

AI ASSISTED CODING

ASSIGNMENT-3

N. Akshitha

2303A52116

Batch-40

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:

```
#Write a python program to find whether the given number is palindrome or not
```

```
# Check if the string representation of the number is equal to its reverse
```

```
# Take input from the user
```

```
# Check and display whether the number is a palindrome
```

Code

and

Output:

The screenshot shows a code editor interface with two tabs: 'Assignment-3.py' and 'simple.py'. The 'Assignment-3.py' tab contains the following Python code:

```
1 #Question-1
2 #Write a python program to find whether the given number is palindrome or not
3 # Check if the string representation of the number is equal to its reverse
4 # Take input from the user
5 # Check and display whether the number is a palindrome
6 def is_palindrome(number):
7     str_num = str(number)
8     return str_num == str_num[::-1]
9 user_input = int(input("Enter a number: "))
10 if is_palindrome(user_input):
11     print(f"{user_input} is a palindrome.")
12 else:
13     print(f"{user_input} is not a palindrome.")
```

The terminal below the code editor shows the execution of the script and its output:

```
PS C:\Users\nalla\OneDrive\Documents\Desktop\AI-ASSIS> & c:/Users/nalla/appData/local/Programs/Python/Python313/python.exe c:/Users/nalla/OneDrive/Documents/Desktop/AI-ASSIS/Assignment-3.py
Enter a number: 121
121 is a palindrome.

PS C:\Users\nalla\OneDrive\Documents\Desktop\AI-ASSIS> & c:/Users/nalla/appData/local/Programs/Python/Python313/python.exe c:/Users/nalla/OneDrive/Documents/Desktop/AI-ASSIS/Assignment-3.py
Enter a number: 156
156 is not a palindrome.

PS C:\Users\nalla\OneDrive\Documents\Desktop\AI-ASSIS>
```

The interface includes a sidebar for 'RECENT SESSIONS' and a 'Build with Agent' section with a message about AI responses being inaccurate.

Analysis:

Works correctly for basic positive numbers

Negative numbers fail due to string behavior, not real logic

No input type checking is done

Relies only on string conversion

Suitable only for simple or beginner-level tasks

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:

```
#Write a python program to find given number is factorial or not e.g input=5 output=120
```

```
# Base case: factorial of 0 or 1 is 1
```

```
# Recursive case
```

```
# Take input from the user
```

Code and Output:

The screenshot shows a code editor interface with two tabs: 'Assignment-3.py' and 'simple.py'. The 'Assignment-3.py' tab contains the following Python code:

```
14 #Question-2
15 #write a python program to find given number is factorial or not e.g input=5 output=120
16 Tabnine | Edit | Test | Explain | Document
17 def factorial(n):
18     # Base case: factorial of 0 or 1 is 1
19     if n == 0 or n == 1:
20         return 1
21     else:
22         # Recursive case
23         return n * factorial(n - 1)
24 user_input = int(input("Enter a number to find its factorial: "))
25 result = factorial(user_input)
26 print(f"The factorial of {user_input} is: {result}")
27
```

The terminal below shows the execution of the code:

```
PS C:\Users\nalla\OneDrive\Documents\Desktop\AI-ASSIS> & c:/Users/nalla/AppData/Local/Programs/Python/Python313/python.exe c:/users/nalla/OneDrive/Documents/Desktop/AI-ASSIS/Assignment-3.py
Enter a number to find its factorial: 6
The factorial of 6 is: 720
PS C:\Users\nalla\OneDrive\Documents\Desktop\AI-ASSIS> & c:/Users/nalla/AppData/Local/Programs/Python/Python313/python.exe c:/users/nalla/OneDrive/Documents/Desktop/AI-ASSIS/Assignment-3.py
Enter a number to find its factorial: 5
The Factorial of 5 is: 120
PS C:\Users\nalla\OneDrive\Documents\Desktop\AI-ASSIS>
```

A sidebar on the right is titled 'Build with Agent' and includes a note: 'AI responses may be inaccurate. Generate Agent Instructions to onboard AI onto your codebase.'

