Assignment-3.1

Name: Sravani

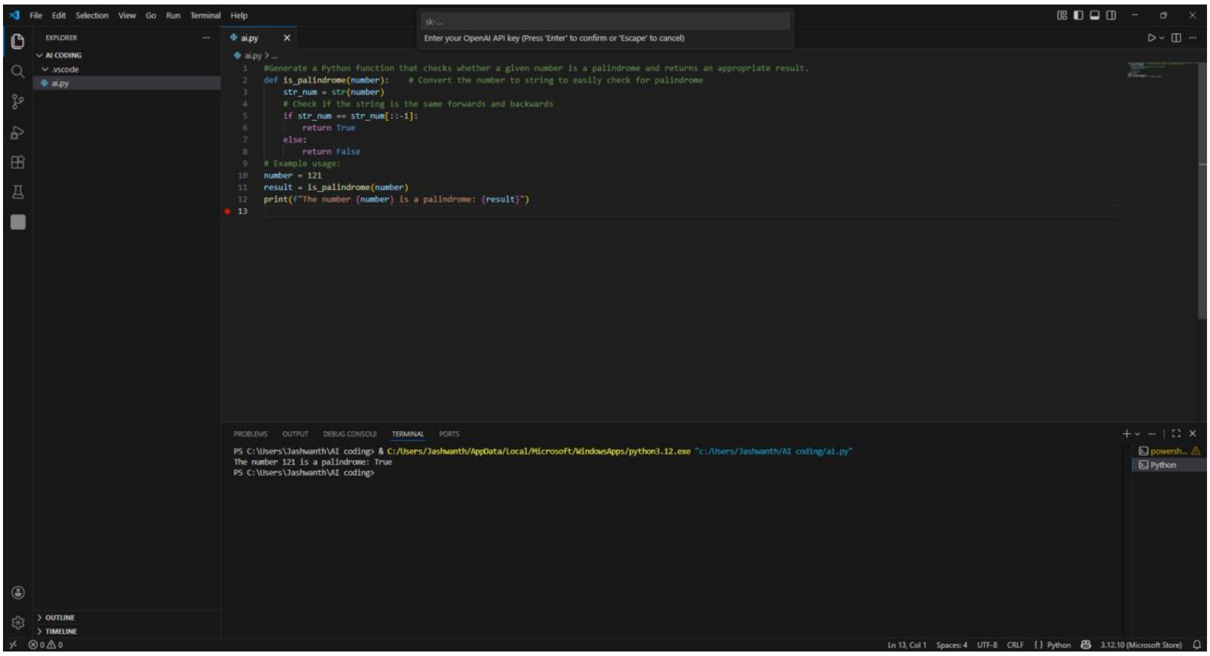
Roll.No:2303A510G7

Batch-30

Task-1

Prompt: Generate a Python function that checks whether a given number is a palindrome and returns an appropriate result

CODE:



sk-...

Enter your OpenAI API key (Press 'Enter' to confirm or 'Escape' to cancel)

```python
#Generate a Python function that checks whether a given number is a palindrome and returns an appropriate result.
def is_palindrome(number):    # Convert the number to string to easily check for palindrome
    str_num = str(number)
    # Check if the string is the same forwards and backwards
    if str_num == str_num[::-1]:
        return True
    else:
        return False
# Example usage:
number = 121
result = is_palindrome(number)
print(f"The number {number} is a palindrome: {result}")

```

```
PS C:\Users\Jashwanth\AI coding> & C:/Users/Jashwanth/AppData/Local/Microsoft/WindowsApps/python3.12.exe "c:/Users/Jashwanth/AI coding/ai.py"
The number 121 is a palindrome: True
PS C:\Users\Jashwanth\AI coding>
```

