 16
16 is not a palindrome number.
Enter a number to calculate its factorial: 11
The factorial of 11 is 39916800.
Enter a number: 28
28 is not an Armstrong number.
Enter a number: 77
The number 77.0 is positive.
Enter a number: 16
16 is not a perfect number.
Enter an integer: 28
The number 28 is even.
PS C:\Users\boora\Downloads>
powersh...
Python Deb...
Spaces: 4 UTF-8 { } Python 3.11.5