

Assignment 3.1

2303A52079

B.Shivamani

Batch 39

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 plaindrome

Code :

```
lab3.py > ...
1  #read input from user number to check if it is plaindrome
2  num = int(input("Enter a number to check if it is a palindrome: "))
3  original_num = num
4  reversed_num = 0
5  while num > 0:
6      digit = num % 10
7      reversed_num = reversed_num * 10 + digit
8      num //= 10
9  if original_num == reversed_num:
10     print(original_num, "is a palindrome")
11 else:
12     print(original_num, "is not a palindrome")
13
14
```

Output :

```
PS C:\Users\THARUN\OneDrive\Desktop\sem_3_2\ai> & C:/Users/THARUN/AppData/Local/Programs/python.exe c:/Users/THARUN/OneDrive/Desktop/sem_3_2/ai/lab3.py
Enter a number to check if it is a palindrome: 12321
12321 is a palindrome
PS C:\Users\THARUN\OneDrive\Desktop\sem_3_2\ai> & C:/Users/THARUN/AppData/Local/Programs/python.exe c:/Users/THARUN/OneDrive/Desktop/sem_3_2/ai/lab3.py
Enter a number to check if it is a palindrome: 1234545
1234545 is not a palindrome
PS C:\Users\THARUN\OneDrive\Desktop\sem_3_2\ai> 
```

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 :

```
lab3.1.py > ...
1  #take input from user and print factorial of that number
2  num = int(input("Enter a number to calculate its factorial: "))
3  factorial = 1
4  for i in range(1, num + 1):
5      factorial *= i
6  print("The factorial of", num, "is", factorial)
7
8
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS QUERY RESULTS GITLENS

```
PS C:\Users\THARUN\OneDrive\Desktop\sem_3_2\ai> & C:/Users/THARUN/AppData/Local/Programs/python.exe c:/Users/THARUN/OneDrive/Desktop/sem_3_2/ai/lab3.1.py
Enter a number to calculate its factorial: 23
The factorial of 23 is 25852016738884976640000
PS C:\Users\THARUN\OneDrive\Desktop\sem_3_2\ai> |
```

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

```
lab3.2.py > ...
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")

PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS  QUERY RESULTS  GITLENS

PS C:\Users\THARUN\OneDrive\Desktop\sem_3_2\ai> & C:/Users/THARUN/AppData/Local/Programs/
python.exe c:/Users/THARUN/OneDrive/Desktop/sem_3_2/ai/lab3.2.py
Enter a number to check if it is an Armstrong number: 153
153 is an Armstrong number
PS C:\Users\THARUN\OneDrive\Desktop\sem_3_2\ai> & C:/Users/THARUN/AppData/Local/Programs/
python.exe c:/Users/THARUN/OneDrive/Desktop/sem_3_2/ai/lab3.2.py
Enter a number to check if it is an Armstrong number: 370
370 is an Armstrong number
PS C:\Users\THARUN\OneDrive\Desktop\sem_3_2\ai> & C:/Users/THARUN/AppData/Local/Programs/
python.exe c:/Users/THARUN/OneDrive/Desktop/sem_3_2/ai/lab3.2.py
Enter a number to check if it is an Armstrong number: 123
123 is not an Armstrong number
PS C:\Users\THARUN\OneDrive\Desktop\sem_3_2\ai> 
```

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.

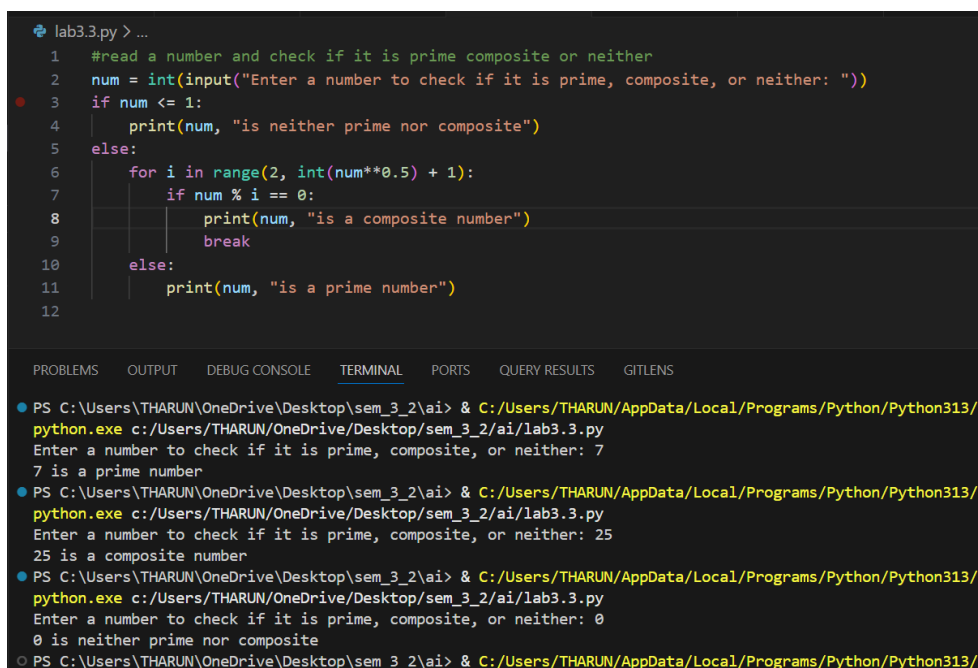
Task:

- Ensure proper input validation.
- Optimize the logic for efficiency.
- Compare the output with earlier prompting strategies.

Prompt :

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

Code – output



