ASSIGNMENT-3.1

NAME: G. SANJAY

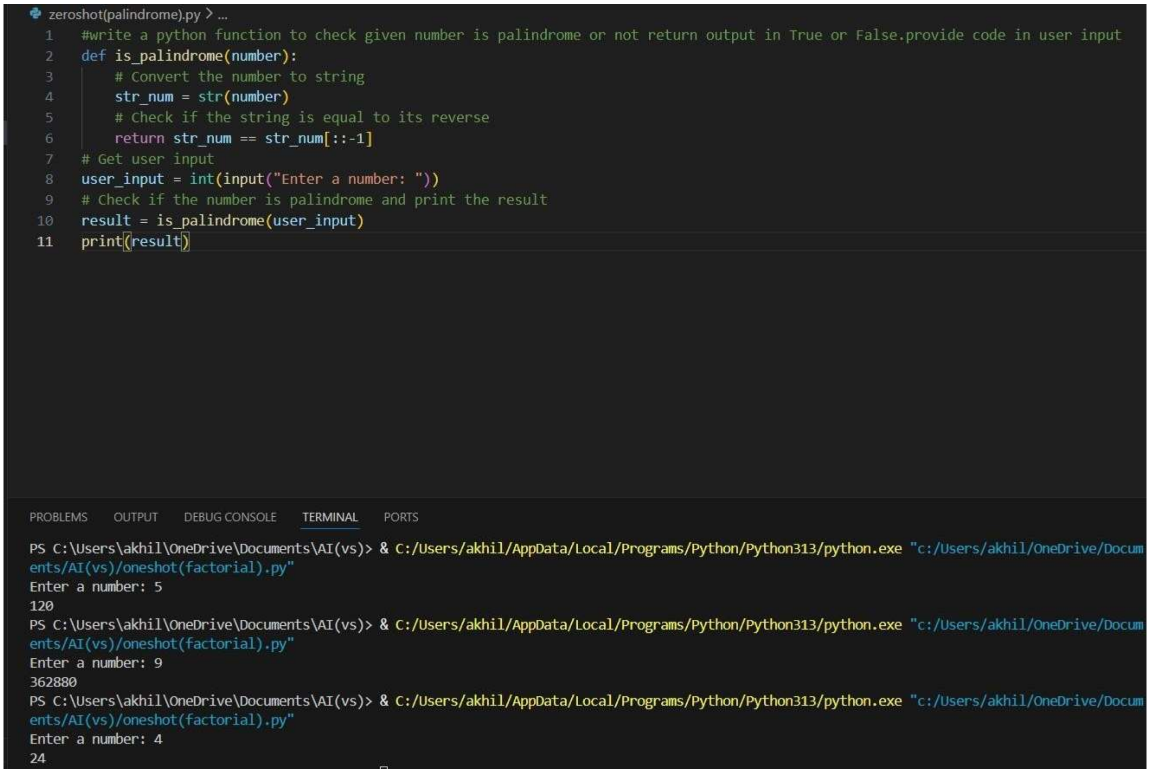
BATCH : 13

ROLL NO: 2303A510B9

Task1:Zero-ShotPrompting(Palindrome Number Program)

Prompt: write a python function to check given number is palindrome or not return output int True or False. Provide code in user input.

Code & output:



```python
#write a python function to check given number is palindrome or not return output in True or False.provide code in user input
def is_palindrome(number):
    # Convert the number to string
    str_num = str(number)
    # Check if the string is equal to its reverse
    return str_num == str_num[::-1]
# Get user input
user_input = int(input("Enter a number: "))
# Check if the number is palindrome and print the result
result = is_palindrome(user_input)
print(result)
```

```
PS C:\Users\akhil\OneDrive\Documents\AI(vs)> & C:/Users/akhil/AppData/Local/Programs/Python/Python313/python.exe "c:/Users/akhil/OneDrive/Docum
ents/AI(vs)/oneshot(factorial).py"
Enter a number: 5
120
PS C:\Users\akhil\OneDrive\Documents\AI(vs)> & C:/Users/akhil/AppData/Local/Programs/Python/Python313/python.exe "c:/Users/akhil/OneDrive/Docum
ents/AI(vs)/oneshot(factorial).py"
Enter a number: 9
362880
PS C:\Users\akhil\OneDrive\Documents\AI(vs)> & C:/Users/akhil/AppData/Local/Programs/Python/Python313/python.exe "c:/Users/akhil/OneDrive/Docum
ents/AI(vs)/oneshot(factorial).py"
Enter a number: 4
24
```

