

## Lab Assignment – 3.1

Hall Ticket No.: 2303A510D9

Name – O.Karthik Reddy

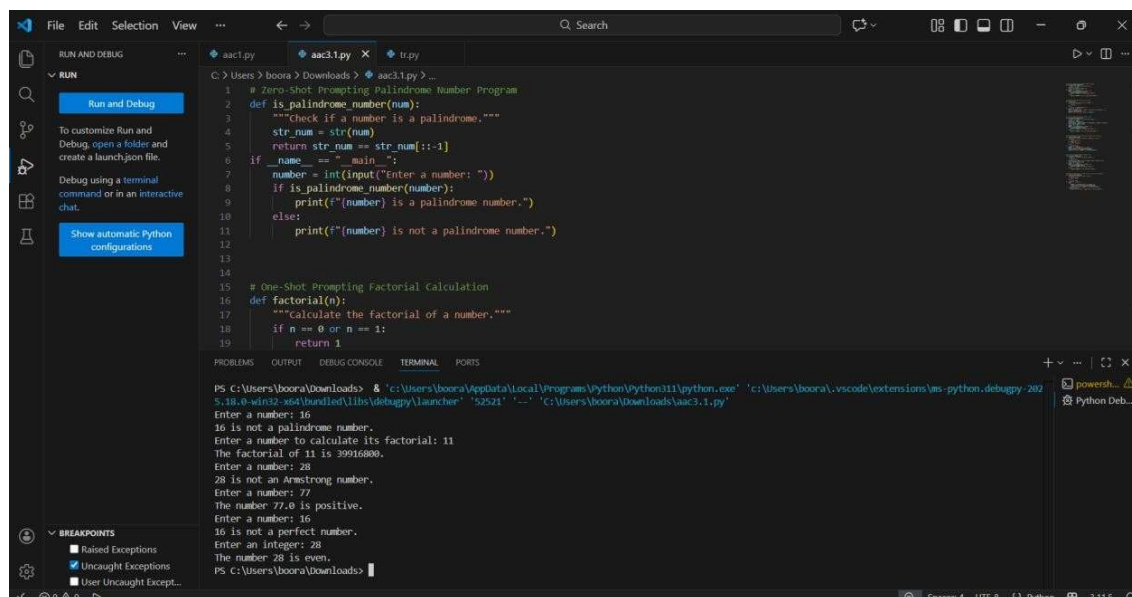
Batch – 29

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.

### CODE & OUTPUT



```
File Edit Selection View ... < -> Search
aaci1.py x aaci1.py x ti.py
C:\Users\boora> Downloads > aaci1.py > _
# Zero-shot Prompting Palindrome Number Program
1 def is_palindrome_number(num):
2     """Check if a number is a palindrome."""
3     str_num = str(num)
4     return str_num == str_num[::-1]
5
6 if __name__ == "__main__":
7     number = int(input("Enter a number: "))
8     if is_palindrome_number(number):
9         print(f"{number} is a palindrome number.")
10    else:
11        print(f"{number} is not a palindrome number.")
12
13
14
15 # One-Shot Prompting Factorial Calculation
16 def factorial(n):
17     """calculate the factorial of a number."""
18     if n == 0 or n == 1:
19         return 1
20
PS C:\Users\boora\Downloads> & 'c:\Users\boora\AppData\Local\Programs\Python\Python311\python.exe' 'c:\Users\boora\.vscode\extensions\ms-python.debugpy-202
5.18.0-win32-x64\bin\debugpy\launcher' '52521' '-' 'c:\Users\boora\Downloads\aac1.py'
Enter a number: 16
16 is not a palindrome number.
Enter a number to calculate its factorial: 11
The factorial of 11 is 39916800.
Enter a number: 28
28 is not an Armstrong number.
Enter a number: 77
The number 77.0 is positive.
Enter a number: 16
16 is not a perfect number.
Enter an integer: 28
The number 28 is even.
PS C:\Users\boora\Downloads>
```

Question 2: One-Shot Prompting (Factorial Calculation)

