NAME:B.VYSHNAVI COURSE NAME : AIAC

2303A51884 WEEK - 3

BATCH-02

Question 1: Zero-Shot Prompting (Palindrome Number Program)

Write a zero-shot prompt (without providing any examples) to generate

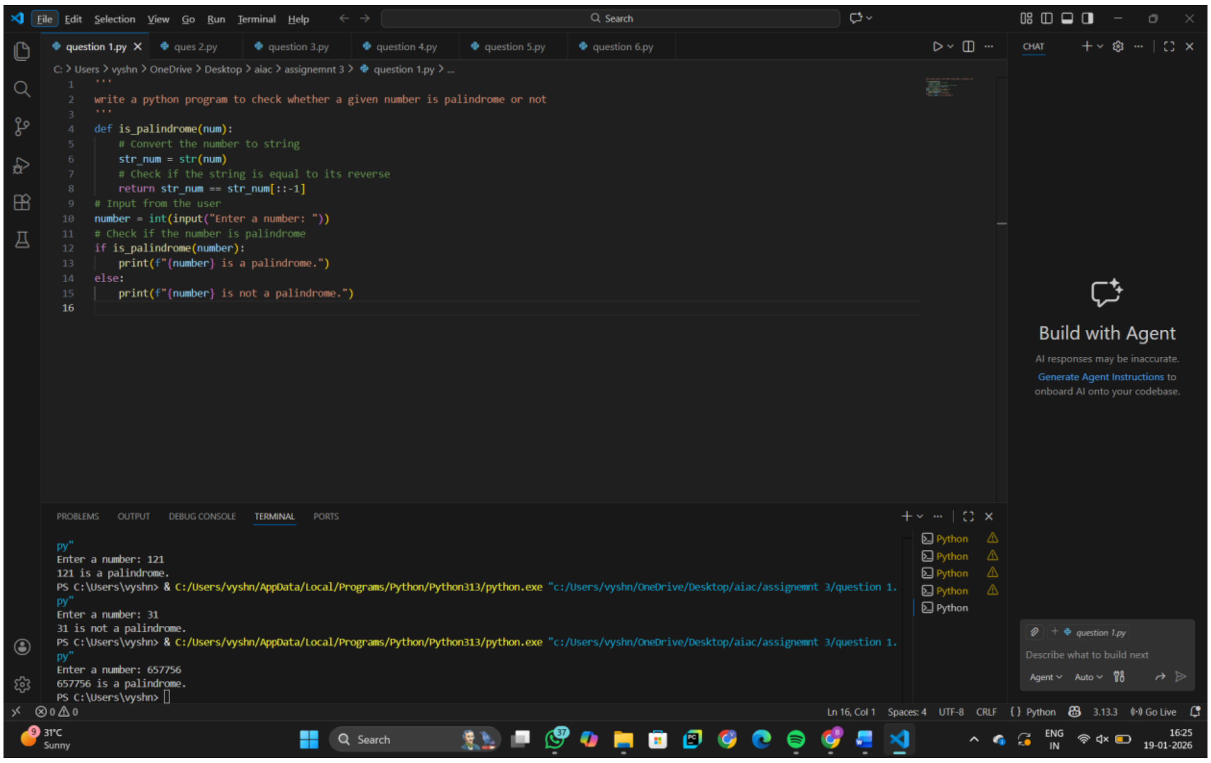
a Python function that checks whether a given number is a palindrome.

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 check whether a given number is palindrome or not
'''
def is_palindrome(num):
    # Convert the number to string
    str_num = str(num)
    # Check if the string is equal to its reverse
    return str_num == str_num[::-1]
# Input from the user
number = int(input("Enter a number: "))
# Check if the number is palindrome
if is_palindrome(number):
    print(f"{number} is a palindrome.")
else:
    print(f"{number} is not a palindrome.")
```

