AI ASSISTED CODING-LAB ASSIGNMENT

ASSIGNMENT – 3.4

Name: K.SUNNY

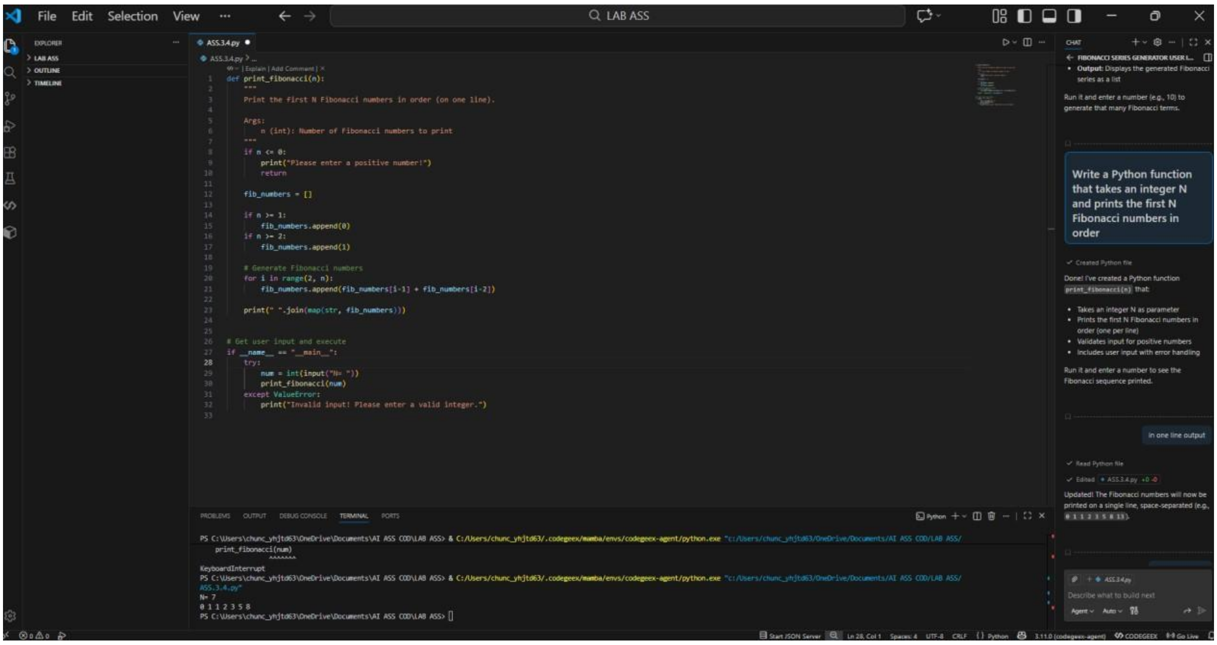
Batch:13

Task 1: Zero-shot Prompt – Fibonacci Series Generator ask

Description #1

- Without giving an example, write a single comment prompt asking GitHub Copilot to generate a Python function to print the first N Fibonacci numbers.

Prompt : "# Write a Python function that takes an integer N and prints the first N Fibonacci numbers in order" Code and output :



```python
def print_fibonacci(n):
    """
    Print the first N Fibonacci numbers in order (on one line).

    Args:
        n (int): Number of Fibonacci numbers to print
    """
    if n <= 0:
        print("Please enter a positive number!")
        return

    fib_numbers = []

    if n >= 1:
        fib_numbers.append(0)
    if n >= 2:
        fib_numbers.append(1)

    # Generate Fibonacci numbers
    for i in range(2, n):
        fib_numbers.append(fib_numbers[i-1] + fib_numbers[i-2])

    print(" ".join(map(str, fib_numbers)))


# Get user input and execute
if __name__ == "__main__":
    try:
        num = int(input("N= "))
        print_fibonacci(num)
    except ValueError:
        print("Invalid input! Please enter a valid integer.")
```

```
PS C:\Users\chunc_yhjtd63\OneDrive\Documents\AI ASS COD\LAB ASS> & C:/Users/chunc_yhjtd63/.codegeex/mamba/envs/codegeex-agent/python.exe "c:/Users/chunc_yhjtd63/OneDrive/Documents/AI ASS COD/LAB ASS/
    print_fibonacci(num)
    ^^^^^^^
KeyboardInterrupt
PS C:\Users\chunc_yhjtd63\OneDrive\Documents\AI ASS COD\LAB ASS> & C:/Users/chunc_yhjtd63/.codegeex/mamba/envs/codegeex-agent/python.exe "c:/Users/chunc_yhjtd63/OneDrive/Documents/AI ASS COD/LAB ASS/
ASS.3.4.py"
N= 7
0 1 1 2 3 5 8
PS C:\Users\chunc_yhjtd63\OneDrive\Documents\AI ASS COD\LAB ASS> 
```

FIBONACCI SERIES GENERATOR USER I...

