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

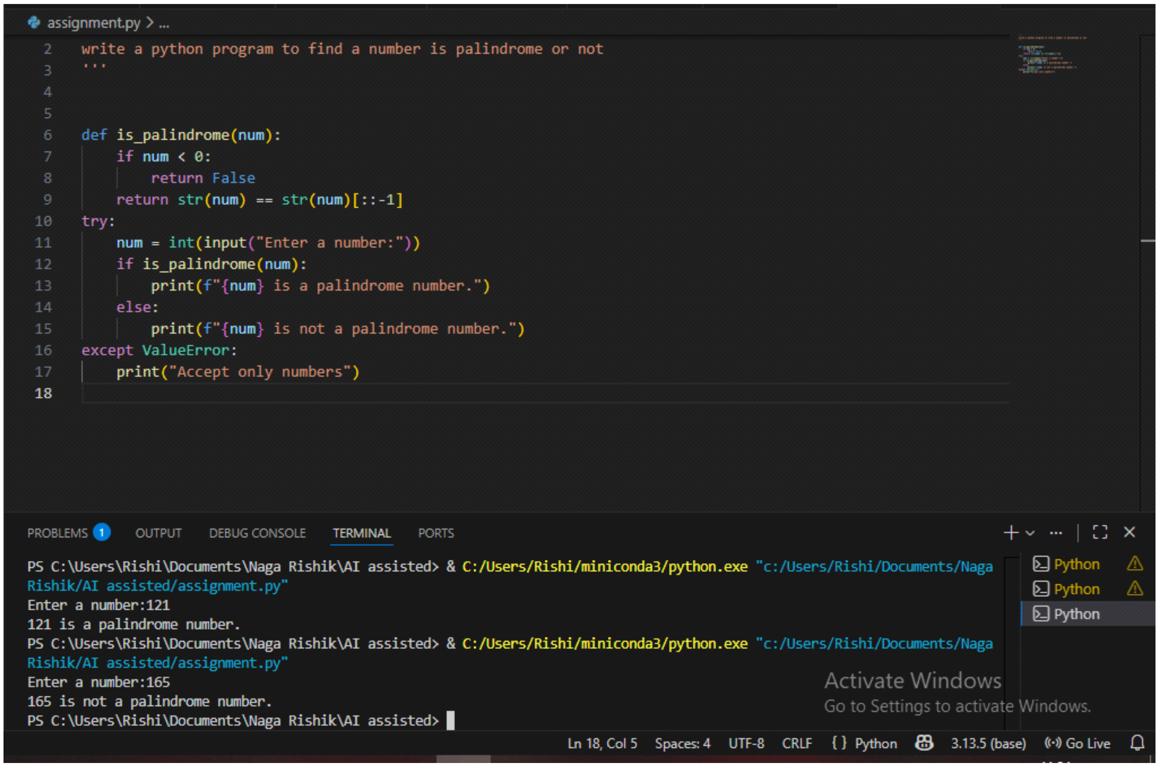
a Python function that checks whether a given number is a palindrome.

Task:

Record the AI-generated code.

Test the code with multiple inputs.

Identify any logical errors or missing edge-case handling.



```python
write a python program to find a number is palindrome or not
'''


def is_palindrome(num):
    if num < 0:
        return False
    return str(num) == str(num)[::-1]
try:
    num = int(input("Enter a number:"))
    if is_palindrome(num):
        print(f"{num} is a palindrome number.")
    else:
        print(f"{num} is not a palindrome number.")
except ValueError:
    print("Accept only numbers")