```
py"
Enter a number: 121
121 is a palindrome.
PS C:\Users\vyshn> & C:/Users/vyshn/AppData/Local/Programs/Python/Python313/python.exe "c:/Users/vyshn/OneDrive/Desktop/aiac/assignemnt 3/question 1.
py"
Enter a number: 31
31 is not a palindrome.
PS C:\Users\vyshn> & C:/Users/vyshn/AppData/Local/Programs/Python/Python313/python.exe "c:/Users/vyshn/OneDrive/Desktop/aiac/assignemnt 3/question 1.
py"
Enter a number: 657756
657756 is a palindrome.
PS C:\Users\vyshn> 
```

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.

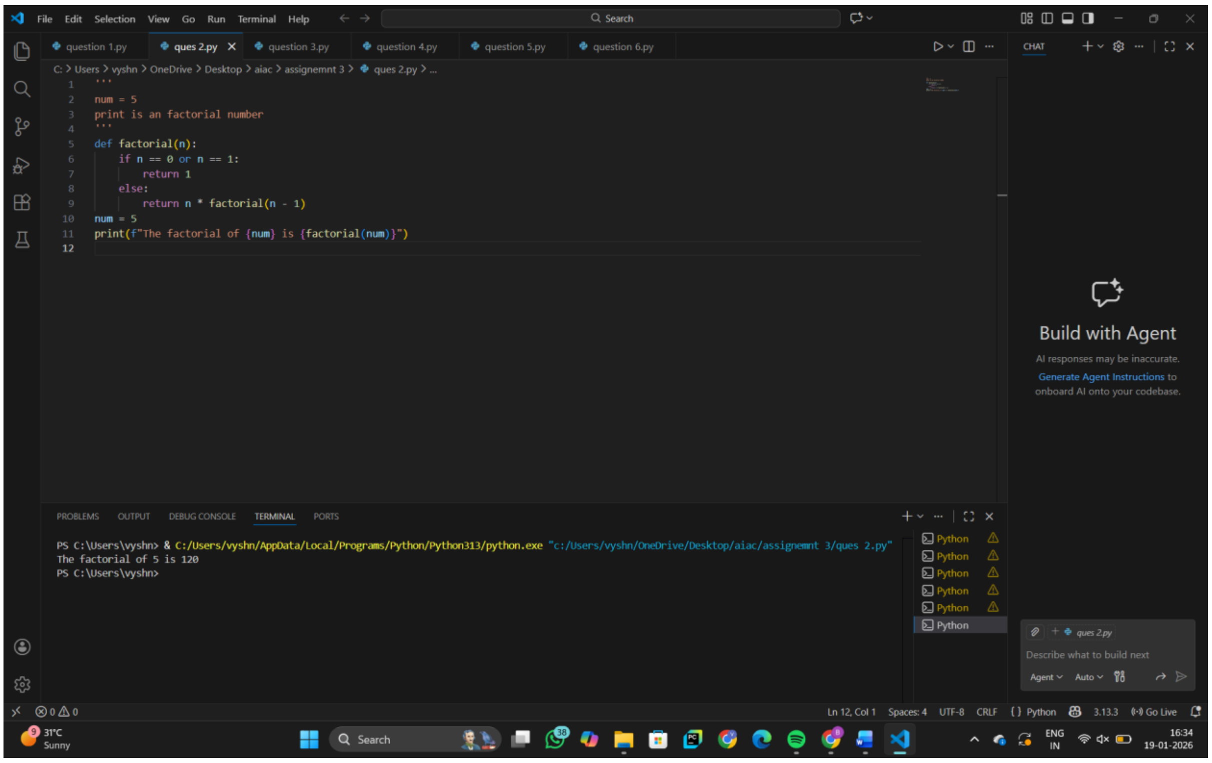
Example:

Input: 5 → Output: 120

Task:

• Compare the generated code with a zero-shot solution.

• Examine improvements in clarity and correctness.



```python
'''
num = 5
print is an factorial number
'''
def factorial(n):
    if n == 0 or n == 1:
        return 1
    else:
        return n * factorial(n - 1)
num = 5
print(f"The factorial of {num} is {factorial(num)}")
```