Analysis:

One-shot code clearly handles the base case ($0! = 1$)

Zero-shot version misses explicit handling of zero

One-shot solution is easier to understand and more structured

One-shot result is mathematically more correct

Example helps the AI generate safer and clearer logic.

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.

Prompt:

```
#write a python code to check whether the given number is armstrong or not
```

```
#Input: 153 → Output: Armstrong Number
```

```
#Input: 370 → Output: Armstrong Number
```

```
#Input: 123 → Output: Not an Armstrong Number
```

```
# Calculate the sum of each digit raised to the power of the number of digits
```

```
# Check if the sum of powers is equal to the original number
```

```
# Take input from the user
```

```
# Check and display whether the number is an Armstrong number
```

Code and Output:

The screenshot shows a code editor with two files open: `Assignment-3.py` and `simple.py`. The `Assignment-3.py` file contains the following code:

```
27 #question-3
28 #write a python code to check whether the given number is armstrong or not
29 #Input: 153 → Output: Armstrong Number
30 #Input: 370 → Output: Armstrong Number
31 #Input: 123 → Output: Not an Armstrong Number
32 # Calculate the sum of each digit raised to the power of the number of digits
33 # Check if the sum of powers is equal to the original number
34 # Take input from the user
35 # Check and display whether the number is an Armstrong number
36 def is_armstrong(number):
37     num_str = str(number)
38     num_digits = len(num_str)
39     sum_of_powers = sum(int(digit) ** num_digits for digit in num_str)
40     return sum_of_powers == number
41 user_input = int(input("Enter a number: "))
42 if is_armstrong(user_input):
43     print(f"{user_input} is an Armstrong Number.")
44 else:
45     print(f"{user_input} is not an Armstrong Number.")
```

The `simple.py` file contains a single line of code: `print("Hello World")`.

To the right of the code editor, there is an AI interface window titled "AI-ASSIS". It shows a "RECENT SESSIONS" list with one entry: "General greeting inquiry" (Completed, Local • 1 wk). Below it is a "Build with Agent" section with a message: "AI responses may be inaccurate. Generate Agent Instructions to onboard AI onto your codebase." A small AI icon is also present.

At the bottom of the screen, a terminal window shows the execution of the Python script. The output is:

```
PS C:\Users\nalla\OneDrive\Documents\Desktop\AI-ASSIS> & C:/Users/nalla/AppData/Local/Programs/Python/Python311/python.exe c:/Users/nalla/OneDrive/Documents/Desktop\AI-ASSIS & C:/Users/nalla/AppData/Local/Programs/Python/Python311/python.exe c:/Users/nalla/OneDrive/Documents/Desktop\AI-ASSIS\Assignment-3.py
PS C:\Users\nalla\OneDrive\Documents\Desktop\AI-ASSIS> & C:/Users/nalla/AppData/Local/Programs/Python/Python311/python.exe c:/Users/nalla/OneDrive/Documents/Desktop\AI-ASSIS & C:/Users/nalla/AppData/Local/Programs/Python/Python311/python.exe c:/Users/nalla/OneDrive/Documents/Desktop\AI-ASSIS\Assignment-3.py
PS C:\Users\nalla\OneDrive\Documents\Desktop\AI-ASSIS> & C:/Users/nalla/AppData/Local/Programs/Python/Python311/python.exe c:/Users/nalla/OneDrive/Documents/Desktop\AI-ASSIS\Assignment-3.py
Enter a number: 370
370 is an Armstrong Number.
PS C:\Users\nalla\OneDrive\Documents\Desktop\AI-ASSIS> & C:/Users/nalla/AppData/Local/Programs/Python/Python311/python.exe c:/Users/nalla/OneDrive/Documents/Desktop\AI-ASSIS\Assignment-3.py
PS C:\Users\nalla\OneDrive\Documents\Desktop\AI-ASSIS> & C:/Users/nalla/AppData/Local/Programs/Python/Python311/python.exe c:/Users/nalla/OneDrive/Documents/Desktop\AI-ASSIS\Assignment-3.py
Enter a number: 67
67 is not an Armstrong Number.
```

The system status bar at the bottom indicates: 28°C, Sunny, 14:22, 19-01-2026.