Justification:

- Correct for positive numbers
- Negative numbers not handled clearly

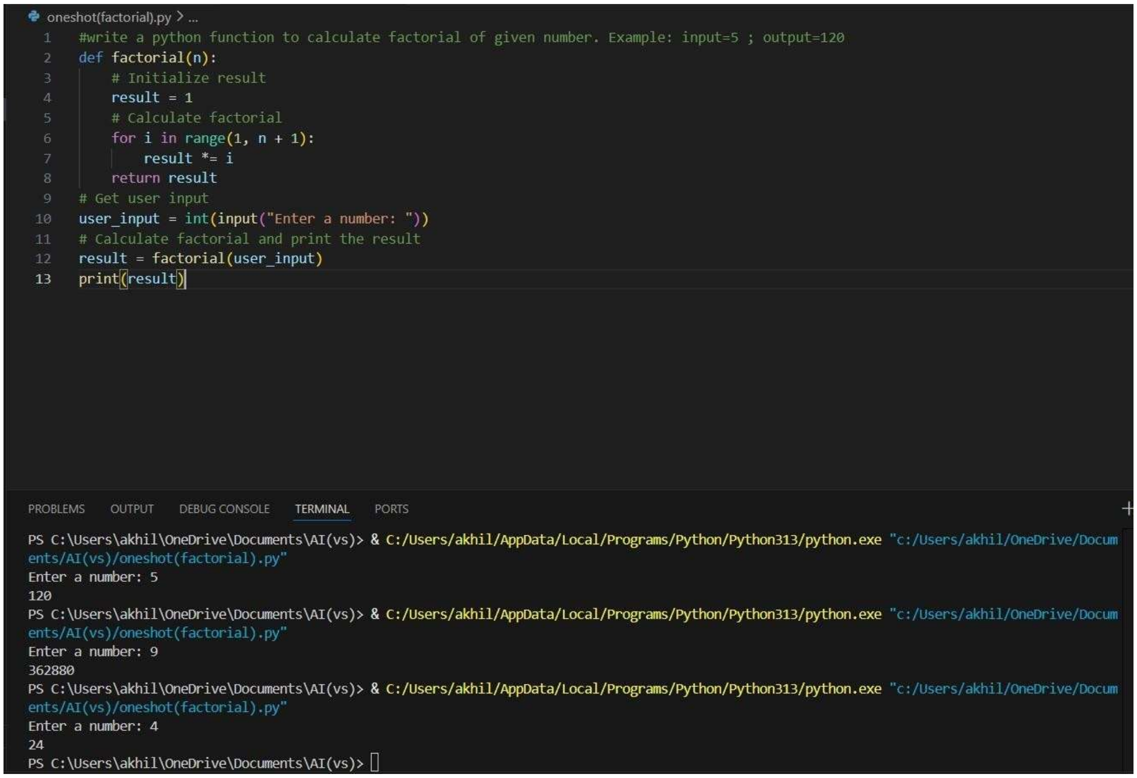
 No input validation for non-integers

Task2:One-ShotPrompting(Factorial Calculation)

Prompt:write a python function to calculate factorial of given number.

Example: input=5; output=120

Code with output:



```python
#write a python function to calculate factorial of given number. Example: input=5 ; output=120
def factorial(n):
    # Initialize result
    result = 1
    # Calculate factorial
    for i in range(1, n + 1):
        result *= i
    return result
# Get user input
user_input = int(input("Enter a number: "))
# Calculate factorial and print the result
result = factorial(user_input)
print(result)
```