```
PS C:\Users\vyshn> & C:/Users/vyshn/AppData/Local/Programs/Python/Python313/python.exe "c:/Users/vyshn/OneDrive/Desktop/aiac/assignemnt 3/ques 2.py"
The factorial of 5 is 120
PS C:\Users\vyshn>
```

File Edit Selection View Go Run Terminal Help
question 1.py
ques 2.py
question 3.py
question 4.py
question 5.py
question 6.py
C: > Users > vyshn > OneDrive > Desktop > aiac > assignemnt 3 > ques 2.py > ...
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
Build with Agent
AI responses may be inaccurate.
Generate Agent Instructions to onboard AI onto your codebase.
Describe what to build next
Ln 12, Col 1 Spaces: 4 UTF-8 CRLF Python 3.13.3 Go Live

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

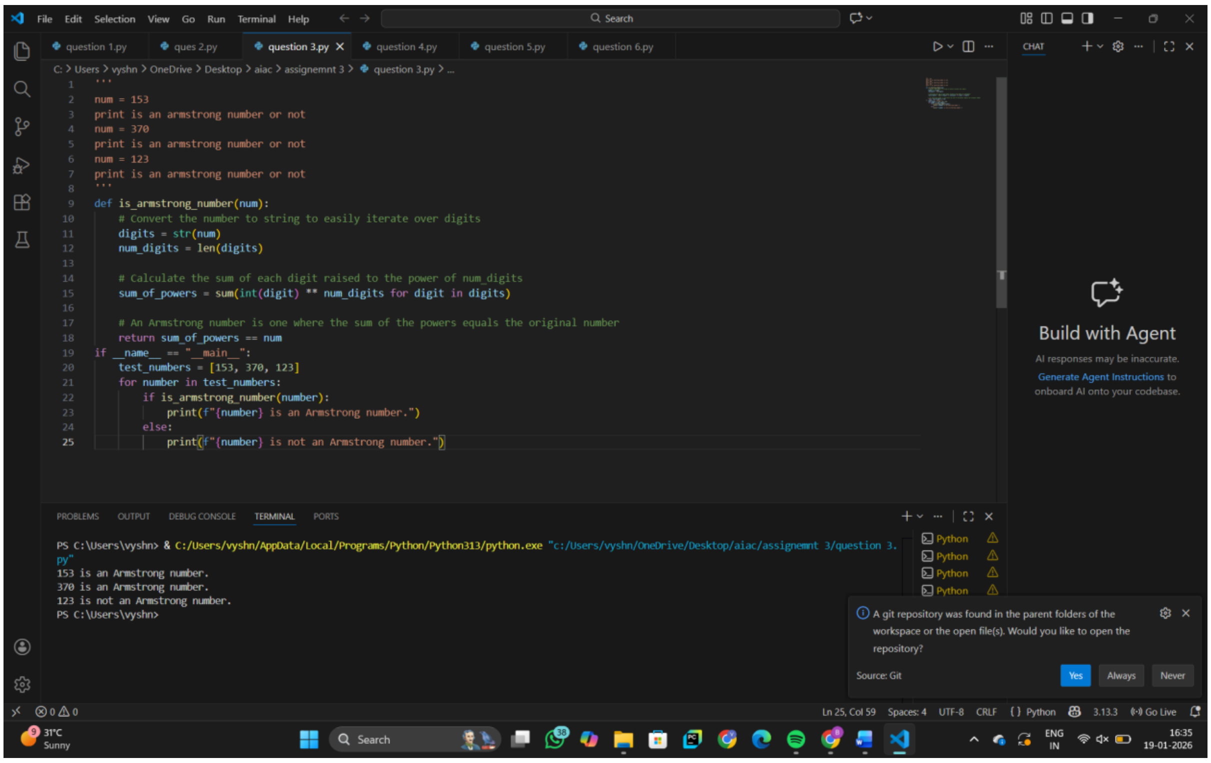
• Input: 370 → Output: Armstrong Number

• 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
'''
num = 153
print is an armstrong number or not
num = 370
print is an armstrong number or not
num = 123
print is an armstrong number or not
'''
def is_armstrong_number(num):
    # Convert the number to string to easily iterate over digits
    digits = str(num)
    num_digits = len(digits)

    # Calculate the sum of each digit raised to the power of num_digits
    sum_of_powers = sum(int(digit) ** num_digits for digit in digits)

    # An Armstrong number is one where the sum of the powers equals the original number
    return sum_of_powers == num
if __name__ == "__main__":
    test_numbers = [153, 370, 123]
    for number in test_numbers:
        if is_armstrong_number(number):
            print(f"{number} is an Armstrong number.")
        else:
            print(f"{number} is not an Armstrong number.")
```

```
PS C:\Users\vyshn> & C:/Users/vyshn/AppData/Local/Programs/Python/Python313/python.exe "c:/Users/vyshn/OneDrive/Desktop/aiac/assignemnt 3/question 3.py"
153 is an Armstrong number.
370 is an Armstrong number.
123 is not an Armstrong number.
PS C:\Users\vyshn>
```

A git repository was found in the parent folders of the workspace or the open file(s). Would you like to open the repository?