Analysis:

Giving examples helps the AI understand what kind of answer is expected.

Multiple examples make the code cleaner and more logical.

Showing both correct and incorrect cases avoids confusion.

Testing small numbers like 0 and 1 ensures the function works properly.

Checking wrong inputs makes the program safer and more reliable.

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:

#Write a python program to find the given number is prime,composite or neither

```
# Check if the number is less than 2 (neither prime nor composite)
```

Check for factors from 2 to the square root of the number

```
# Take input from the user
```

```
# Check and display whether the number is prime, composite, or neither
```

Code and Output:

The screenshot shows a dark-themed instance of Visual Studio Code (VS Code) with several tabs open. The active tab contains Python code for determining if a number is prime or composite. The code includes a function `check_prime_composite` and a user input section. Below the editor is a terminal window displaying command-line interactions with the Python interpreter, showing the execution of the script and its output for various numbers. The status bar at the bottom provides standard system information like battery level and network status.

```
File Edit Selection View Go Run Terminal Help < > Q AI-ASSIS
Assignment-3.py simple.py Assignment-3.py x
Assignment-3.py ...
simple.py
#Question-4
#Write a python program to find the given number is prime,composite or neither
# Check if the number is less than 2 (neither prime nor composite)
# Check for factors from 2 to the square root of the number
Tabnine | Edit | Test | Explain | Document
50 def check_prime_composite(number):
51     if number < 2:
52         return "Neither prime nor composite"
53     for i in range(2, int(number**0.5) + 1):
54         if number % i == 0:
55             return "Composite"
56     return "Prime"
57 # Take input from the user
58 user_input = int(input("Enter a number: "))
59 result = check_prime_composite(user_input)
60 print(f"{user_input} is {result}.")
61
62
63
Build with Agent
AI responses may be inaccurate.
Generate Agent Instructions to onboard AI onto your codebase.
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS C:\Users\Nalla\OneDrive\Documents\Desktop\AI-ASSIS> & C:/Users/nalla/AppData/Local/Programs/Python/Python310/python.exe c:/users/nalla/OneDrive/Documents/Desktop/AI-ASSIS/Assignment-3.py
Enter a number: 67
67 is Prime.
PS C:\Users\Nalla\OneDrive\Documents\Desktop\AI-ASSIS> & C:/Users/nalla/AppData/Local/Programs/Python/Python310/python.exe c:/users/nalla/OneDrive/Documents/Desktop/AI-ASSIS/Assignment-3.py
Enter a number: 126
126 is Composite.
PS C:\Users\Nalla\OneDrive\Documents\Desktop\AI-ASSIS>
Ln 61, Col 1 Spaces: 4 UFT-8 CRLF (.) Python 3.13.7 ⚡ Port: 5500
Launchpad AFG - WI Match
CHAT RECENT SESSIONS
General greeting inquiry
Completed
Show More
ENG IN
14:31 19-01-2026
```

Analysis:

Clear instructions help the AI understand exactly what is needed.

Input validation avoids crashes from wrong inputs.

Efficient logic makes the program faster and smarter.

The AI performs better than with simple prompts.

Results are clearer and more reliable.

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:

```
#Write python program to find whether given input is perfect number or not
# Calculate the sum of divisors of the number
# Check if the sum of divisors is equal to the original number
# Take input from the user
# Check and display whether the number is a perfect number
```

Code and Output:

The screenshot shows a code editor interface with several tabs open. The main tab contains the following Python code:

```
62 #Question-5
63 #write python program to find whether given input is perfect number or not
64 # Calculate the sum of divisors of the number
65 # check if the sum of divisors is equal to the original number
66 def is_perfect_number(number):
67     if number < 1:
68         return False
69     divisors_sum = sum(i for i in range(1, number) if number % i == 0)
70     return divisors_sum == number
71 # Take input from the user
72 user_input = int(input("Enter a number: "))
73 # Check and display whether the number is a perfect number
74 if is_perfect_number(user_input):
75     print(f"{user_input} is a perfect number.")
76 else:
77     print(f"{user_input} is not a perfect number.")
```