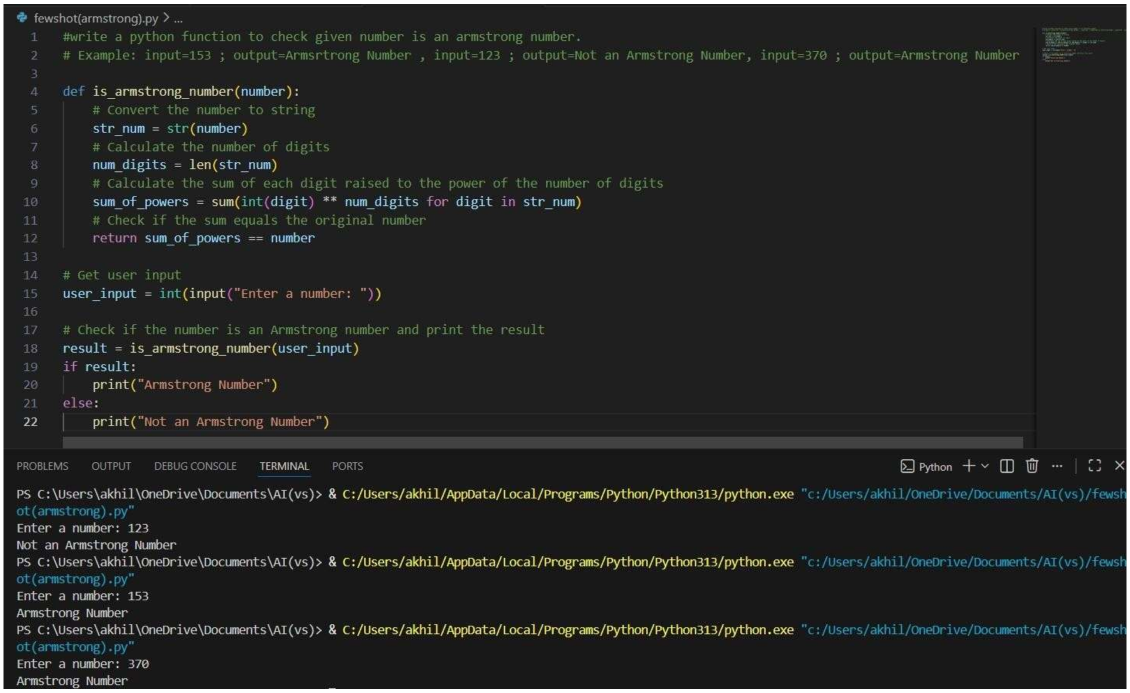
```
PS C:\Users\akhil\OneDrive\Documents\AI(vs)> & C:/Users/akhil/AppData/Local/Programs/Python/Python313/python.exe "c:/Users/akhil/OneDrive/Docum
ents/AI(vs)/oneshot(factorial).py"
Enter a number: 5
120
PS C:\Users\akhil\OneDrive\Documents\AI(vs)> & C:/Users/akhil/AppData/Local/Programs/Python/Python313/python.exe "c:/Users/akhil/OneDrive/Docum
ents/AI(vs)/oneshot(factorial).py"
Enter a number: 9
362880
PS C:\Users\akhil\OneDrive\Documents\AI(vs)> & C:/Users/akhil/AppData/Local/Programs/Python/Python313/python.exe "c:/Users/akhil/OneDrive/Docum
ents/AI(vs)/oneshot(factorial).py"
Enter a number: 4
24
PS C:\Users\akhil\OneDrive\Documents\AI(vs)> 
```

Justification:

□ One-shotgivesclearerlogic

□ Morereadable andreliable

Task3:Few-ShotPrompting(Armstrong NumberCheck) Code with output:



```python
#write a python function to check given number is an armstrong number.
# Example: input=153 ; output=Armsrtrong Number , input=123 ; output=Not an Armstrong Number, input=370 ; output=Armstrong Number

def is_armstrong_number(number):
    # Convert the number to string
    str_num = str(number)
    # Calculate the number of digits
    num_digits = len(str_num)
    # Calculate the sum of each digit raised to the power of the number of digits
    sum_of_powers = sum(int(digit) ** num_digits for digit in str_num)
    # Check if the sum equals the original number
    return sum_of_powers == number

# Get user input
user_input = int(input("Enter a number: "))

# Check if the number is an Armstrong number and print the result
result = is_armstrong_number(user_input)
if result:
    print("Armstrong Number")
else:
    print("Not an Armstrong Number")
```

```
PS C:\Users\akhil\OneDrive\Documents\AI(vs)> & C:/Users/akhil/AppData/Local/Programs/Python/Python313/python.exe "c:/Users/akhil/OneDrive/Documents/AI(vs)/fewsh
ot(armstrong).py"
Enter a number: 123
Not an Armstrong Number
PS C:\Users\akhil\OneDrive\Documents\AI(vs)> & C:/Users/akhil/AppData/Local/Programs/Python/Python313/python.exe "c:/Users/akhil/OneDrive/Documents/AI(vs)/fewsh
ot(armstrong).py"
Enter a number: 153
Armstrong Number
PS C:\Users\akhil\OneDrive\Documents\AI(vs)> & C:/Users/akhil/AppData/Local/Programs/Python/Python313/python.exe "c:/Users/akhil/OneDrive/Documents/AI(vs)/fewsh
ot(armstrong).py"
Enter a number: 370
Armstrong Number
```