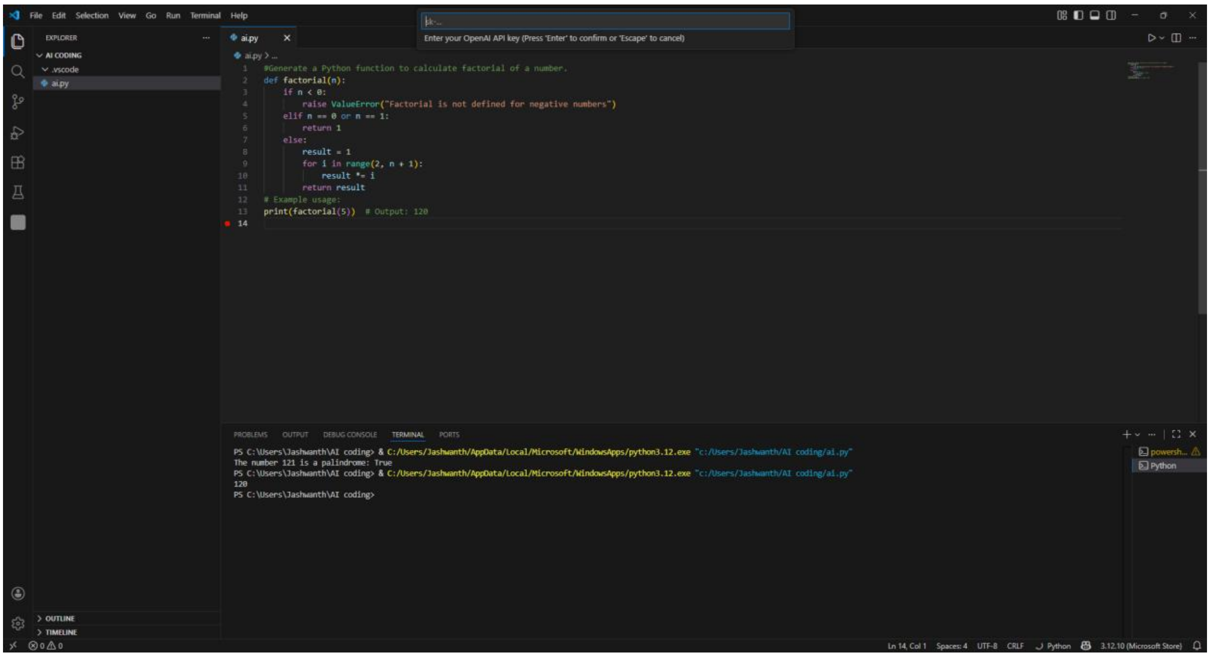
OBSERVATION:

- **The program correctly checks whether the given number is a palindrome.**
- **For the input 121, the output is correct.**
- **The logic works properly for positive numbers but does not handle invalid inputs.**

Task-2

Prompt: Generate a Python function to calculate factorial of a number.

CODE:



File Edit Selection View Go Run Terminal Help
EXPLORER
AI CODING
.vscode
ai.py
OUTLINE
TIMELINE
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
powersh...
Python
Ln 14, Col 1 Spaces: 4 UTF-8 CRLF Python 3.12.10 (Microsoft Store)

sk-...

Enter your OpenAI API key (Press 'Enter' to confirm or 'Escape' to cancel)

```python
#Generate a Python function to calculate factorial of a number.
def factorial(n):
    if n < 0:
        raise ValueError("Factorial is not defined for negative numbers")
    elif n == 0 or n == 1:
        return 1
    else:
        result = 1
        for i in range(2, n + 1):
            result *= i
        return result
# Example usage:
print(factorial(5))  # Output: 120

```

```
PS C:\Users\Jashwanth\AI coding> & C:/Users/Jashwanth/AppData/Local/Microsoft/WindowsApps/python3.12.exe "c:/Users/Jashwanth/AI coding/ai.py"
The number 121 is a palindrome: True
PS C:\Users\Jashwanth\AI coding> & C:/Users/Jashwanth/AppData/Local/Microsoft/WindowsApps/python3.12.exe "c:/Users/Jashwanth/AI coding/ai.py"
120
PS C:\Users\Jashwanth\AI coding>
```