- **Output:** Displays the generated Fibonacci series as a list

Run it and enter a number (e.g., 10) to generate that many Fibonacci terms.

Write a Python function that takes an integer N and prints the first N Fibonacci numbers in order

✓ Created Python file

Done! I've created a Python function `print_fibonacci(n)` that:

- Takes an integer N as parameter
- Prints the first N Fibonacci numbers in order (one per line)
- Validates input for positive numbers
- Includes user input with error handling

Run it and enter a number to see the Fibonacci sequence printed.

in one line output

✓ Read Python file

✓ Edited ASS.3.4.py +0 -0

Updated! The Fibonacci numbers will now be printed on a single line, space-separated (e.g., `0 1 1 2 3 5 8 13`).

Task 2: One-shot Prompt – List Reversal Function

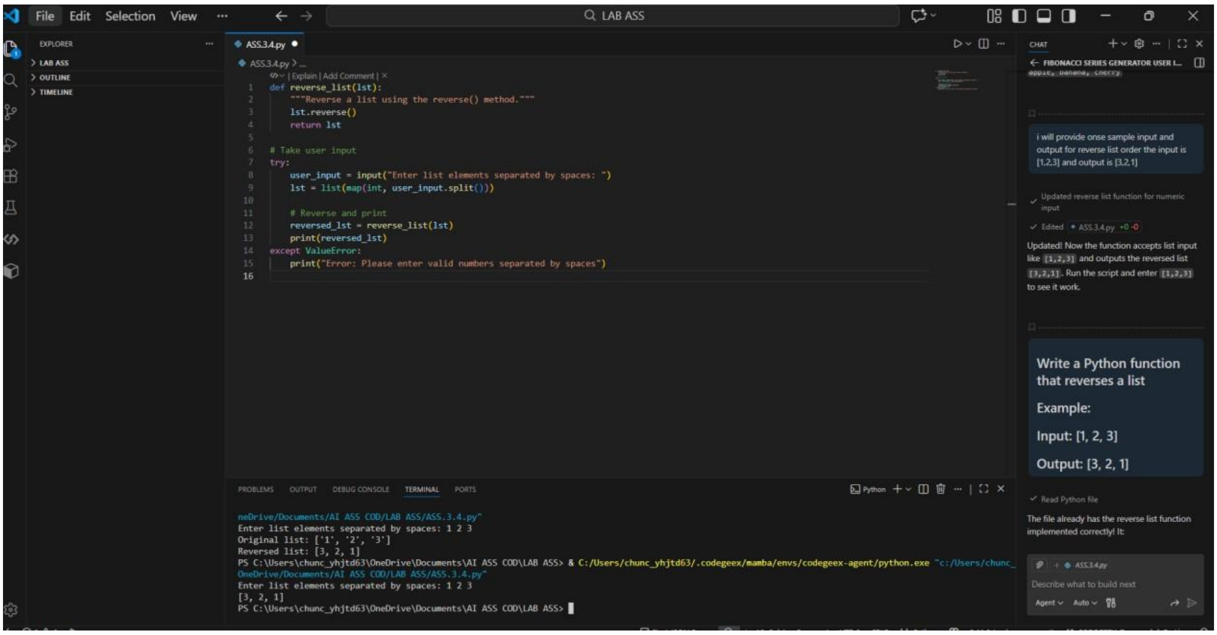
Task Description #2

- Write a comment prompt to reverse a list and provide one example

  below the comment to guide Copilot.

Prompt: # Write a Python function that takes a list as input and returns the list in reverse order

Code and output :



```python
def reverse_list(lst):
    """Reverse a list using the reverse() method."""
    lst.reverse()
    return lst

# Take user input
try:
    user_input = input("Enter list elements separated by spaces: ")
    lst = list(map(int, user_input.split()))

    # Reverse and print
    reversed_lst = reverse_list(lst)
    print(reversed_lst)
except ValueError:
    print("Error: Please enter valid numbers separated by spaces")
```

```
neDrive/Documents/AI ASS COD/LAB ASS/ASS.3.4.py"
Enter list elements separated by spaces: 1 2 3
Original list: ['1', '2', '3']
Reversed list: [3, 2, 1]
PS C:\Users\chunc_yhjtd63\OneDrive\Documents\AI ASS COD\LAB ASS> & C:/Users/chunc_yhjtd63/.codegeex/mamba/envs/codegeex-agent/python.exe "c:/Users/chunc_
OneDrive/Documents/AI ASS COD/LAB ASS/ASS.3.4.py"
Enter list elements separated by spaces: 1 2 3
[3, 2, 1]
PS C:\Users\chunc_yhjtd63\OneDrive\Documents\AI ASS COD\LAB ASS>
```

apple, banana, cherry

i will provide onse sample input and output for reverse list order the input is [1.2.3] and output is [3.2.1]

✓ Updated reverse list function for numeric input

✓ Edited ASS.3.4.py +0 -0

Updated! Now the function accepts list input like `[1,2,3]` and outputs the reversed list `[3,2,1]`. Run the script and enter `[1,2,3]` to see it work.

