

# LAB ASSIGNMENT - 03

---

**Hall Ticket No :** 2303a51027

**Name :** Aravind Reddy

**Course Title :** AI Assistant Coding

---

## **Experiment – Prompt Engineering Techniques**

### **Task Description :**

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

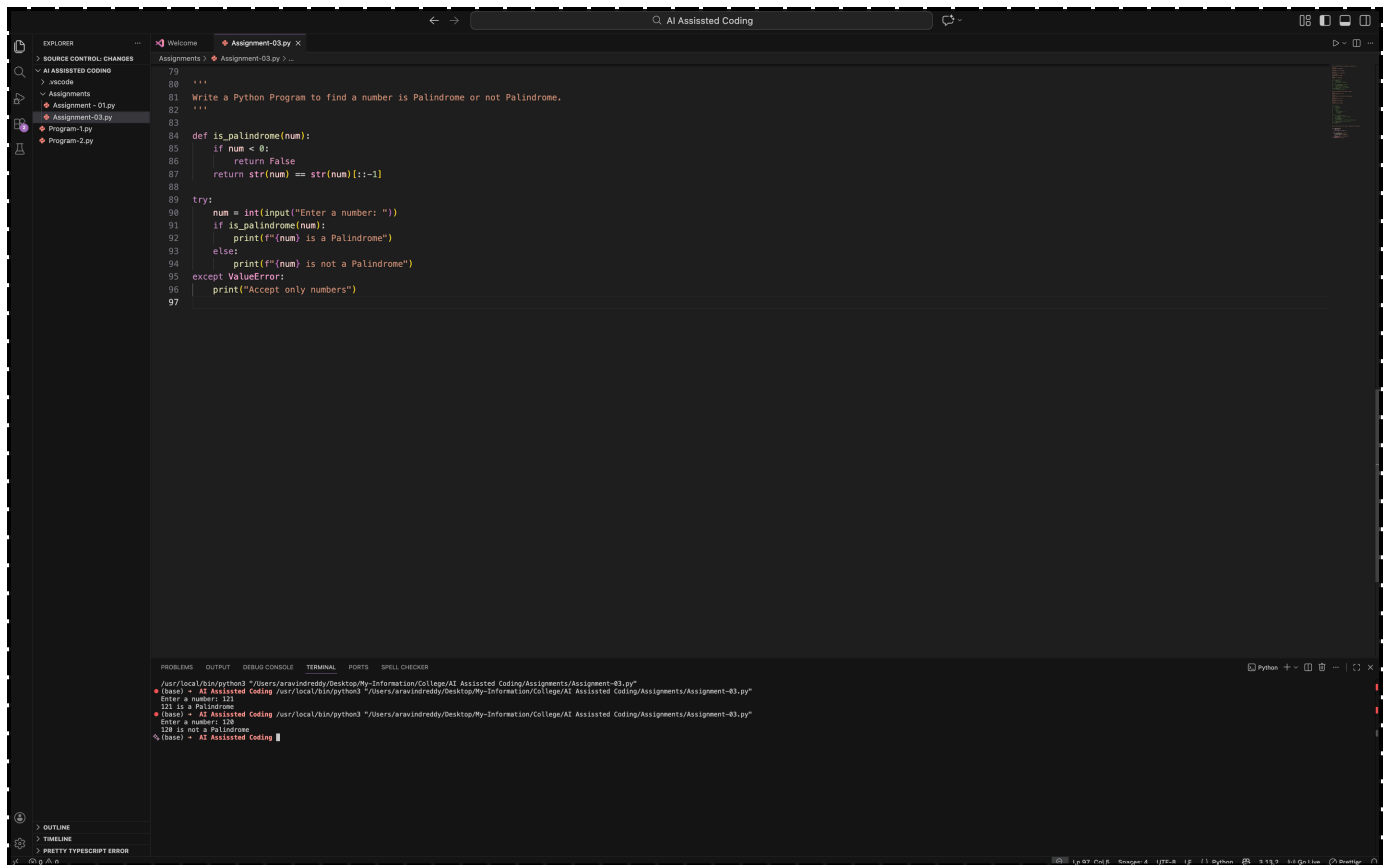
---

### **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.



## 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.

The screenshot shows a VS Code editor with a Python file named 'Assignment-03.py'. The code defines a factorial function and uses it to calculate the factorial of 5. The terminal output shows the execution of the program, which prompts the user to enter a number and displays the factorial result.

```
98
99
100 Write a Python Program to find factorial of a Number.
101
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
```