Justification:

- Few-shot improves accuracy
- Code structure matches examples

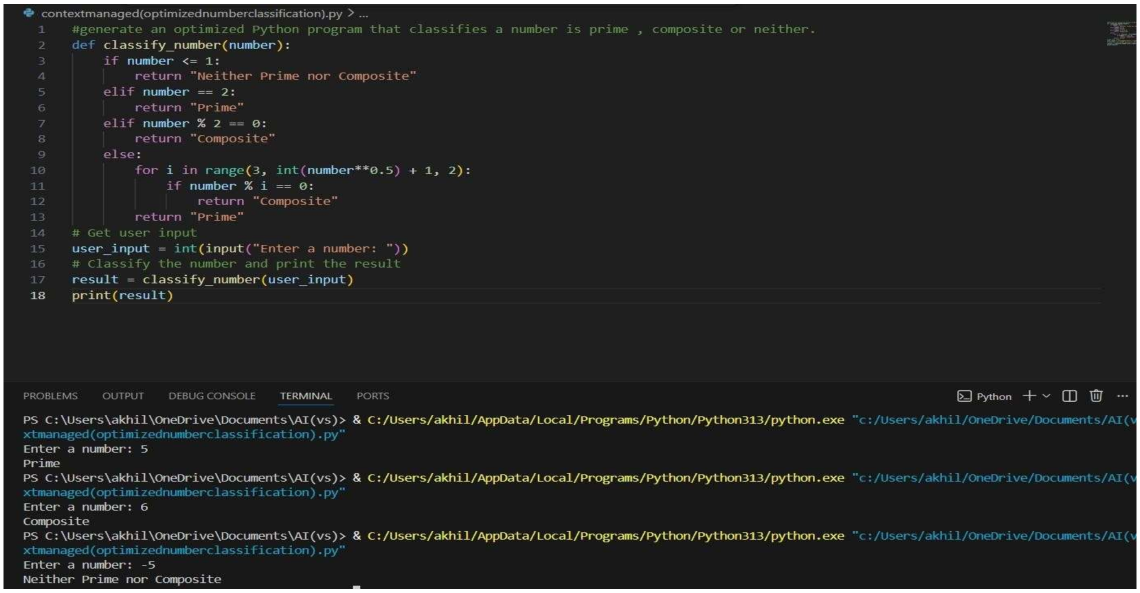
 Still lacks input validation for negatives

Task4:Context-ManagedPrompting(Optimized Number

Classification)

Prompt:generateanoptimizedpythonprogramthatclassifiesanumberisprime, composite or neither

Codewith output:



```python
#generate an optimized Python program that classifies a number is prime , composite or neither.
def classify_number(number):
    if number <= 1:
        return "Neither Prime nor Composite"
    elif number == 2:
        return "Prime"
    elif number % 2 == 0:
        return "Composite"
    else:
        for i in range(3, int(number**0.5) + 1, 2):
            if number % i == 0:
                return "Composite"
        return "Prime"
# Get user input
user_input = int(input("Enter a number: "))
# Classify the number and print the result
result = classify_number(user_input)
print(result)
```

```
PS C:\Users\akhil\OneDrive\Documents\AI(vs)> & C:/Users/akhil/AppData/Local/Programs/Python/Python313/python.exe "c:/Users/akhil/OneDrive/Documents/AI(v
xtmanaged(optimizednumberclassification).py"
Enter a number: 5
Prime
PS C:\Users\akhil\OneDrive\Documents\AI(vs)> & C:/Users/akhil/AppData/Local/Programs/Python/Python313/python.exe "c:/Users/akhil/OneDrive/Documents/AI(v
xtmanaged(optimizednumberclassification).py"
Enter a number: 6
Composite
PS C:\Users\akhil\OneDrive\Documents\AI(vs)> & C:/Users/akhil/AppData/Local/Programs/Python/Python313/python.exe "c:/Users/akhil/OneDrive/Documents/AI(v
xtmanaged(optimizednumberclassification).py"
Enter a number: -5
Neither Prime nor Composite
```