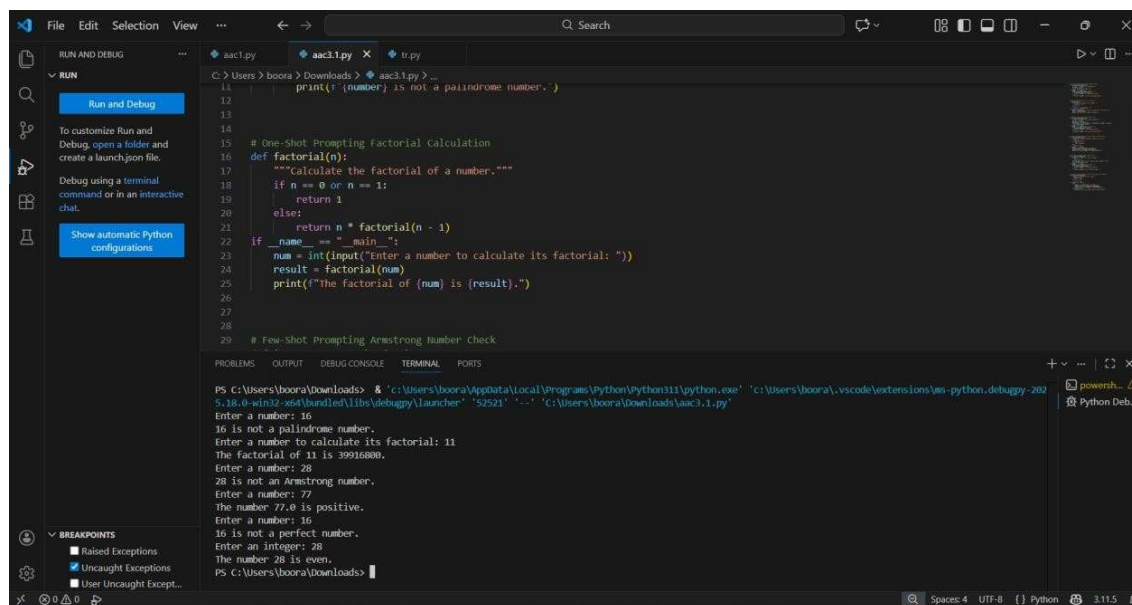
Write a one-shot prompt by providing one input-output example and ask the AI to generate a Python function to compute the factorial of a given number.

Example:

Input: 5 → Output: 120 Task:

- Compare the generated code with a zero-shot solution.
- Examine improvements in clarity and correctness.

## CODE & OUTPUT



The screenshot shows a VS Code editor with a Python file named `aac3.1.py`. The code defines a `factorial` function and includes a main block for user input and output. The terminal window at the bottom shows the execution of the script, displaying prompts for input and the resulting factorial values for several test cases.

```
File Edit Selection View ... Search
C:\Users\boora> Downloads > aac3.1.py > _
11 print(f'{number} is not a palindrome number.')
12
13
14
15 # One-Shot Prompting Factorial Calculation
16 def factorial(n):
17     """Calculate the factorial of a number."""
18     if n == 0 or n == 1:
19         return 1
20     else:
21         return n * factorial(n - 1)
22
23 if __name__ == "__main__":
24     num = int(input("Enter a number to calculate its factorial: "))
25     result = factorial(num)
26     print(f"the factorial of {num} is {result}.")
27
28
29 # Few-Shot Prompting Armstrong Number Check
30
```

Terminal Output:

```
PS C:\Users\boora\Downloads> & 'c:\Users\boora\AppData\Local\Programs\Python\Python111\python.exe' 'c:\Users\boora\.vscode\extensions\ms-python.debugpy-202
5.18.0-win32-x64\bin\debugpy_launcher' "52521" "... 'C:\Users\boora\Downloads\aac3.1.py'"
Enter a number: 16
16 is not a palindrome number.
Enter a number to calculate its factorial: 11
The factorial of 11 is 39916800.
Enter a number: 28
28 is not an Armstrong number.
Enter a number: 77
The number 77.0 is positive.
Enter a number: 16
16 is not a perfect number.
Enter an integer: 28
The number 28 is even.
PS C:\Users\boora\Downloads>
```

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.