```

```
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>
```

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

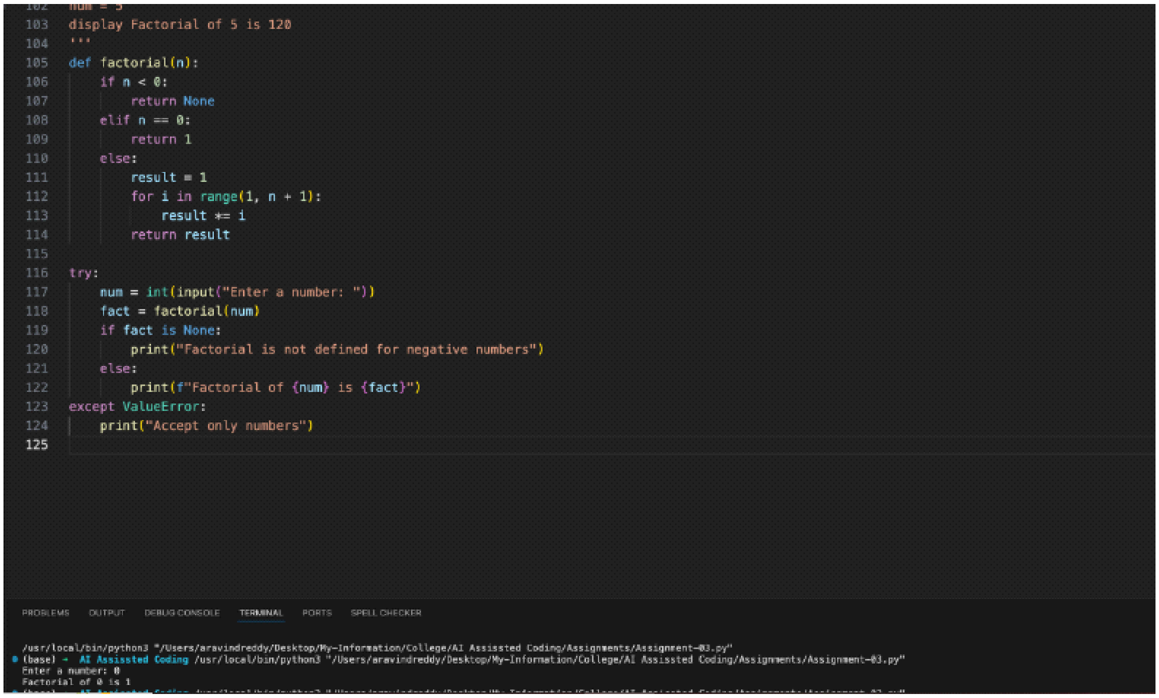
given number.

Example:

Input: 5 → Output: 120

Task:

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



```python
num = 5
display Factorial of 5 is 120
'''
def factorial(n):
    if n < 0:
        return None
    elif n == 0:
        return 1
    else:
        result = 1
        for i in range(1, n + 1):
            result *= i
        return result

try:
    num = int(input("Enter a number: "))
    fact = factorial(num)
    if fact is None:
        print("Factorial is not defined for negative numbers")
    else:
        print(f"Factorial of {num} is {fact}")
except ValueError:
    print("Accept only numbers")
```

```
/usr/local/bin/python3 "/Users/aravindreddy/Desktop/My-Information/College/AI Assissted Coding/Assignments/Assignment-03.py"
● (base) → AI Assissted Coding /usr/local/bin/python3 "/Users/aravindreddy/Desktop/My-Information/College/AI Assissted Coding/Assignments/Assignment-03.py"
Enter a number: 0
Factorial of 0 is 1
```

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:

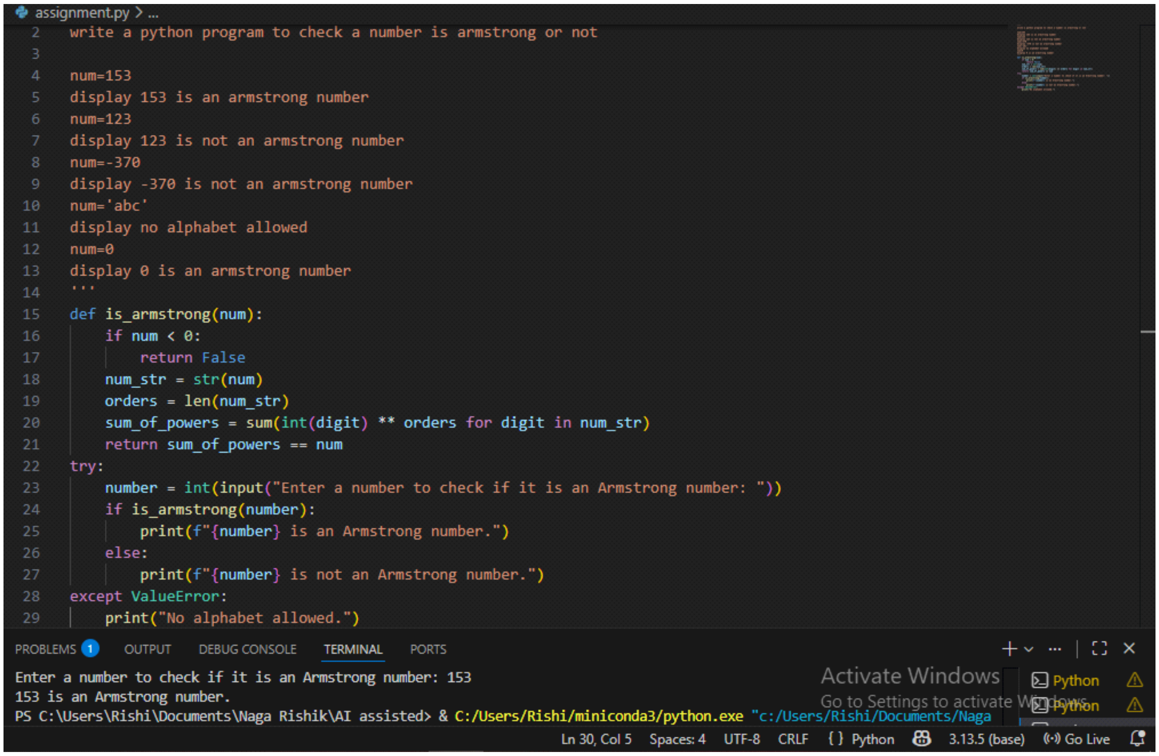
- Input: 153 → Output: Armstrong Number
- Input: 370 → Output: Armstrong Number
- Input: 123 → Output: Not an Armstrong Number