Source: Git

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

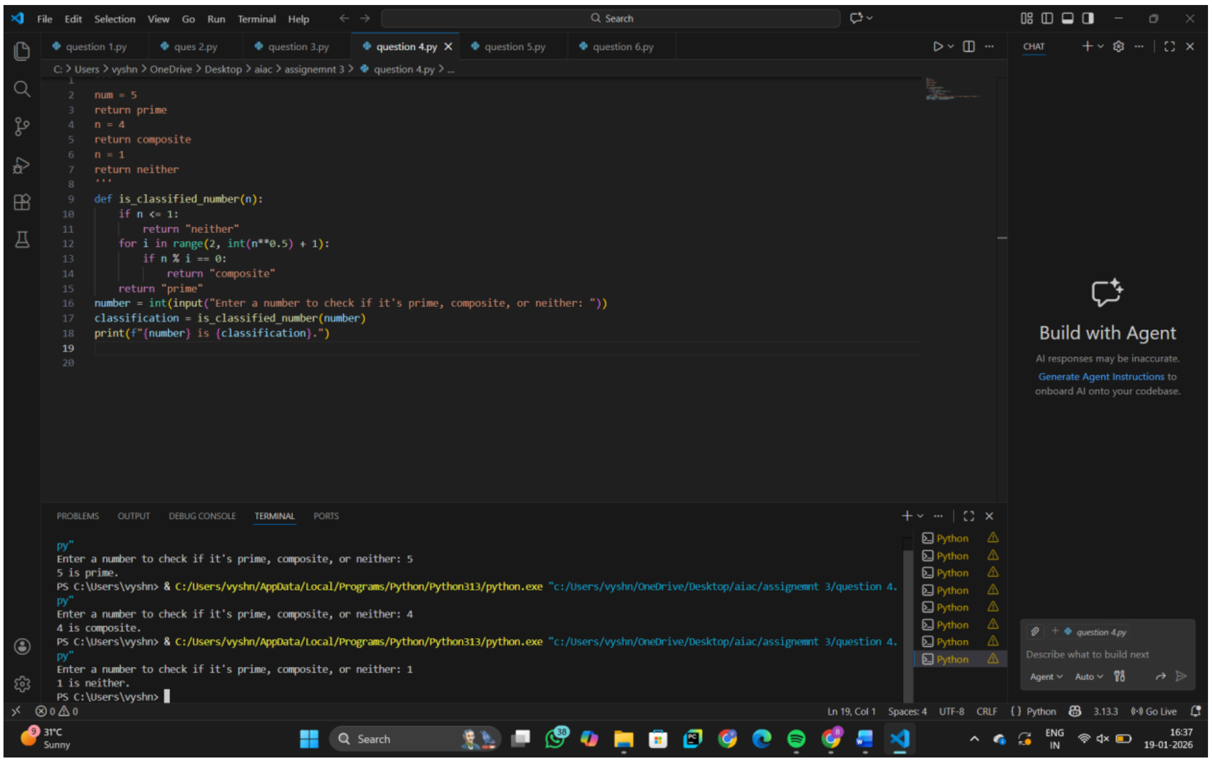
Design a context-managed prompt with clear instructions and constraints to generate an optimized Python program that classifies a number as prime, composite, or neither.

Task:

• Ensure proper input validation.

• Optimize the logic for efficiency.

• Compare the output with earlier prompting strategies.



```python
'''
num = 5
return prime
n = 4
return composite
n = 1
return neither
'''
def is_classified_number(n):
    if n <= 1:
        return "neither"
    for i in range(2, int(n**0.5) + 1):
        if n % i == 0:
            return "composite"
    return "prime"
number = int(input("Enter a number to check if it's prime, composite, or neither: "))
classification = is_classified_number(number)
print(f"{number} is {classification}.")
```

```
py"
Enter a number to check if it's prime, composite, or neither: 5
5 is prime.
PS C:\Users\vyshn> & C:/Users/vyshn/AppData/Local/Programs/Python/Python313/python.exe "c:/Users/vyshn/OneDrive/Desktop/aiac/assignemnt 3/question 4.
py"
Enter a number to check if it's prime, composite, or neither: 4
4 is composite.
PS C:\Users\vyshn> & C:/Users/vyshn/AppData/Local/Programs/Python/Python313/python.exe "c:/Users/vyshn/OneDrive/Desktop/aiac/assignemnt 3/question 4.
py"
Enter a number to check if it's prime, composite, or neither: 1
1 is neither.
PS C:\Users\vyshn>
```