## CODE & OUTPUT

The screenshot shows a VS Code editor with a Python file named `aac3.1.py`. The code defines a function `is_armstrong_number(num)` that checks if a number is an Armstrong number. The function converts the number to a string, calculates the sum of each digit raised to the power of the number of digits, and compares it to the original number. The main block prompts the user to enter a number and prints the result.

```

27
28
29 # Few-Shot Prompting Armstrong Number Check
30 def is_armstrong_number(num):
31     """Check if a number is an Armstrong number."""
32     num_str = str(num)
33     num_digits = len(num_str)
34     sum_of_powers = sum(int(digit) ** num_digits for digit in num_str)
35     return sum_of_powers == num
36
37 if __name__ == "__main__":
38     number = int(input("Enter a number: "))
39     if is_armstrong_number(number):
40         print(f"{number} is an Armstrong number.")
41     else:
42         print(f"{number} is not an Armstrong number.")
43
44

```

The terminal output shows the execution of the script. It prompts the user to enter a number and prints the result. The output includes several test cases, including numbers that are not Armstrong numbers, a number that is an Armstrong number (370), and a number that is not an Armstrong number (123).

```

PS C:\Users\boora\Downloads> & 'c:\Users\boora\AppData\Local\Programs\Python\Python111\python.exe' 'c:\Users\boora\.vscode\extensions\ms-python.debugpy-202
5.18.0-win32-x64\lib\debugpy\launcher' '52521' '-' 'c:\Users\boora\Downloads\aac3.1.py'
Enter a number: 16
16 is not a palindrome number.
Enter a number to calculate its factorial: 11
The factorial of 11 is 39916800.
Enter a number: 28
28 is not an Armstrong number.
Enter a number: 77
The number 77.0 is positive.
Enter a number: 16
16 is not a perfect number.
Enter an integer: 28
The number 28 is even.
PS C:\Users\boora\Downloads>

```

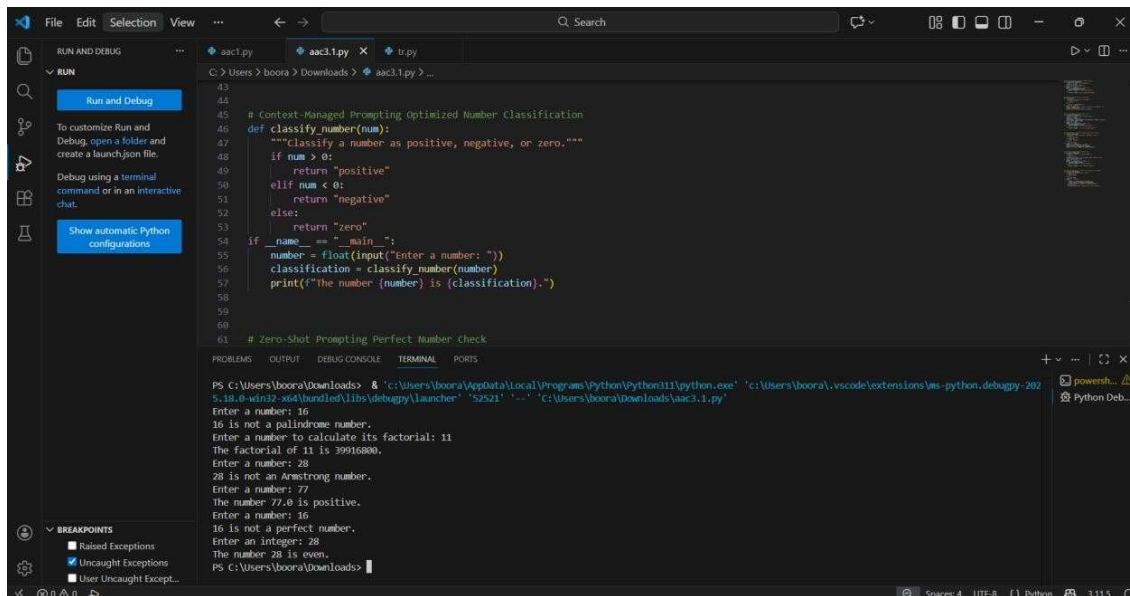
## Question 4: Context-Managed Prompting (Optimized Number Classification)

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.