Task:

- Analyze how multiple examples influence code structure and

accuracy.

- Test the function with boundary values and invalid inputs.



```python
write a python program to check a number is armstrong or not

num=153
display 153 is an armstrong number
num=123
display 123 is not an armstrong number
num=-370
display -370 is not an armstrong number
num='abc'
display no alphabet allowed
num=0
display 0 is an armstrong number
'''
def is_armstrong(num):
    if num < 0:
        return False
    num_str = str(num)
    orders = len(num_str)
    sum_of_powers = sum(int(digit) ** orders for digit in num_str)
    return sum_of_powers == num
try:
    number = int(input("Enter a number to check if it is an Armstrong number: "))
    if is_armstrong(number):
        print(f"{number} is an Armstrong number.")
    else:
        print(f"{number} is not an Armstrong number.")
except ValueError:
    print("No alphabet allowed.")
```

```
Enter a number to check if it is an Armstrong number: 153
153 is an Armstrong number.
PS C:\Users\Rishi\Documents\Naga Rishik\AI assisted> & C:/Users/Rishi/miniconda3/python.exe "c:/Users/Rishi/Documents/Naga
```

(Optional Extension)

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

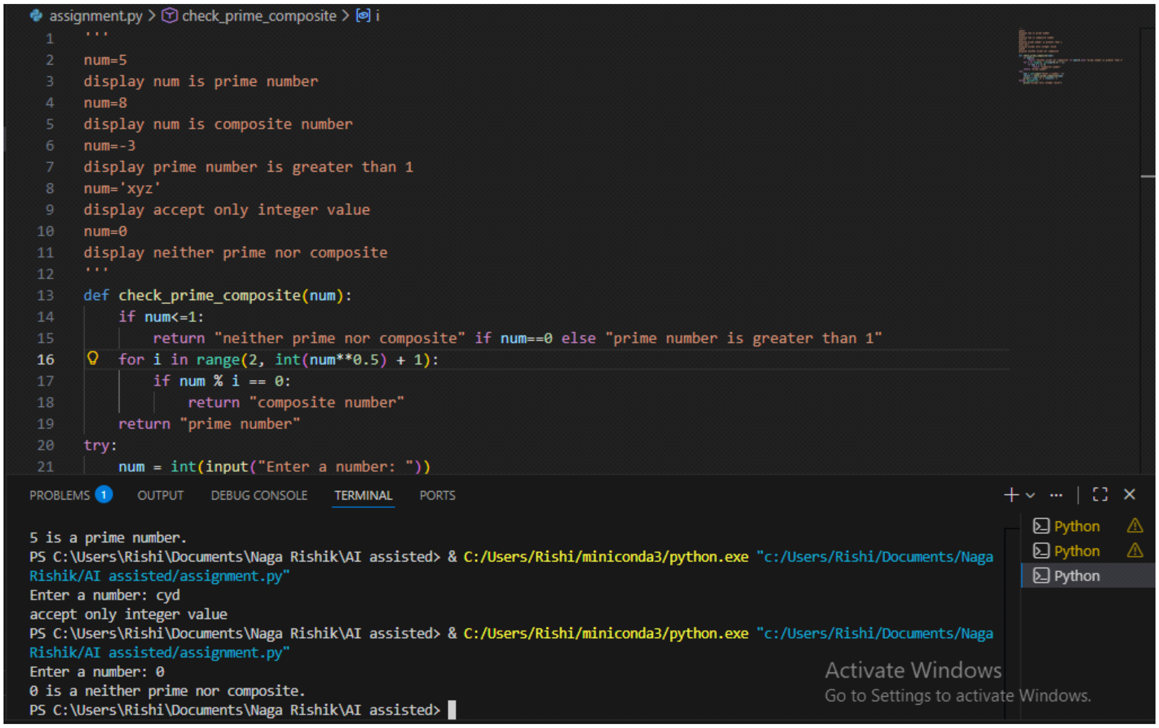
Design a context-managed prompt with clear instructions and