Below the code editor is a terminal window showing the execution of the script. The terminal output is as follows:

```
PS C:\Users\nalla\OneDrive\Documents\Desktop\AI-ASSIS> & c:/users/nalla/appdata/local/python/python313/python.exe c:/users/nalla/OneDrive/Documents/Desktop/AI-ASSIS/Assignment-3.py
Enter a number: 67
67 is Prime.
PS C:\Users\nalla\OneDrive\Documents\Desktop\AI-ASSIS> & c:/users/nalla/appdata/local/python/python313/python.exe c:/users/nalla/OneDrive/Documents/Desktop/AI-ASSIS/Assignment-3.py
Enter a number: 126
126 is Composite.
PS C:\Users\nalla\OneDrive\Documents\Desktop\AI-ASSIS>
```

Analysis:

The AI works only with instructions, no examples.

The code usually works but may miss some cases.

Testing with different numbers shows if it's correct.

The logic may be slower without optimization.

Extra checks improve accuracy.

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:

```
#Write a python code to check whether given input is even or odd number  
#e.g input=4 output=Even  
#e.g input=15 output=Odd  
# Take input from the user  
# Check and display whether the number is even or odd
```

Code and Output:

The screenshot shows a dark-themed instance of Visual Studio Code (VS Code) with several windows open. The main editor window displays a Python script named `Assignment-3.py`. The script contains a function `check_even_odd` that takes a number as input and returns "Even" if it's divisible by 2, and "Odd" otherwise. It then prompts the user for input, checks it, and prints the result. The terminal window at the bottom shows the script being run in a Python 3 environment, with the output indicating that 9 is odd and 98 is even. A floating panel titled "Build with Agent" is visible on the right, suggesting AI integration.

```
File Edit Selection View Go Run Terminal Help < > AI-ASSIS

simple.py Assignment-3.py
Assignment-3.py ...
simple.py
80 #Question-6
81 #Write a python code to check whether given input is even or odd number
82 #e.g input=4 output=Even
83 #e.g input=15 output=Odd
84 Tabnine | Edit | Test | Explain | Document
85 def check_even_odd(number):
86     if number % 2 == 0:
87         return "Even"
88     else:
89         return "Odd"
90 # Take input from the user
91 user_input = int(input("Enter a number: "))
92 # Check and display whether the number is even or odd
93 result = check_even_odd(user_input)
94 print(f"(user_input) is {result}.")
95

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS C:\Users\nalla\OneDrive\Documents\Desktop\AI-ASSIS> & c:/Users/nalla/AppData/Local/Programs/Python/Python310/python.exe c:/Users/nalla/OneDrive/Documents/Desktop/AI-ASSIS/Assignment-3.py
Enter a number: 9
9 is odd.
PS C:\Users\nalla\OneDrive\Documents\Desktop\AI-ASSIS> & c:/Users/nalla/AppData/Local/Programs/Python/Python310/python.exe c:/Users/nalla/OneDrive/Documents/Desktop/AI-ASSIS/Assignment-3.py
Enter a number: 98
98 is Even.
PS C:\Users\nalla\OneDrive\Documents\Desktop\AI-ASSIS>

Ln 94 Col 1 Spaces: 4 UFT-8 CRLF Python 3.13.7 ⚡ Port: 5500
Launchpad 29°C Sunny
CHAT RECENT SESSIONS
General greeting inquiry
Completed
Show More
Local • 1 wk
Build with Agent
AI responses may be inaccurate.
Generate Agent Instructions to onboard AI onto your codebase.

& + Assignment-3.py
Describe what to build next
Agent Auto
14:46
19-01-2026
```

Analysis:

Examples make the task easy to understand.

The AI gives clear even or odd results.

Input validation improves with examples.

Negative numbers are handled properly.

Wrong inputs are easier to detect.