Write a Python function that reverses a list

Example:

Input: [1, 2, 3]

Output: [3, 2, 1]

✓ Read Python file

The file already has the reverse list function implemented correctly! It:

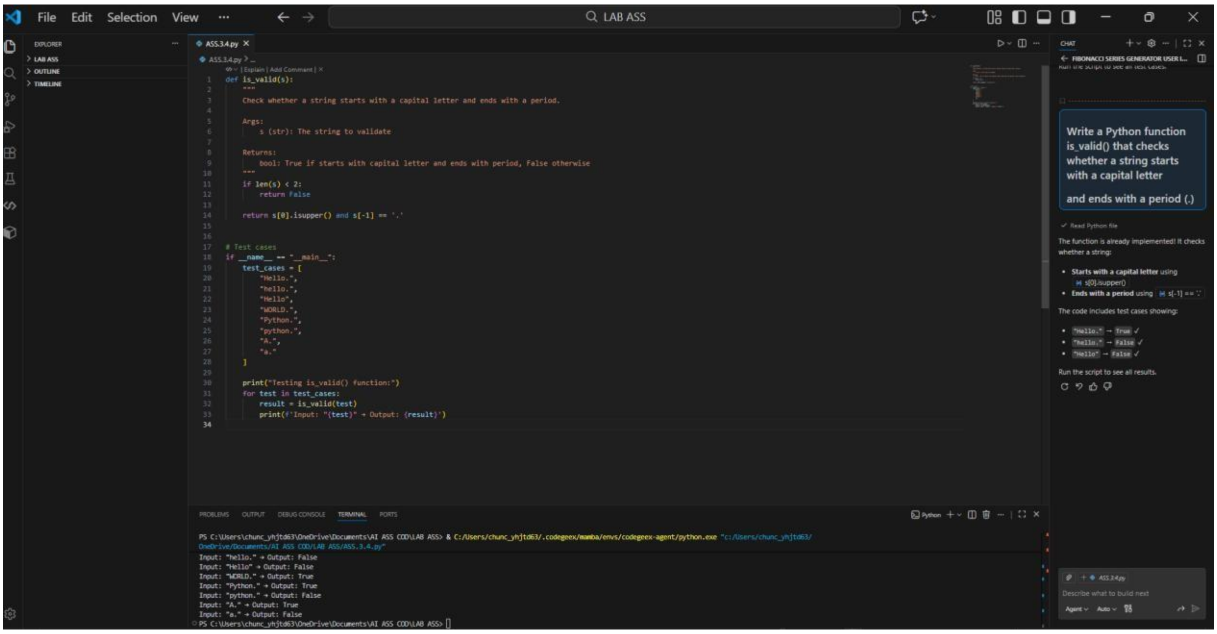
Task 3: Few-shot Prompt – String Pattern Matching

Task Description #3

- Write a comment with 2–3 examples to help Copilot understand how to check if a string starts with a capital letter and ends with a period.

Prompt: # Write a Python function is_valid() that checks whether a string starts with a capital letter
# and ends with a period (.)

Code and output :



```python
def is_valid(s):
    """
    Check whether a string starts with a capital letter and ends with a period.

    Args:
        s (str): The string to validate

    Returns:
        bool: True if starts with capital letter and ends with period, False otherwise
    """
    if len(s) < 2:
        return False

    return s[0].isupper() and s[-1] == '.'


# Test cases
if __name__ == "__main__":
    test_cases = [
        "Hello.",
        "hello.",
        "Hello",
        "WORLD.",
        "Python.",
        "python.",
        "A.",
        "a."
    ]

    print("Testing is_valid() function:")
    for test in test_cases:
        result = is_valid(test)
        print(f'Input: "{test}" → Output: {result}')
```

```
PS C:\Users\chunc_yhjtd63\OneDrive\Documents\AI ASS COD\LAB ASS> & C:/Users/chunc_yhjtd63/.codegeex/mamba/envs/codegeex-agent/python.exe "c:/Users/chunc_yhjtd63/
OneDrive/Documents/AI ASS COD/LAB ASS/ASS.3.4.py"
Input: "hello." → Output: False
Input: "Hello" → Output: False
Input: "WORLD." → Output: True
Input: "Python." → Output: True
Input: "python." → Output: False
Input: "A." → Output: True
Input: "a." → Output: False
PS C:\Users\chunc_yhjtd63\OneDrive\Documents\AI ASS COD\LAB ASS> 
```

Write a Python function is_valid() that checks whether a string starts with a capital letter

and ends with a period (.)