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.


assignment.py > check_prime_composite > i
PROBLEMS 1
OUTPUT
DEBUG CONSOLE
TERMINAL
PORTS
Python
Python
Python
Activate Windows
Go to Settings to activate Windows.

```
'''
num=5
display num is prime number
num=8
display num is composite number
num=-3
display prime number is greater than 1
num='xyz'
display accept only integer value
num=0
display neither prime nor composite
'''
def check_prime_composite(num):
    if num<=1:
        return "neither prime nor composite" if num==0 else "prime number is greater than 1"
    for i in range(2, int(num**0.5) + 1):
        if num % i == 0:
            return "composite number"
    return "prime number"
try:
    num = int(input("Enter a number: "))
```

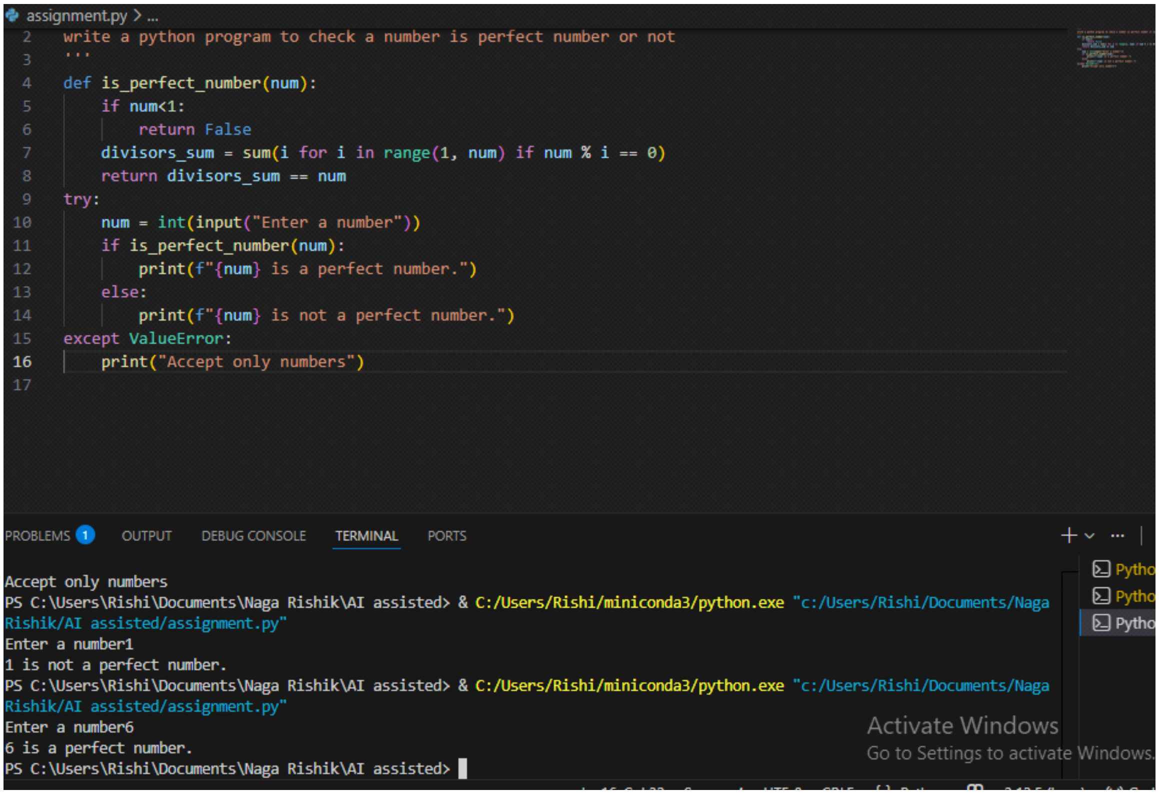
```
5 is a prime number.
PS C:\Users\Rishi\Documents\Naga Rishik\AI assisted> & C:/Users/Rishi/miniconda3/python.exe "c:/Users/Rishi/Documents/Naga
Rishik/AI assisted/assignment.py"
Enter a number: cyd
accept only integer value
PS C:\Users\Rishi\Documents\Naga Rishik\AI assisted> & C:/Users/Rishi/miniconda3/python.exe "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