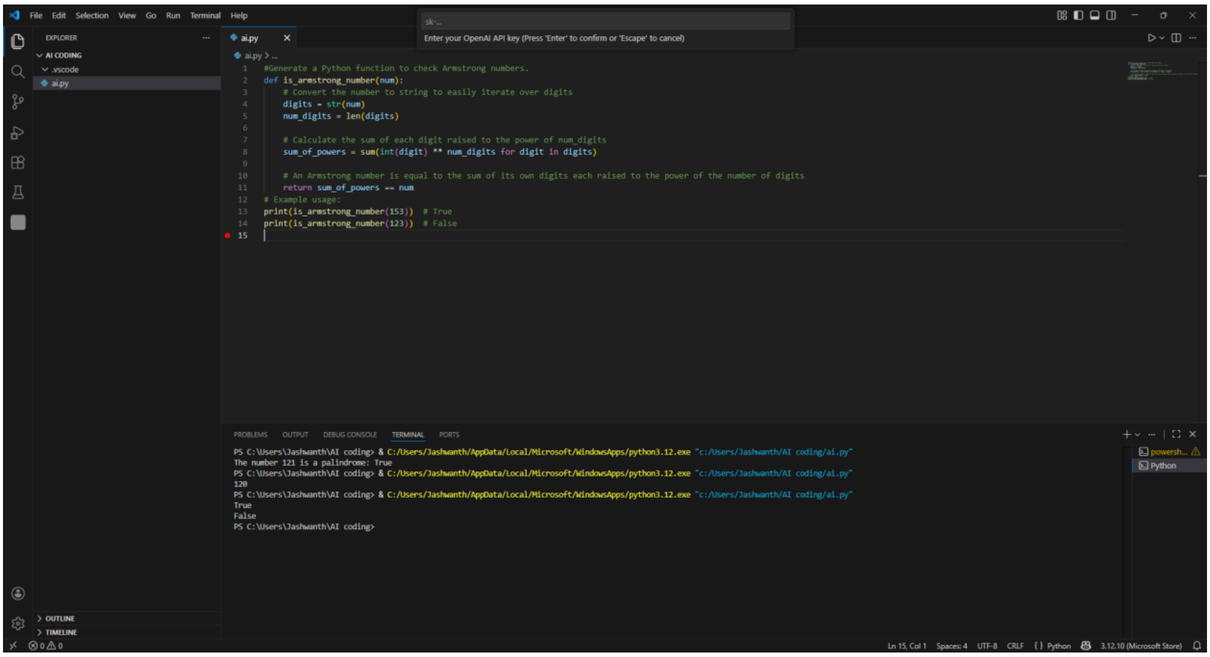
OBSERVATION:

- The program correctly calculates the factorial of the given number.
- For the input **5**, the output **120** is obtained.
- It works properly for valid positive numbers.

Task-3

Prompt: Generate a Python function to check Armstrong numbers.

CODE:



sk-...

Enter your OpenAI API key (Press 'Enter' to confirm or 'Escape' to cancel)

```python
#Generate a Python function to check Armstrong numbers.
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 equal to the sum of its own digits each raised to the power of the number of digits
    return sum_of_powers == num
# Example usage:
print(is_armstrong_number(153))  # True
print(is_armstrong_number(123))  # False

```

```
PS C:\Users\Jashwanth\AI coding> & C:/Users/Jashwanth/AppData/Local/Microsoft/WindowsApps/python3.12.exe "c:/Users/Jashwanth/AI coding/ai.py"
The number 121 is a palindrome: True
PS C:\Users\Jashwanth\AI coding> & C:/Users/Jashwanth/AppData/Local/Microsoft/WindowsApps/python3.12.exe "c:/Users/Jashwanth/AI coding/ai.py"
120
PS C:\Users\Jashwanth\AI coding> & C:/Users/Jashwanth/AppData/Local/Microsoft/WindowsApps/python3.12.exe "c:/Users/Jashwanth/AI coding/ai.py"
True
False
PS C:\Users\Jashwanth\AI coding>
```