Question 5: Zero-Shot Prompting (Perfect Number Check)

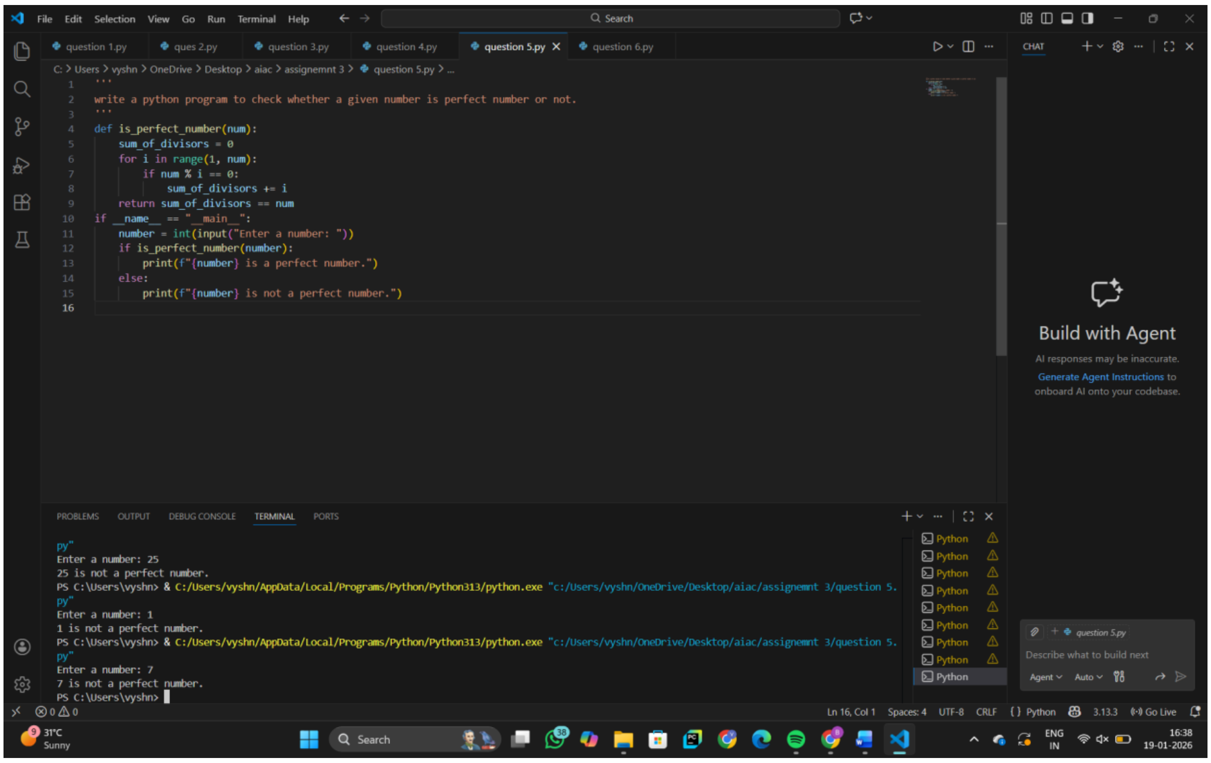
Write a zero-shot prompt (without providing any examples) to generate a Python function that checks whether a given number is a perfect number.

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 whether a given number is perfect number or not.
'''
def is_perfect_number(num):
    sum_of_divisors = 0
    for i in range(1, num):
        if num % i == 0:
            sum_of_divisors += i
    return sum_of_divisors == 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.")
```

