

# **Assignment-3.1**

**Name-Naga Rishik Reddy**

**Batch-02**

**Hall ticket num-2303A51089**

**Experiment – Prompt Engineering Techniques**

**Task Description :**

**Design and refine prompts using different prompting strategies to generate  
Python programs  
for basic computational problems.**

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 Python script named `assignment.py` in a code editor. The script defines a function `is_palindrome` that checks if a given number is a palindrome. It uses string slicing to reverse the number and compares it with the original. The main part of the script takes user input, calls the function, and prints the result. A terminal window below shows the script running and outputting "121 is a palindrome number." and "165 is not a palindrome number." The status bar at the bottom indicates the current file is `c:/Users/Rishi/Documents/Naga Rishik/AI assisted/assignment.py`.

```
assignment.py > ...
2 write a python program to find a number is palindrome or not
3 ...
4
5
6 def is_palindrome(num):
7     if num < 0:
8         return False
9     return str(num) == str(num)[::-1]
10 try:
11     num = int(input("Enter a number:"))
12     if is_palindrome(num):
13         print(f"{num} is a palindrome number.")
14     else:
15         print(f"{num} is not a palindrome number.")
16 except ValueError:
17     print("Accept only numbers")
18
```

PROBLEMS 1 OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS C:\Users\Rishi\Documents\Naga Rishik\AI assisted> & C:/Users/Rishi/miniconda3/python.exe "c:/Users/Rishi/Documents/Naga Rishik/AI assisted/assignment.py"  
Enter a number:121  
121 is a palindrome number.  
PS C:\Users\Rishi\Documents\Naga Rishik\AI assisted> & C:/Users/Rishi/miniconda3/python.exe "c:/Users/Rishi/Documents/Naga Rishik/AI assisted/assignment.py"  
Enter a number:165  
165 is not a palindrome number.  
PS C:\Users\Rishi\Documents\Naga Rishik\AI assisted>

Activate Windows  
Go to Settings to activate Windows.

LN 18, COL 5 SPACES: 4 UFT-8 CRLF {} Python 3.13.5 (base) (Go Live)

## 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

```
102 num = 5
103 display Factorial of 5 is 120
104 ...
105 def factorial(n):
106     if n < 0:
107         return None
108     elif n == 0:
109         return 1
110     else:
111         result = 1
112         for i in range(1, n + 1):
113             result *= i
114         return result
115
116 try:
117     num = int(input("Enter a number: "))
118     fact = factorial(num)
119     if fact is None:
120         print("Factorial is not defined for negative numbers")
121     else:
122         print(f"Factorial of {num} is {fact}")
123 except ValueError:
124     print("Accept only numbers")
125
```

The screenshot shows a code editor interface with a dark theme. At the top, there are tabs for PROBLEMS, OUTPUT, DEBUG CONSOLE, TERMINAL, PORTS, and SPELL CHECKER. The TERMINAL tab is selected. Below the tabs, the terminal window displays the following text:  
/usr/local/bin/python3 "/Users/aravindreddy/Desktop/My-Information/Collage/AI Assisted Coding/Assignments/Assignment-#3.py"  
● (base) Aravindreddy@Aravindreddy-MacBook-Pro ~ %  
\$ Enter a number: 5  
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.

```
assignment.py > ...
2   write a python program to check a number is armstrong or not
3
4   num=153
5   display 153 is an armstrong number
6   num=123
7   display 123 is not an armstrong number
8   num=-370
9   display -370 is not an armstrong number
10  num='abc'
11  display no alphabet allowed
12  num=0
13  display 0 is an armstrong number
...
14
15 def is_armstrong(num):
16     if num < 0:
17         return False
18     num_str = str(num)
19     orders = len(num_str)
20     sum_of_powers = sum(int(digit) ** orders for digit in num_str)
21     return sum_of_powers == num
22 try:
23     number = int(input("Enter a number to check if it is an Armstrong number: "))
24     if is_armstrong(number):
25         print(f"{number} is an Armstrong number.")
26     else:
27         print(f"{number} is not an Armstrong number.")
28 except ValueError:
29     print("No alphabet allowed.")

PROBLEMS 1 OUTPUT DEBUG CONSOLE TERMINAL PORTS
Enter a number to check if it is an Armstrong number: 153
153 is an Armstrong number.
Activate Windows [ ] Python [ ]
Go to Settings to activate Windows [ ] Python [ ]
PS C:\Users\Rishi\Documents\Naga Rishi\AI assisted> & C:/Users/Rishi/miniconda3/python.exe "c:/Users/Rishi/Documents/Naga
Ln 30, Col 5 Spaces: 4 UTF-8 CRLF {} Python 3.13.5 (base) (n) Go Live
```