```python
write a python program to check a number is perfect number or not
'''
def is_perfect_number(num):
    if num<1:
        return False
    divisors_sum = sum(i for i in range(1, num) if num % i == 0)
    return divisors_sum == num
try:
    num = int(input("Enter a number"))
    if is_perfect_number(num):
        print(f"{num} is a perfect number.")
    else:
        print(f"{num} is not a perfect number.")
except ValueError:
    print("Accept only numbers")

```

```
Accept only numbers
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 number1
1 is not a perfect 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 number6
6 is a perfect number.
PS C:\Users\Rishi\Documents\Naga Rishik\AI assisted> 
```

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:

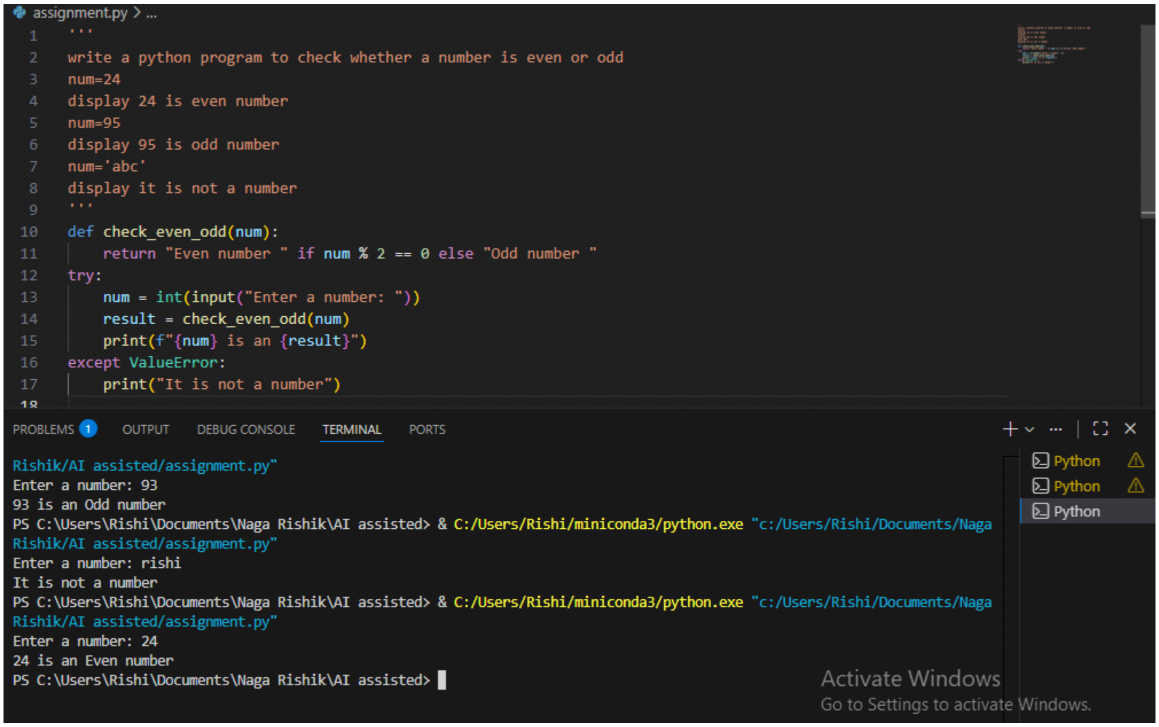
- Input: 8 → Output: Even
- Input: 15 → Output: Odd
- Input: 0 → Output: Even

Task:

- Analyze how examples improve input handling and output

clarity.

- Test the program with negative numbers and non-integer inputs



```python
'''
write a python program to check whether a number is even or odd
num=24
display 24 is even number
num=95
display 95 is odd number
num='abc'
display it is not a number
'''
def check_even_odd(num):
    return "Even number " if num % 2 == 0 else "Odd number "
try:
    num = int(input("Enter a number: "))
    result = check_even_odd(num)
    print(f"{num} is an {result}")
except ValueError:
    print("It is not a number")
```

```
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>
```