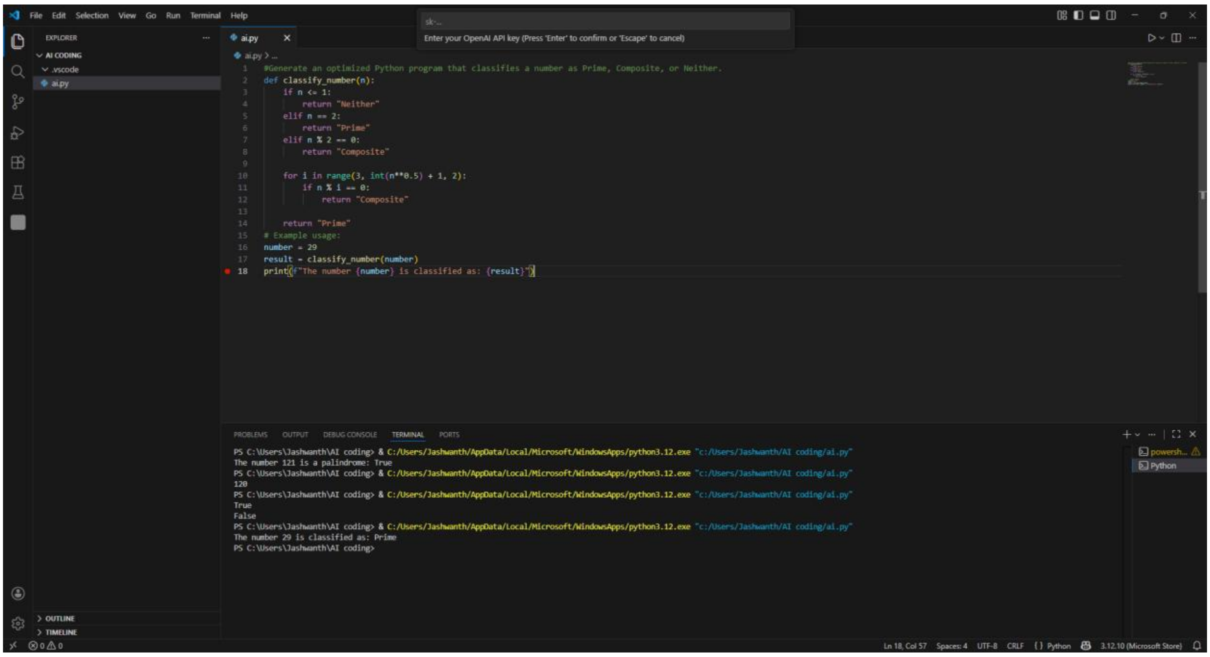
OBSERVATION:

- The program correctly checks whether a number is an Armstrong number.
- For the input **153**, the output is **True**, and for **123**, the output is **False**.
- The logic works correctly for valid inputs.

Task-4

Prompt: Generate an optimized Python program that classifies a number as Prime, Composite, or Neither.

CODE:



File Edit Selection View Go Run Terminal Help
EXPLORER
AI CODING
.vscode
ai.py
ai.py
ai.py > ...
sk-...
Enter your OpenAI API key (Press 'Enter' to confirm or 'Escape' to cancel)
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
powersh...
Python
OUTLINE
TIMELINE
Ln 18, Col 57 Spaces: 4 UTF-8 CRLF {} Python 3.12.10 (Microsoft Store)

```python
#Generate an optimized Python program that classifies a number as Prime, Composite, or Neither.
def classify_number(n):
    if n <= 1:
        return "Neither"
    elif n == 2:
        return "Prime"
    elif n % 2 == 0:
        return "Composite"

    for i in range(3, int(n**0.5) + 1, 2):
        if n % i == 0:
            return "Composite"

    return "Prime"
# Example usage:
number = 29
result = classify_number(number)
print(f"The number {number} is classified as: {result}")
```

```
PS C:\Users\Jashwanth\AI coding> & C:/Users/Jashwanth/AppData/Local/Microsoft/WindowsApps/python3.12.exe "c:/Users/Jashwanth/AI coding/ai.py"
The number 121 is a palindrome: True
PS C:\Users\Jashwanth\AI coding> & C:/Users/Jashwanth/AppData/Local/Microsoft/WindowsApps/python3.12.exe "c:/Users/Jashwanth/AI coding/ai.py"
120
PS C:\Users\Jashwanth\AI coding> & C:/Users/Jashwanth/AppData/Local/Microsoft/WindowsApps/python3.12.exe "c:/Users/Jashwanth/AI coding/ai.py"
True
False
PS C:\Users\Jashwanth\AI coding> & C:/Users/Jashwanth/AppData/Local/Microsoft/WindowsApps/python3.12.exe "c:/Users/Jashwanth/AI coding/ai.py"
The number 29 is classified as: Prime
PS C:\Users\Jashwanth\AI coding>
```