## CODE & OUTPUT



The screenshot shows a VS Code editor with a Python file named `aac3.1.py`. The code defines a `classify_number` function that takes a number and returns a string classification: "positive", "negative", or "zero". The function is then used in a `__main__` block to prompt the user for a number and print its classification. Below the code, the terminal output shows the program's execution with various inputs and their corresponding classifications.

```
44
45 # Context-Managed Prompting Optimized Number Classification
46 def classify_number(num):
47     """Classify a number as positive, negative, or zero."""
48     if num > 0:
49         return "positive"
50     elif num < 0:
51         return "negative"
52     else:
53         return "zero"
54 if __name__ == "__main__":
55     number = float(input("Enter a number: "))
56     classification = classify_number(number)
57     print(f"The number {number} is {classification}.")
58
59
60
61 # Zero-Shot Prompting Perfect Number Check
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

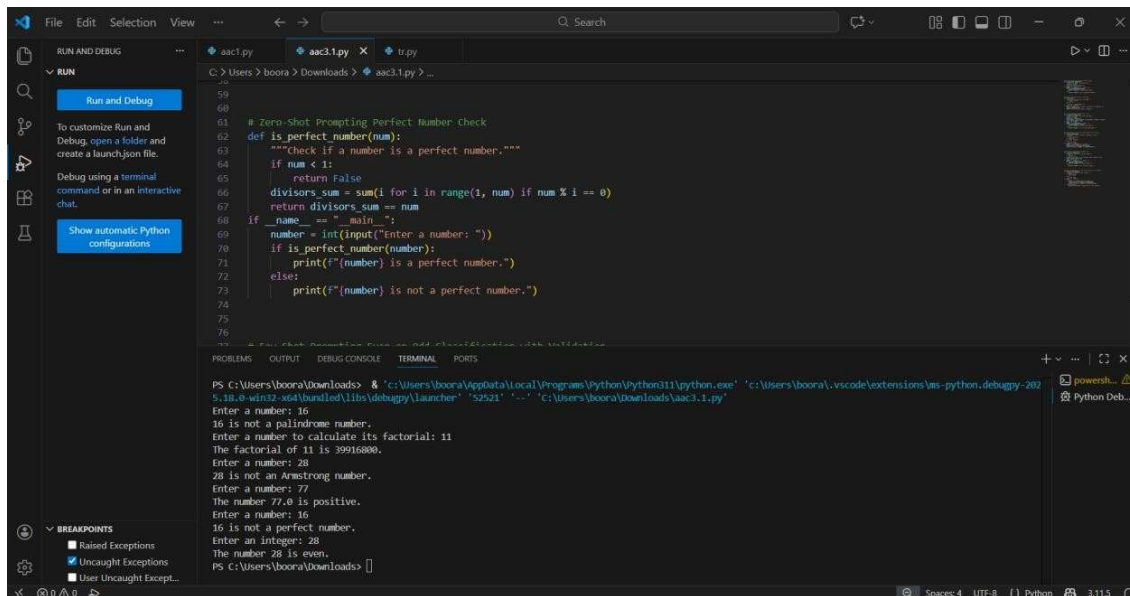
```
PS C:\Users\boora\Downloads> & 'c:\Users\boora\AppData\Local\Programs\Python\Python111\python.exe' 'c:\Users\boora\.vscode\extensions\ms-python.debugpy-202
5.18.0-win32-x64\libs\debugpy\launcher' '52521' '-' 'c:\Users\boora\Downloads\aac3.1.py'
Enter a number: 16
16 is not a palindrome number.
Enter a number to calculate its factorial: 11
The factorial of 11 is 39916800.
Enter a number: 28
28 is not an Armstrong number.
Enter a number: 77
The number 77.0 is positive.
Enter a number: 16
16 is not a perfect number.
Enter an integer: 28
The number 28 is even.
PS C:\Users\boora\Downloads>
```

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.

## CODE & OUTPUT



The screenshot shows a VS Code editor with a Python file named `aac3.1.py`. The code is a Zero-Shot Prompting script for checking perfect numbers. The terminal output shows the program's execution with various inputs and their corresponding outputs.

```
# Zero-Shot Prompting: Perfect Number Check
def is_perfect_number(num):
    """Check if a number is a perfect number."""
    if num < 1:
        return False
    divisors_sum = sum(1 for i in range(1, num) if num % i == 0)
    return divisors_sum == num
if __name__ == "__main__":
    number = int(input("Enter a number: "))
    if is_perfect_number(number):
        print(f"{number} is a perfect number.")
    else:
        print(f"{number} is not a perfect number.")
```

Terminal Output:

```
PS C:\Users\boora\Downloads> & 'c:\Users\boora\AppData\Local\Programs\Python\Python111\python.exe' 'c:\Users\boora\.vscode\extensions\ms-python.debugpy-202
5.18.0-win32-x64\lib\debugpy\launcher' '52521' '-' 'c:\Users\boora\Downloads\aac3.1.py'
Enter a number: 16
16 is not a palindrome number.
Enter a number to calculate its factorial: 11
The factorial of 11 is 39916800.
Enter a number: 28
28 is not an Armstrong number.
Enter a number: 77
The number 77.0 is positive.
Enter a number: 16
16 is not a perfect number.
Enter an integer: 28
The number 28 is even.
PS C:\Users\boora\Downloads>
```

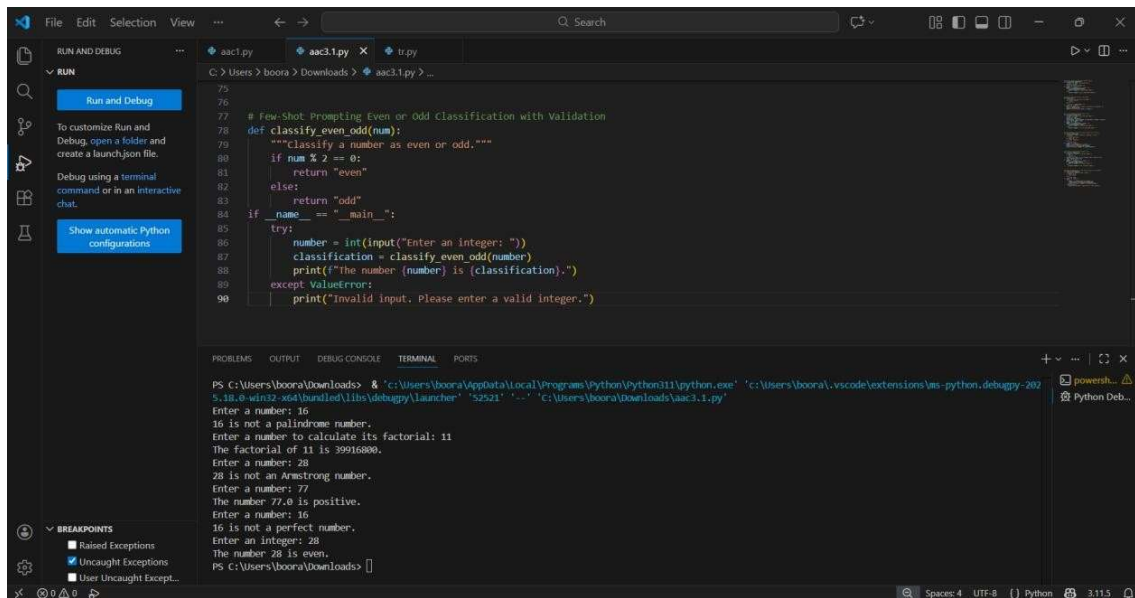
Question 6: Few-Shot Prompting (Even or Odd Classification with Validation)

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.

## CODE & OUTPUT



The image shows a Visual Studio Code editor window with a Python file named `aac3.1.py` open. The file contains a function `classify_even_odd(num)` that checks if a number is even or odd, and a `__main__` block that prompts the user for an integer and prints the classification. The terminal at the bottom shows the execution of the script, with the user entering several numbers and seeing the corresponding output.

```
75
76
77 # Few Shot Prompting Even or Odd Classification with Validation
78 def classify_even_odd(num):
79     """classify a number as even or odd."""
80     if num % 2 == 0:
81         return "even"
82     else:
83         return "odd"
84 if __name__ == "__main__":
85     try:
86         number = int(input("Enter an integer: "))
87         classification = classify_even_odd(number)
88         print(f"The number {number} is {classification}.")
89     except ValueError:
90         print("Invalid input. Please enter a valid integer.")
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS C:\Users\boora\Downloads> & 'c:\Users\boora\AppData\Local\Programs\Python\Python111\python.exe' 'c:\Users\boora\.vscode\extensions\ms-python.debugpy-202
5.18.0-win32-x64\lib\debugpy\launcher' '52521' '-' 'c:\Users\boora\Downloads\aac3.1.py'
Enter a number: 16
16 is not a palindrome number.
Enter a number to calculate its factorial: 11
The factorial of 11 is 39916800.
Enter a number: 28
28 is not an Armstrong number.
Enter a number: 77
The number 77.0 is positive.
Enter a number: 16
16 is not a perfect number.
Enter an integer: 28
The number 28 is even.
PS C:\Users\boora\Downloads>
```

BREAKPOINTS

- ☐ Raised Exceptions
- ☒ Uncaught Exceptions
- ☐ User Uncaught Except...