```
py"
Enter a number: 25
25 is not a perfect number.
PS C:\Users\vyshn> & C:/Users/vyshn/AppData/Local/Programs/Python/Python313/python.exe "c:/Users/vyshn/OneDrive/Desktop/aiac/assignemnt 3/question 5.
py"
Enter a number: 1
1 is not a perfect number.
PS C:\Users\vyshn> & C:/Users/vyshn/AppData/Local/Programs/Python/Python313/python.exe "c:/Users/vyshn/OneDrive/Desktop/aiac/assignemnt 3/question 5.
py"
Enter a number: 7
7 is not a perfect number.
PS C:\Users\vyshn>
```

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

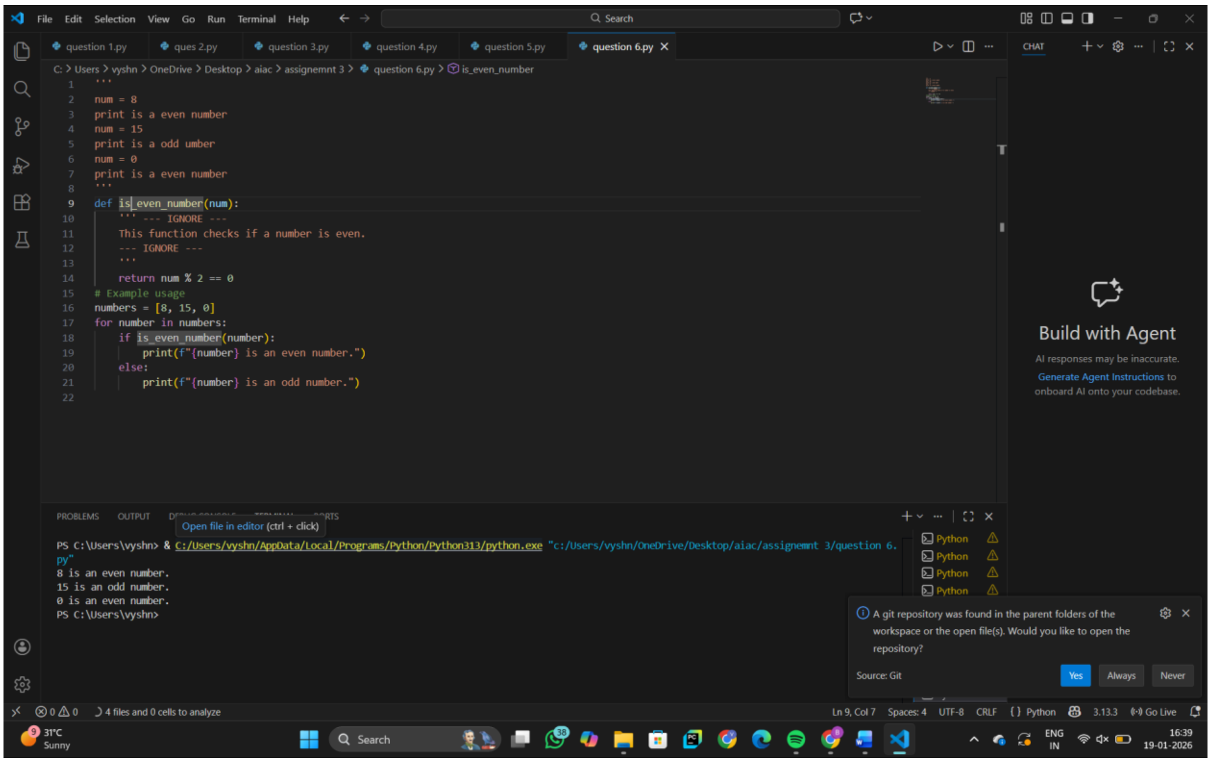
• Input: 15 → Output: Odd

• 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
'''
num = 8
print is a even number
num = 15
print is a odd umber
num = 0
print is a even number
'''
def is_even_number(num):
    ''' --- IGNORE ---
    This function checks if a number is even.
    --- IGNORE ---
    '''
    return num % 2 == 0
# Example usage
numbers = [8, 15, 0]
for number in numbers:
    if is_even_number(number):
        print(f"{number} is an even number.")
    else:
        print(f"{number} is an odd number.")
```

```
PS C:\Users\vyshn> & C:/Users/vyshn/AppData/Local/Programs/Python/Python313/python.exe "c:/Users/vyshn/OneDrive/Desktop/aiac/assignemnt 3/question 6.py"
8 is an even number.
15 is an odd number.
0 is an even number.
PS C:\Users\vyshn>
```

C: > Users > vyshn > OneDrive > Desktop > aiac > assignemnt 3 > question 6.py > is_even_number
Build with Agent
AI responses may be inaccurate.
Generate Agent Instructions to onboard AI onto your codebase.
A git repository was found in the parent folders of the workspace or the open file(s). Would you like to open the repository?
Source: Git
Yes
Always
Never