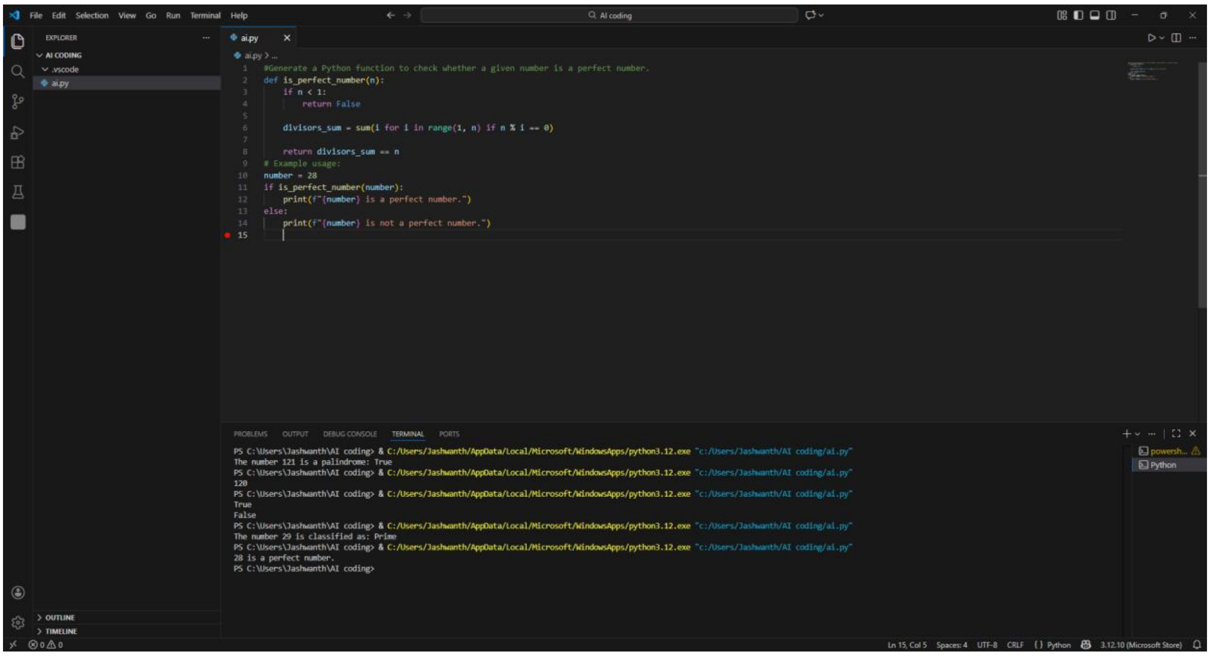
OBSERVATION:

- **The program correctly classifies the given number as prime, composite, or neither.**
- **For the input 29, the output is Prime.**
- **The logic works properly for valid numbers.**

Task-5

Prompt: Generate a Python function to check whether a given number is a perfect number

CODE:



```python
#Generate a Python function to check whether a given number is a perfect number.
def is_perfect_number(n):
    if n < 1:
        return False

    divisors_sum = sum(i for i in range(1, n) if n % i == 0)

    return divisors_sum == n
# Example usage:
number = 28
if is_perfect_number(number):
    print(f"{number} is a perfect number.")
else:
    print(f"{number} is not a perfect number.")

```

```
PS C:\Users\Jashwanth\AI coding> & C:/Users/Jashwanth/AppData/Local/Microsoft/WindowsApps/python3.12.exe "c:/Users/Jashwanth/AI coding/ai.py"
The number 121 is a palindrome: True
PS C:\Users\Jashwanth\AI coding> & C:/Users/Jashwanth/AppData/Local/Microsoft/WindowsApps/python3.12.exe "c:/Users/Jashwanth/AI coding/ai.py"
120
PS C:\Users\Jashwanth\AI coding> & C:/Users/Jashwanth/AppData/Local/Microsoft/WindowsApps/python3.12.exe "c:/Users/Jashwanth/AI coding/ai.py"
True
False
PS C:\Users\Jashwanth\AI coding> & C:/Users/Jashwanth/AppData/Local/Microsoft/WindowsApps/python3.12.exe "c:/Users/Jashwanth/AI coding/ai.py"
The number 29 is classified as: Prime
PS C:\Users\Jashwanth\AI coding> & C:/Users/Jashwanth/AppData/Local/Microsoft/WindowsApps/python3.12.exe "c:/Users/Jashwanth/AI coding/ai.py"
28 is a perfect number.
PS C:\Users\Jashwanth\AI coding>
```

