

Assignment – 3.1

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.

The screenshot shows a terminal window with the following content:

```
1 #Question 1
2 #generate a python function to check the give number is palindrome or not
3 def is_palindrome(number):
4     # Convert the number to string
5     str_num = str(number)
6     # Check if the string is equal to its reverse
7     return str_num == str_num[::-1]
8 number = int(input("Enter a number: "))
9 if is_palindrome(number):
10     print(f"{number} is a palindrome.")
11 else:
12     print(f"{number} is not a palindrome.")

PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL    PORTS

Microsoft Windows [Version 10.0.26200.7623]
(c) Microsoft Corporation. All rights reserved.

C:\Users\sathw\OneDrive\Documents\Samyuu\3-2\AI-assistant_coding>C:/Users/sathw/AppData/Local/Programs/Python/Python311/python.exe C:/Users/sathw/OneDrive/Documents/Samyuu/3-2/AI-assistant_coding/Assign-3.py
Enter a number: 45
45 is not a palindrome.
```

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.

```
14 #Question 2
15 #generate a python function to check the Factorial Calculation
16 def factorial(n):
17     if n < 0:
18         return "Factorial is not defined for negative numbers."
19     elif n == 0 or n == 1:
20         return 1
21     else:
22         result = 1
23         for i in range(2, n + 1):
24             result *= i
25         return result
26 num = int(input("Enter a number to calculate its factorial: "))
27 print(f"The factorial of {num} is {factorial(num)}.")
28
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
Microsoft Windows [Version 10.0.26200.7623]
(c) Microsoft Corporation. All rights reserved.
```

```
C:\Users\sathw\OneDrive\Documents\Samyuu\3-2\AI-assistent_coding>C:/Users/s
thon.exe c:/Users/sathw/OneDrive/Documents/Samyuu/3-2/AI-assistent_coding/A
Enter a number to calculate its factorial: 5
The factorial of 5 is 120.
```

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.

```

29  #Question 3
30  #generate a python function to check the Armstrong Number
31  def is_armstrong(number):
32      # Convert the number to string to easily iterate over digits
33      str_num = str(number)
34      num_digits = len(str_num)
35      sum_of_powers = sum(int(digit) ** num_digits for digit in str_num)
36      return sum_of_powers == number
37  number = int(input("Enter a number: "))
38  if is_armstrong(number):
39      print(f"{number} is an Armstrong number.")
40  else:
41      print(f"{number} is not an Armstrong number.")

PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL    PORTS

Microsoft Windows [Version 10.0.26200.7623]
(c) Microsoft Corporation. All rights reserved.

C:\Users\sathw\OneDrive\Documents\Samyuu\3-2\AI-assistent_coding>C:/Users/sathw/A
thon.exe c:/Users/sathw/OneDrive/Documents/Samyuu/3-2/AI-assistent_coding/Assign-
Enter a number: 470
470 is not an Armstrong number.

C:\Users\sathw\OneDrive\Documents\Samyuu\3-2\AI-assistent_coding>C:/Users/sathw/A
thon.exe c:/Users/sathw/OneDrive/Documents/Samyuu/3-2/AI-assistent_coding/Assign-
Enter a number: 112
112 is not an Armstrong number.

```

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.

```

43  #Question 4
44  #generate a python function to check the Optimized Number Classification
45  def classify_number(number):
46      if number < 0:
47          return "Negative"
48      elif number == 0:
49          return "Zero"
50      elif number % 2 == 0:
51          return "Even"
52      else:
53          return "Odd"
54  num = int(input("Enter a number: "))
55  classification = classify_number(num)
56  print(f"The number {num} is classified as: {classification}.")
57

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
C:\Users\sathw\OneDrive\Documents\Samyuu\3-2\AI-assistent_coding>C:/Users/sathw/App
thon.exe c:/Users/sathw/OneDrive/Documents/Samyuu/3-2/AI-assistent_coding/Assign-3.
Enter a number: 520
The number 520 is classified as: Even.
```

```
C:\Users\sathw\OneDrive\Documents\Samyuu\3-2\AI-assistent_coding>C:/Users/sathw/App
thon.exe c:/Users/sathw/OneDrive/Documents/Samyuu/3-2/AI-assistent_coding/Assign-3.
Enter a number: 153
The number 153 is classified as: Odd.
```

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.

```
58 #Question 5
59 #generate a python function to check the Perfect Number Check
60 def is_perfect(number):
61     if number < 1:
62         return False
63     sum_of_divisors = sum(i for i in range(1, number) if number % i == 0)
64     return sum_of_divisors == number
65 number = int(input("Enter a number: "))
66 if is_perfect(number):
67     print(f"{number} is a perfect number.")
68 else:
69     print(f"{number} is not a perfect number.")

PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL    PORTS

(c) Microsoft Corporation. All rights reserved.

C:\Users\sathw\OneDrive\Documents\Samyuu\3-2\AI-assistent_coding>C:/Users/sathw/AppD
thon.exe c:/Users/sathw/OneDrive/Documents/Samyuu/3-2/AI-assistent_coding/Assign-3.p
Enter a number: 85
85 is not a perfect number.

C:\Users\sathw\OneDrive\Documents\Samyuu\3-2\AI-assistent_coding>C:/Users/sathw/AppD
thon.exe c:/Users/sathw/OneDrive/Documents/Samyuu/3-2/AI-assistent_coding/Assign-3.p
Enter a number: 90
90 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.

```
71  #Question 6
72  #generate a python function to check Even or Odd Classification with Validation
73  def even_or_odd(number):
74      if number % 2 == 0:
75          return "Even"
76      else:
77          return "Odd"
78  try:
79      num = int(input("Enter an integer: "))
80      result = even_or_odd(num)
81      print(f"The number {num} is {result}.")
82  except ValueError:
83      print("Invalid input! Please enter a valid integer.")

PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL    PORTS
```

thon.exe c:/Users/sathw/OneDrive/Documents/Samyuu/3-2/AI-assistant_coding/Assign-3.py
Enter an integer: 52
The number 52 is Even.

C:\Users\sathw\OneDrive\Documents\Samyuu\3-2\AI-assistant_coding>C:/Users/sathw/AppData/Local
thon.exe c:/Users/sathw/OneDrive/Documents/Samyuu/3-2/AI-assistant_coding/Assign-3.py
Enter an integer: 47
The number 47 is Odd.