Terminal Output:

```
/usr/local/bin/python3 /Users/aravindreddy/Desktop/My-Information/College/AI Assisted Coding/Assignments/Assignment-03.py
(base) ~ - AI Assisted Coding /usr/local/bin/python3 /Users/aravindreddy/Desktop/My-Information/College/AI Assisted Coding/Assignments/Assignment-03.py
Enter a number: 5
Factorial of 5 is 120
(base) ~ - AI Assisted Coding /usr/local/bin/python3 /Users/aravindreddy/Desktop/My-Information/College/AI Assisted Coding/Assignments/Assignment-03.py
Enter a number: 5
Factorial of 5 is 120
(base) ~ - AI Assisted Coding
```

### 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.**

```
126
127 Write a Python Program to check a Number is Armstrong or not.
128
129 num = 153
130 display 153 is an Armstrong number
131
132 num = 123
133 display 123 is not an Armstrong number
134
135 num = -370
136 display -370 is not an Armstrong number
137
138 num = 'abc'
139 display accept only numbers
140
141 num = 0
142 display 0 is an Armstrong number
143 '''
144
145 def is_armstrong(num):
146     if num < 0:
147         return False
148     num_str = str(num)
149     order = len(num_str)
150     sum_of_powers = sum(int(digit) ** order for digit in num_str)
151     return sum_of_powers == num
152
153 try:
154     num = int(input("Enter a number: "))
155     if is_armstrong(num):
156         print(f"{num} is an Armstrong number")
157     else:
158         print(f"{num} is not an Armstrong number")
159 except ValueError:
160     print("Accept only numbers")
161
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS SPELL CHECKER

```
/usr/local/bin/python3 "/Users/aravindreddy/Desktop/My-Information/College/AI Assisted Coding/Assignments/Assignment-03.py"
(base) ~ AI Assisted Coding /usr/local/bin/python3 "/Users/aravindreddy/Desktop/My-Information/College/AI Assisted Coding/Assignments/Assignment-03.py"
Enter a number: 153
153 is an Armstrong number
(base) ~ AI Assisted Coding /usr/local/bin/python3 "/Users/aravindreddy/Desktop/My-Information/College/AI Assisted Coding/Assignments/Assignment-03.py"
Enter a number: 123
123 is not an Armstrong number
(base) ~ AI Assisted Coding /usr/local/bin/python3 "/Users/aravindreddy/Desktop/My-Information/College/AI Assisted Coding/Assignments/Assignment-03.py"
Enter a number: -370
-370 is not an Armstrong number
(base) ~ AI Assisted Coding /usr/local/bin/python3 "/Users/aravindreddy/Desktop/My-Information/College/AI Assisted Coding/Assignments/Assignment-03.py"
Enter a number: query
Accept only numbers
(base) ~ AI Assisted Coding /usr/local/bin/python3 "/Users/aravindreddy/Desktop/My-Information/College/AI Assisted Coding/Assignments/Assignment-03.py"
Enter a number: 0
0 is an Armstrong number
(base) ~ AI Assisted Coding
```

## (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 displays the Visual Studio Code interface with a dark theme. The Explorer sidebar on the left shows a project structure with folders like 'SOURCE CONTROL: CHANGES' and 'AI ASSISTED CODING'. The main editor window contains a file named 'Assignment-03.py' with the following Python code:

```
161  
162  
163  
164 '''  
165 num = 5  
166 display num is a prime number  
167  
168 num = 8  
169 display num is a composite number  
170  
171 num = -3  
172 display prime number is greater than 1  
173  
174 num = 'xyz'  
175 display accept only numbers  
176  
177 num = 0  
178 display neither prime nor composite  
179  
180 '''  
181  
182 def check_prime_composite(num):  
183     if num <= 1:  
184         return "neither prime nor composite" if num == 0 else "prime number is greater than 1"  
185     for i in range(2, int(num**0.5) + 1):  
186         if num % i == 0:  
187             return "composite number"  
188     return "prime number"  
189  
190 try:  
191     num = int(input("Enter a number: "))  
192     result = check_prime_composite(num)  
193     print(f"{num} is a {result}")  
194 except ValueError:  
195     print("accept only numbers")
```

Below the code editor, the TERMINAL panel shows the execution of the script using the command prompt. It includes prompts for input and corresponding outputs for various test cases:

```
/usr/local/bin/python3 "/Users/aravindreddy/Desktop/My-Information/Collage/AI Assisted Coding/Assignments/Assignment-03.py"  
● (base) ~ AI Assisted Coding /usr/local/bin/python3 "/Users/aravindreddy/Desktop/My-Information/Collage/AI Assisted Coding/Assignments/Assignment-03.py"  
Enter a number: 5  
5 is a prime number  
● (base) ~ AI Assisted Coding /usr/local/bin/python3 "/Users/aravindreddy/Desktop/My-Information/Collage/AI Assisted Coding/Assignments/Assignment-03.py"  
Enter a number: 8  
8 is a composite number  
● (base) ~ AI Assisted Coding /usr/local/bin/python3 "/Users/aravindreddy/Desktop/My-Information/Collage/AI Assisted Coding/Assignments/Assignment-03.py"  
Enter a number: 0  
0 is a neither prime nor composite  
● (base) ~ AI Assisted Coding /usr/local/bin/python3 "/Users/aravindreddy/Desktop/My-Information/Collage/AI Assisted Coding/Assignments/Assignment-03.py"  
Enter a number: -2  
-2 is a prime number is greater than 1  
% (base) ~ AI Assisted Coding █
```

### 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 VS Code editor with a Python file named 'Assignment-03.py'. The code defines a function 'is\_perfect\_number(num)' that checks if a number is perfect by summing its divisors. It then uses a try-except block to handle user input, printing 'is a Perfect Number' or 'is not a Perfect Number' based on the function's result. The terminal at the bottom shows the execution of the script, with prompts for 'Enter a number:' and corresponding outputs for inputs 6, 16, and 0.

```
197
198
199
200 '''
201 Write a Python Program to check a number is Perfect Number or Not.
202 '''
203
204 def is_perfect_number(num):
205     if num < 1:
206         return False
207     divisors_sum = sum(i for i in range(1, num) if num % i == 0)
208     return divisors_sum == num
209
210 try:
211     num = int(input("Enter a number: "))
212     if is_perfect_number(num):
213         print(f"{num} is a Perfect Number")
214     else:
215         print(f"{num} is not a Perfect Number")
216 except ValueError:
217     print("Accept only numbers")
218
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS SPELL CHECKER

```
/usr/local/bin/python3 "/Users/aravindreddy/Desktop/My-Information/College/AI Assisted Coding/Assignments/Assignment-03.py"
(base) - AI Assisted Coding /usr/local/bin/python3 "/Users/aravindreddy/Desktop/My-Information/College/AI Assisted Coding/Assignments/Assignment-03.py"
Enter a number: 6
6 is a Perfect Number
(base) - AI Assisted Coding /usr/local/bin/python3 "/Users/aravindreddy/Desktop/My-Information/College/AI Assisted Coding/Assignments/Assignment-03.py"
Enter a number: 16
16 is not a Perfect Number
(base) - AI Assisted Coding /usr/local/bin/python3 "/Users/aravindreddy/Desktop/My-Information/College/AI Assisted Coding/Assignments/Assignment-03.py"
Enter a number: 0
0 is not a Perfect Number
(base) - AI Assisted Coding
```

## 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