```
lab3.3.py > ...
1  #read a number and check if it is prime composite or neither
2  num = int(input("Enter a number to check if it is prime, composite, or neither: "))
3  if num <= 1:
4      print(num, "is neither prime nor composite")
5  else:
6      for i in range(2, int(num**0.5) + 1):
7          if num % i == 0:
8              print(num, "is a composite number")
9              break
10         else:
11             print(num, "is a prime number")
12
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS QUERY RESULTS GITLENS

```
PS C:\Users\THARUN\OneDrive\Desktop\sem_3_2\ai> & C:/Users/THARUN/AppData/Local/Programs/Python/Python313/python.exe c:/Users/THARUN/OneDrive/Desktop/sem_3_2/ai/lab3.3.py
Enter a number to check if it is prime, composite, or neither: 7
7 is a prime number

PS C:\Users\THARUN\OneDrive\Desktop\sem_3_2\ai> & C:/Users/THARUN/AppData/Local/Programs/Python/Python313/python.exe c:/Users/THARUN/OneDrive/Desktop/sem_3_2/ai/lab3.3.py
Enter a number to check if it is prime, composite, or neither: 25
25 is a composite number

PS C:\Users\THARUN\OneDrive\Desktop\sem_3_2\ai> & C:/Users/THARUN/AppData/Local/Programs/Python/Python313/python.exe c:/Users/THARUN/OneDrive/Desktop/sem_3_2/ai/lab3.3.py
Enter a number to check if it is prime, composite, or neither: 0
0 is neither prime nor composite

PS C:\Users\THARUN\OneDrive\Desktop\sem_3_2\ai> & C:/Users/THARUN/AppData/Local/Programs/Python/Python313/
```

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

```
lab3.4.py > ...
1  #take input from user and chcek if it is perfect number
2  num = int(input("Enter a number to check if it is a perfect number: "))
3  sum_of_divisors = 0
4  for i in range(1, num):
5      if num % i == 0:
6          sum_of_divisors += i
7  if sum_of_divisors == num:
8      print(num, "is a perfect number")
9  else:
10     print(num, "is not a perfect number")
11
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS QUERY RESULTS GITLENS

```
PS C:\Users\THARUN\OneDrive\Desktop\sem_3_2\ai> & C:/Users/THARUN/AppData/Local/Programs/Python/Python313/python.exe c:/Users/THARUN/OneDrive/Desktop/sem_3_2/ai/lab3.4.py
Enter a number to check if it is a perfect number: 6
6 is a perfect number
PS C:\Users\THARUN\OneDrive\Desktop\sem_3_2\ai> & C:/Users/THARUN/AppData/Local/Programs/Python/Python313/python.exe c:/Users/THARUN/OneDrive/Desktop/sem_3_2/ai/lab3.4.py
Enter a number to check if it is a perfect number: 28
28 is a perfect number
PS C:\Users\THARUN\OneDrive\Desktop\sem_3_2\ai> & C:/Users/THARUN/AppData/Local/Programs/Python/Python313/python.exe c:/Users/THARUN/OneDrive/Desktop/sem_3_2/ai/lab3.4.py
Enter a number to check if it is a perfect number: 23
23 is not a perfect number
```

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

```
lab3.5.py > ...
1  #take input from user and check even or odd
2  num = int(input("Enter a number to check if it is even or odd: "))
3  if num % 2 == 0:
4      print(num, " even number")
5  else:
6      print(num, " odd number")
7
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS QUERY RESULTS GITLENS

```
PS C:\Users\THARUN\OneDrive\Desktop\sem_3_2\ai> & C:/Users/THARUN/AppData/Local/Programs/python.exe c:/Users/THARUN/OneDrive/Desktop/sem_3_2/ai/lab3.5.py
Enter a number to check if it is even or odd: 15
15 odd number
PS C:\Users\THARUN\OneDrive\Desktop\sem_3_2\ai> & C:/Users/THARUN/AppData/Local/Programs/python.exe c:/Users/THARUN/OneDrive/Desktop/sem_3_2/ai/lab3.5.py
Enter a number to check if it is even or odd: 16
16 even number
PS C:\Users\THARUN\OneDrive\Desktop\sem_3_2\ai> 
```