

Assignment-3.1

Name: Sravani

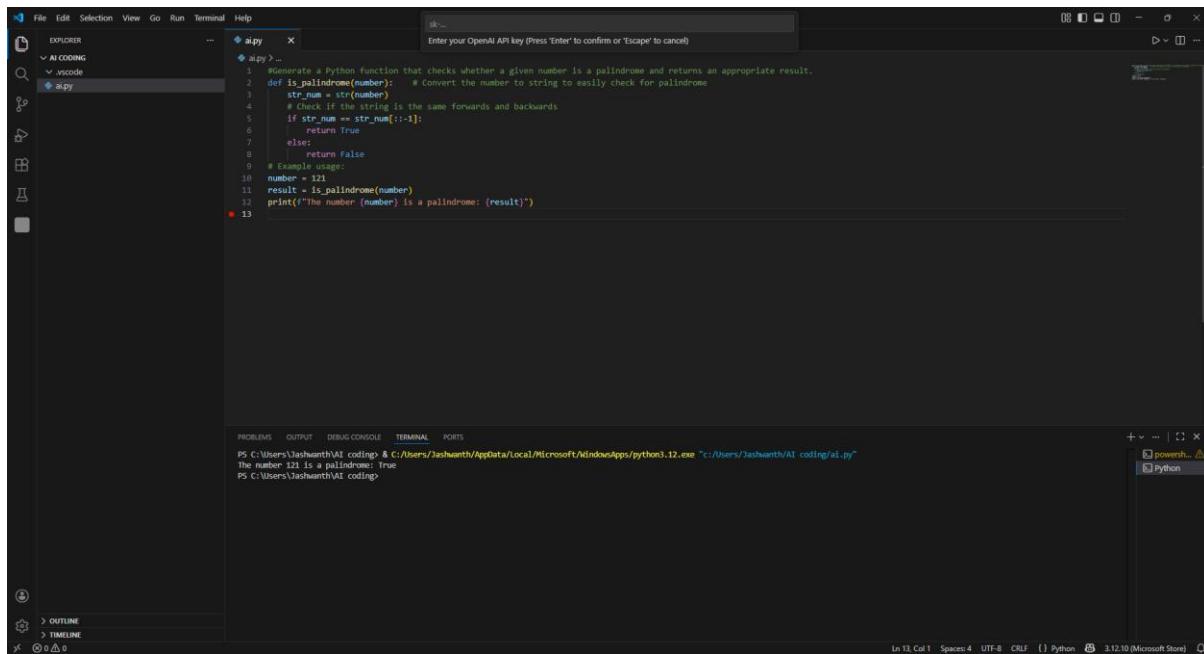
Roll.No:2303A510G7

Batch-30

Task-1

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

CODE:



```
#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("The number {} is a palindrome: {}".format(number, result))
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

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:

The screenshot shows the Visual Studio Code interface with the following details:

- File Explorer:** Shows a folder named "AI CODING" containing a file named "ai.py".
- Code Editor:** Displays the following Python code:

```
# 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
```
- Terminal:** Shows the command line output:

```
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 120 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>
```
- Bottom Status Bar:** Shows "In 14, Col 1" and "3.12.10 (Microsoft Store)".

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:

The screenshot shows a Microsoft Visual Studio Code (VS Code) interface. The top bar includes File, Edit, Selection, View, Go, Run, Terminal, and Help menus. A status bar at the bottom indicates the file is 'ai.py', has 312.10 (Microsoft Store) open, and shows 'In 15, Col 1' with 'Spaces: 4' and 'UTF-8' encoding.

The main area displays the following Python code:

```
ai.py
1  #Generate a Python function to check Armstrong numbers.
2  def is_armstrong_number(num):
3      # Convert the number to string to easily iterate over digits
4      digits = str(num)
5      num_digits = len(digits)
6
7      # Calculate the sum of each digit raised to the power of num_digits
8      sum_of_powers = sum(int(digit)**num_digits for digit in digits)
9
10     # An Armstrong number is equal to the sum of its own digits each raised to the power of the number of digits
11     return sum_of_powers == num
12
13 # Example usage:
14 print(is_armstrong_number(153)) # True
15 print(is_armstrong_number(123)) # False
```

The terminal below the editor shows the output of running the script:

```
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"
153
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:

The screenshot shows the Visual Studio Code interface with the following details:

- File Explorer:** Shows a folder named "AI CODING" containing ".vscode" and "ai.py".
- Code Editor:** Displays the content of "ai.py".

```
#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"
    else:
        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}")
```
- Terminal:** Shows the output of running the script with input 29.

```
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 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"
121
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:

The screenshot shows a dark-themed instance of Visual Studio Code. In the center, there's a code editor window titled 'ai.py' containing the following Python script:

```
#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)
    if divisors_sum == n:
        print(f"{n} is a perfect number.")
    else:
        print(f"{n} is not a perfect number.")

# 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.")
```

Below the code editor, the terminal tab is active, showing the output of running the script with the input 28. The output is as follows:

```
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 28 is a palindrome.
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 28 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>
```

At the bottom right of the terminal, it says 'Ln 15, Col 5'. To the right of the terminal, there's a small Python icon and the text '3.12.10 (Microsoft Store)'.

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:

The screenshot shows the VS Code interface with a dark theme. The Explorer sidebar on the left shows a folder named 'AI CODING' containing a '.vscode' folder and an 'ai.py' file. The code editor window displays the following Python script:

```
#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}.")
```

The terminal at the bottom shows the output of running the script with different inputs:

```
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"
29 is not a prime 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.