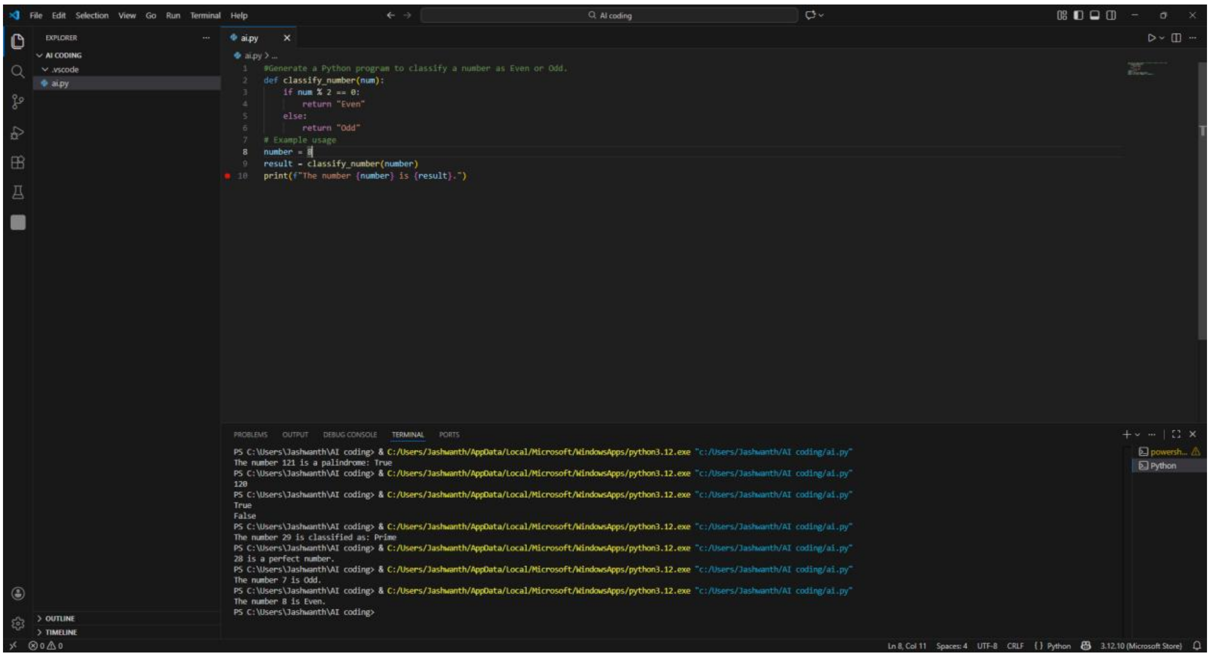
OBSERVATION:

- The program correctly checks whether a number is a perfect number.
- For the input **28**, the output shows that it is a perfect number.
- The logic works correctly for valid inputs.

Task-6

Prompt: Generate a Python program to classify a number as Even or Odd.

CODE:



```python
#Generate a Python program to classify a number as Even or Odd.
def classify_number(num):
    if num % 2 == 0:
        return "Even"
    else:
        return "Odd"
# Example usage
number = 8
result = classify_number(number)
print(f"The number {number} is {result}.")
```

```
PS C:\Users\Jashwanth\AI coding> & C:/Users/Jashwanth/AppData/Local/Microsoft/WindowsApps/python3.12.exe "c:/Users/Jashwanth/AI coding/ai.py"
The number 121 is a palindrome: True
PS C:\Users\Jashwanth\AI coding> & C:/Users/Jashwanth/AppData/Local/Microsoft/WindowsApps/python3.12.exe "c:/Users/Jashwanth/AI coding/ai.py"
120
PS C:\Users\Jashwanth\AI coding> & C:/Users/Jashwanth/AppData/Local/Microsoft/WindowsApps/python3.12.exe "c:/Users/Jashwanth/AI coding/ai.py"
True
False
PS C:\Users\Jashwanth\AI coding> & C:/Users/Jashwanth/AppData/Local/Microsoft/WindowsApps/python3.12.exe "c:/Users/Jashwanth/AI coding/ai.py"
The number 29 is classified as: Prime
PS C:\Users\Jashwanth\AI coding> & C:/Users/Jashwanth/AppData/Local/Microsoft/WindowsApps/python3.12.exe "c:/Users/Jashwanth/AI coding/ai.py"
28 is a perfect number.
PS C:\Users\Jashwanth\AI coding> & C:/Users/Jashwanth/AppData/Local/Microsoft/WindowsApps/python3.12.exe "c:/Users/Jashwanth/AI coding/ai.py"
The number 7 is Odd.
PS C:\Users\Jashwanth\AI coding> & C:/Users/Jashwanth/AppData/Local/Microsoft/WindowsApps/python3.12.exe "c:/Users/Jashwanth/AI coding/ai.py"
The number 8 is Even.
PS C:\Users\Jashwanth\AI coding>
```

OBSERVATION:

- The program correctly identifies whether a number is even or odd.
- For the input **8**, the output is **Even**.
- The logic works correctly for valid inputs.