Justification:

- Best performance among all prompts

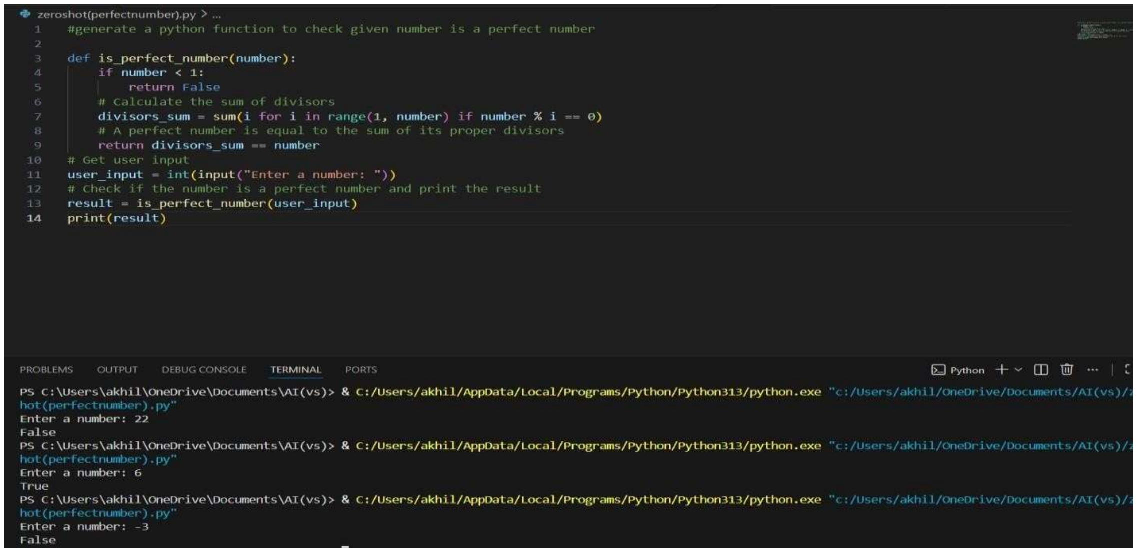
- Clear validation and optimized logic

 Context improves efficiency and correctness

Task5:Zero-ShotPrompting(Perfect Number Check)

Prompt: generate a python function to check given number is a perfect number.

Code with output:



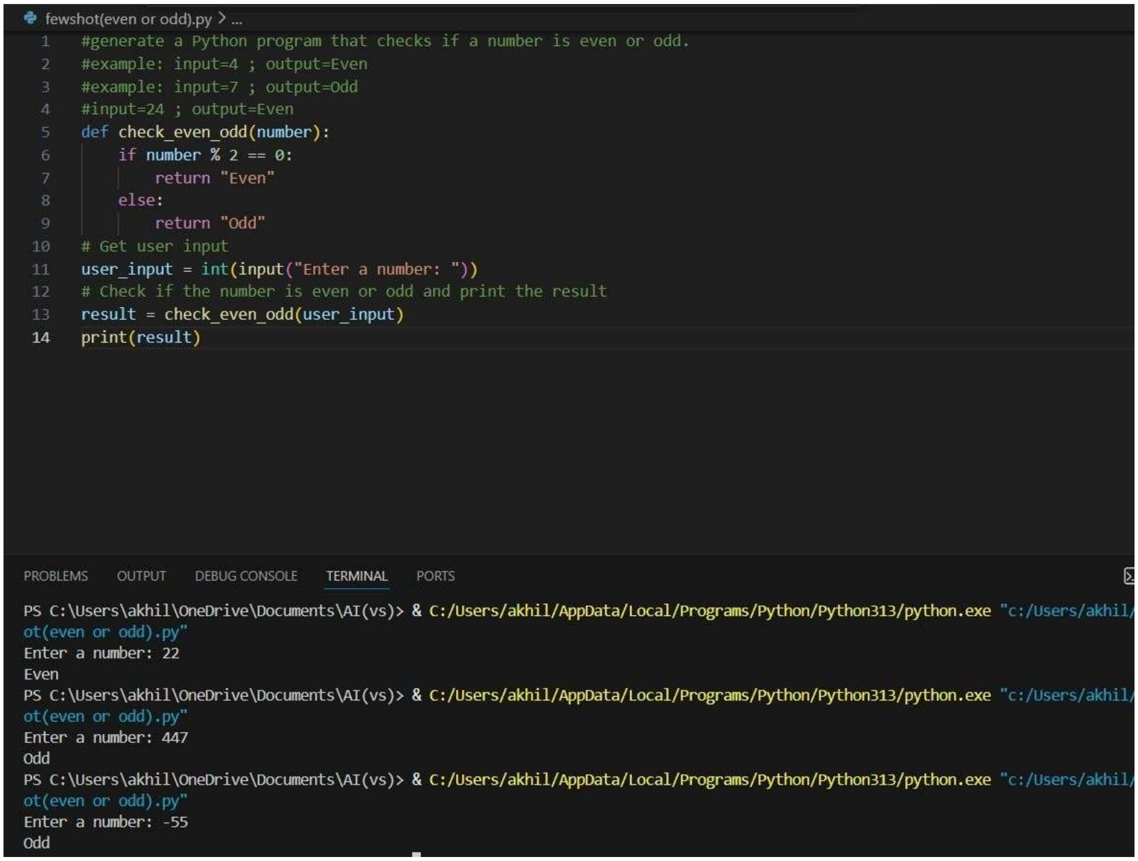
```python
#generate a python function to check given number is a perfect number

def is_perfect_number(number):
    if number < 1:
        return False
    # Calculate the sum of divisors
    divisors_sum = sum(i for i in range(1, number) if number % i == 0)
    # A perfect number is equal to the sum of its proper divisors
    return divisors_sum == number
# Get user input
user_input = int(input("Enter a number: "))
# Check if the number is a perfect number and print the result
result = is_perfect_number(user_input)
print(result)
```

```
PS C:\Users\akhil\OneDrive\Documents\AI(vs)> & C:/Users/akhil/AppData/Local/Programs/Python/Python313/python.exe "c:/Users/akhil/OneDrive/Documents/AI(vs)/z
hot(perfectnumber).py"
Enter a number: 22
False
PS C:\Users\akhil\OneDrive\Documents\AI(vs)> & C:/Users/akhil/AppData/Local/Programs/Python/Python313/python.exe "c:/Users/akhil/OneDrive/Documents/AI(vs)/z
hot(perfectnumber).py"
Enter a number: 6
True
PS C:\Users\akhil\OneDrive\Documents\AI(vs)> & C:/Users/akhil/AppData/Local/Programs/Python/Python313/python.exe "c:/Users/akhil/OneDrive/Documents/AI(vs)/z
hot(perfectnumber).py"
Enter a number: -3
False
```

Task6:Few-ShotPrompting(Even or Odd Classification with Validation)

Code without put:



```python
#generate a Python program that checks if a number is even or odd.
#example: input=4 ; output=Even
#example: input=7 ; output=Odd
#input=24 ; output=Even
def check_even_odd(number):
    if number % 2 == 0:
        return "Even"
    else:
        return "Odd"
# Get user input
user_input = int(input("Enter a number: "))
# Check if the number is even or odd and print the result
result = check_even_odd(user_input)
print(result)
```

```
PS C:\Users\akhil\OneDrive\Documents\AI(vs)> & C:/Users/akhil/AppData/Local/Programs/Python/Python313/python.exe "c:/Users/akhil/
ot(even or odd).py"
Enter a number: 22
Even
PS C:\Users\akhil\OneDrive\Documents\AI(vs)> & C:/Users/akhil/AppData/Local/Programs/Python/Python313/python.exe "c:/Users/akhil/
ot(even or odd).py"
Enter a number: 447
Odd
PS C:\Users\akhil\OneDrive\Documents\AI(vs)> & C:/Users/akhil/AppData/Local/Programs/Python/Python313/python.exe "c:/Users/akhil/
ot(even or odd).py"
Enter a number: -55
Odd
```