

ASSIGNMENT 3.1

Name:T.Rakshitha

Hall Ticket No:2303a51172

Batch:18

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.

Prompt :

```
#read input from user number to check if it is palindrome
```

Code :

The screenshot shows a code editor interface with a dark theme. On the left is the Explorer sidebar, which lists several files under 'AI ASSISTANT'. In the center is the main code editor area where a Python script is written. The script reads a number from the user and checks if it is a palindrome. At the bottom, the terminal window shows the script being run and a sample input-output session where the user enters '121' and the script correctly identifies it as a palindrome.

```
#take input from user and check if it is leapyear.py
#read input from user number to check if it is palindrome.py

#read input from user number to check if it is palindrome
num = int(input("Enter a number to check if it is a palindrome: "))
original_num = num
reversed_num = 0
while num > 0:
    digit = num % 10
    reversed_num = reversed_num * 10 + digit
    num //= 10
if original_num == reversed_num:
    print(original_num, "is a palindrome")
else:
    print(original_num, "is not a palindrome")
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS C:\Users\raksh\OneDrive\Desktop\AI ASSISTANT> & C:/Users/raksh/AppData/Local/Programs/Python/Python313/powershell
ndrome.py"
Enter a number to check if it is a palindrome: 121
121 is a palindrome
PS C:\Users\raksh\OneDrive\Desktop\AI ASSISTANT>

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.

Prompt :

#take input from user and print factorial of that number

Code :

The screenshot shows a code editor interface with a dark theme. On the left is the Explorer sidebar, which lists several files under 'AI ASSISTANT', including 'read input from user to check if it is palindrome.py', 'take input from user and print factorial of that number.py', 'armstrong.py', and 'read a number and ...'. The main workspace displays the following Python code:

```
#take input from user and print factorial of that number
num = int(input("Enter a number to calculate its factorial: "))
factorial = 1
for i in range(1, num + 1):
    factorial *= i
print("The factorial of", num, "is", factorial)
```

The terminal tab at the bottom shows the command being run and the output: 'PS C:\Users\raksh\OneDrive\Desktop\AI ASSISTANT> & c:/Users/raksh/AppData/Local/Programs/Python/Python313/python.exe "c:/Users/raksh/OneDrive/Desktop/ai_assistant/take input from user and print factorial of that number.py"' followed by 'Enter a number to calculate its factorial: 23' and 'The factorial of 23 is 2585201673884976640000'. The status bar at the bottom indicates the file is '3.13.2' and the date is '22-01-2026'.

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.**

(Optional Extension)

Prompt :

#take input from user and check if it is armstrong number

Code – output

The screenshot shows the Visual Studio Code interface. The Explorer sidebar on the left lists several Python files, including one named '#take input from user and check if it is armstrong number.py'. The main editor area displays the following Python code:

```
1 #take input from user and check if it is armstrong number
2 num = int(input("Enter a number to check if it is an Armstrong number: "))
3 order = len(str(num))
4 sum_of_powers = 0
5 temp = num
6 while temp > 0:
7     digit = temp % 10
8     sum_of_powers += digit ** order
9     temp /= 10
10 if sum_of_powers == num:
11     print(num, "is an Armstrong number")
12 else:
13     print(num, "is not an Armstrong number")
```

Below the editor, the terminal tab is active, showing the output of running the script:

```
Sum of odd numbers in the tuple: 3
PS C:\Users\raksh\OneDrive\Desktop\AI ASSISTANT> & C:/Users/raksh/AppData/Local/Programs/Python/Python313/python.exe "c:/U
Enter a number to check if it is an Armstrong number: 153
○ 153 is an Armstrong number
PS C:\Users\raksh\OneDrive\Desktop\AI ASSISTANT>
```

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

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.

Prompt :

read a number and check if it is prime composite or neither

Code – output

The screenshot shows the Visual Studio Code interface with the following details:

- File Explorer:** Shows several files including "it is palindrome.py", "#take input from user and print factorial of that number.py", "#read a number and check if it is prime composite or neither.py", "armstrong.py", and "#read a number and check if it is prime composite or neither.py".
- Code Editor:** Displays a Python script for checking if a number is prime, composite, or neither. The code uses a loop to iterate from 2 to the square root of the number to check for factors.
- Terminal:** Shows command-line history for running the script. It includes commands like "sktop\AI ASSISTANT\armstrong.py" and "sktop\AI ASSISTANT\#read a number and check if it is prime composite or neither.py". The output shows the script's logic and a test run for the number 7.
- Status Bar:** Provides information about the file (Line 1, Col 1), encoding (UTF-8), and system status (Python 3.13.2, ENG IN, 22-01-2026).

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.

Prompt :

#take input from user and chcek if it is perfect number

Code - output

The screenshot shows the Visual Studio Code interface with the AI Assistant extension active. The Explorer sidebar on the left lists several Python files, including 'check even or odd.py' and 'check if it is a perfect number.py'. The main editor area displays code for checking if a number is perfect. The bottom right corner shows the system tray with the date and time.

```

#take input from user and check if it is a perfect number > ...
1 num=int(input("Enter a number to check if it is a perfect number: "))
2 sum_of_divisors=0
3 for i in range(1,num):
4     if num%i==0:
5         sum_of_divisors+=i
6 if sum_of_divisors==num:
7     print(f"{num} is a perfect number")
8 else:
9     print(f"{num} is not a perfect number")
10

```

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.

Prompt :

take input from user and check even or odd

Code – output

The screenshot shows a code editor interface with several windows open. In the center, there is a code editor window titled "take input from user and check even or odd.py" containing the following Python code:

```
#take input from user and check even or odd.py
#read a number and check if it is prime co...
#read a number and check if it is prime co...
#read input from user to check if it ...
#read tuple from user and print sum of eve...
#take input from user and check even or od...
#take input from user and check if it is a pe...
#take input from user and check if it is arms...
#take input from user and check if it is leap ...
#take input from user and print factorial of ...
armstrong.py
```

Below the code editor is a terminal window titled "TERMINAL" showing the following command-line session:

```
PS C:\Users\raksh\OneDrive\Desktop\AI ASSISTANT> PS C:\Users\raksh\OneDrive\Desktop\AI ASSISTANT> & C:/Users/raksh/AppData/Local/Programs/Python/Python313/python.exe "c:/Users/raksh/OneDrive/De...
sktop/AI ASSISTANT/take input from user and check even or odd.py"
Enter a number to check if it is even or odd: 15
15 odd number
● PS C:\Users\raksh\OneDrive\Desktop\AI ASSISTANT> PS C:\Users\raksh\OneDrive\Desktop\AI ASSISTANT> & C:/Users/raksh/AppData/Local/Programs/Python/Python313/python.exe "c:/Users/raksh/OneDrive/De...
sktop/AI ASSISTANT/take input from user and check even or odd.py"
Enter a number to check if it is even or odd: 28
28 even number
○ PS C:\Users\raksh\OneDrive\Desktop\AI ASSISTANT>
```

The bottom status bar displays various system information including the date and time (22-01-2026), weather (28°C, Sunny), and system settings.