(Optional Extension)

#### 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.

The screenshot shows a code editor in VS Code with a dark theme. A Python script named `assignment.py` is open, containing logic to determine if a number is prime, composite, or neither. The terminal below shows the execution of the script and its output for various inputs.

```
assignment.py > check_prime_composite > i
1 ...
2 num=5
3 display num is prime number
4 num=8
5 display num is composite number
6 num=-3
7 display prime number is greater than 1
8 num='xyz'
9 display accept only integer value
10 num=0
11 display neither prime nor composite
12 ...
13 def check_prime_composite(num):
14     if num<=1:
15         return "neither prime nor composite" if num==0 else "prime number is greater than 1"
16     for i in range(2, int(num**0.5) + 1):
17         if num % i == 0:
18             return "composite number"
19     return "prime number"
20 try:
21     num = int(input("Enter a number: "))
PROBLEMS 1 OUTPUT DEBUG CONSOLE TERMINAL PORTS
5 is a prime number.
PS C:\Users\Rishi\Documents\Naga Rishik\AI assisted> & C:/Users/Rishi/miniconda3/python.exe "c:/Users/Rishi/Documents/Naga Rishik/AI assisted/assignment.py"
Enter a number: cyd
accept only integer value
PS C:\Users\Rishi\Documents\Naga Rishik\AI assisted> & C:/Users/Rishi/miniconda3/python.exe "c:/Users/Rishi/Documents/Naga Rishik/AI assisted/assignment.py"
Enter a number: 0
0 is a neither prime nor composite.
PS C:\Users\Rishi\Documents\Naga Rishik\AI assisted>
```

### 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

The screenshot shows a code editor with a dark theme and a terminal window below it. The code editor has a file named 'assignment.py' open, containing the following Python script:

```
1 assignment.py > ...
2     write a python program to check a number is perfect number or not
3     ...
4     def is_perfect_number(num):
5         if num<1:
6             return False
7         divisors_sum = sum(i for i in range(1, num) if num % i == 0)
8         return divisors_sum == num
9     try:
10         num = int(input("Enter a number"))
11         if is_perfect_number(num):
12             print(f"{num} is a perfect number.")
13         else:
14             print(f"{num} is not a perfect number.")
15     except ValueError:
16         print("Accept only numbers")
17
```

The terminal window shows the execution of the script. It prompts for a number, accepts '1' (which is not a perfect number), and then accepts '6' (which is a perfect number). The terminal interface includes tabs for PROBLEMS, OUTPUT, DEBUG CONSOLE, TERMINAL, and PORTS, and a sidebar with multiple Python environments.

#### 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

The screenshot shows a code editor interface with a Python file named `assignment.py`. The code defines a function `check_even_odd` to check if a number is even or odd, and a `try-except` block to handle user input. The terminal below shows the execution of the script and its output for different inputs.

```
assignment.py > ...
1   ...
2   write a python program to check whether a number is even or odd
3   num=24
4   display 24 is even number
5   num=95
6   display 95 is odd number
7   num='abc'
8   display it is not a number
9   ...
10  def check_even_odd(num):
11      return "Even number " if num % 2 == 0 else "Odd number "
12  try:
13      num = int(input("Enter a number: "))
14      result = check_even_odd(num)
15      print(f"{num} is an {result}")
16  except ValueError:
17      print("It is not a number")
18

PROBLEMS 1 OUTPUT DEBUG CONSOLE TERMINAL PORTS
```

Rishik/AI assisted/assignment.py"  
Enter a number: 93  
93 is an Odd number  
PS C:\Users\Rishi\Documents\Naga Rishik\AI assisted> & C:/Users/Rishi/miniconda3/python.exe "c:/Users/Rishi/Documents/Naga Rishik/AI assisted/assignment.py"  
Enter a number: rishi  
It is not a number  
PS C:\Users\Rishi\Documents\Naga Rishik\AI assisted> & C:/Users/Rishi/miniconda3/python.exe "c:/Users/Rishi/Documents/Naga Rishik/AI assisted/assignment.py"  
Enter a number: 24  
24 is an Even number  
PS C:\Users\Rishi\Documents\Naga Rishik\AI assisted>

Activate Windows  
Go to Settings to activate Windows.