✓ Read Python file

The function is already implemented! It checks whether a string:

- **Starts with a capital letter** using s[0].isupper()
- **Ends with a period** using s[-1] == '.'

The code includes test cases showing:

- "Hello." → True ✓
- "hello." → False ✓
- "Hello" → False ✓

Run the script to see all results.

Task 4: Zero-shot vs Few-shot – Email Validator

Task Description #4

- First, prompt Copilot to write an email validation function using zero-shot (just the task in comment).
- Then, rewrite the prompt using few-shot examples.

Prompt: # Write a Python function that validates whether a given string is a valid email address

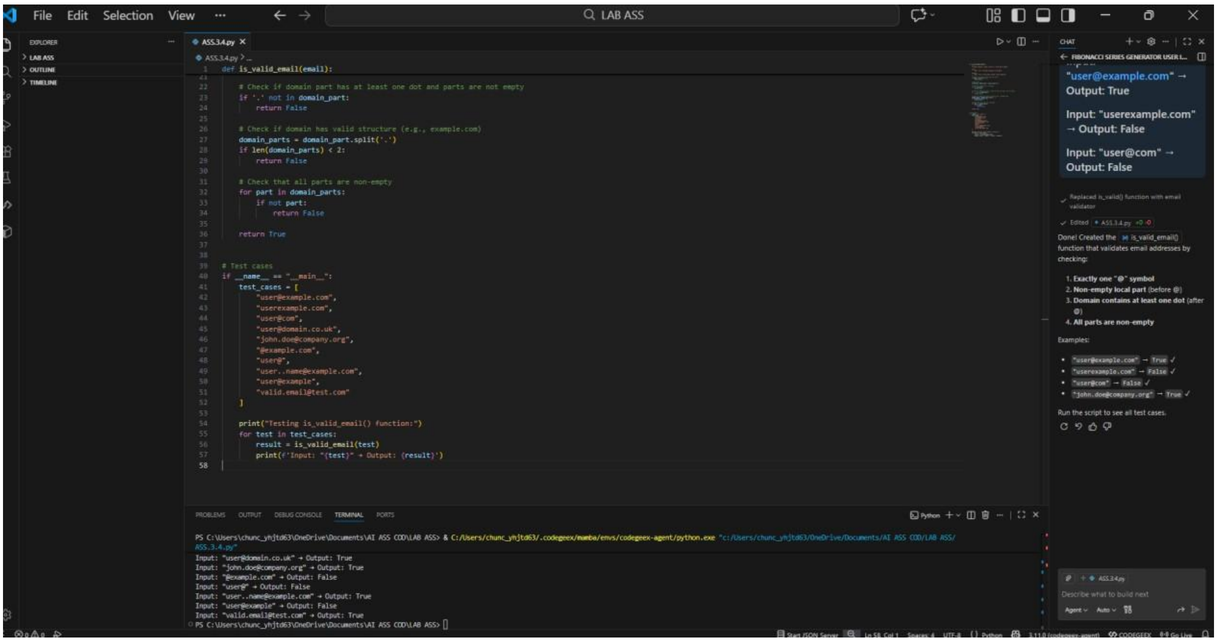
# Write a Python function that validates whether a given string is a valid email address # Examples:

# Input: "user@example.com" → Output: True

# Input: "userexample.com" → Output: False

# Input: "user@com"   → Output: False

Code and output :



File Edit Selection View
LAB ASS
ASS.3.4.py
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
CHAT
FIBONACCI SERIES GENERATOR USER I...
Describe what to build next
Agent Auto

```python
def is_valid_email(email):
        # Check if domain part has at least one dot and parts are not empty
        if '.' not in domain_part:
            return False

        # Check if domain has valid structure (e.g., example.com)
        domain_parts = domain_part.split('.')
        if len(domain_parts) < 2:
            return False

        # Check that all parts are non-empty
        for part in domain_parts:
            if not part:
                return False

        return True


# Test cases
if __name__ == "__main__":
    test_cases = [
        "user@example.com",
        "userexample.com",
        "user@com",
        "user@domain.co.uk",
        "john.doe@company.org",
        "@example.com",
        "user@",
        "user..name@example.com",
        "user@example",
        "valid.email@test.com"
    ]

    print("Testing is_valid_email() function:")
    for test in test_cases:
        result = is_valid_email(test)
        print(f'Input: "{test}" → Output: {result}')
```

```
PS C:\Users\chunc_yhjtd63\OneDrive\Documents\AI ASS COD\LAB ASS> & C:/Users/chunc_yhjtd63/.codegeex/mamba/envs/codegeex-agent/python.exe "c:/Users/chunc_yhjtd63/OneDrive/Documents/AI ASS COD/LAB ASS/ASS.3.4.py"
Input: "user@domain.co.uk" → Output: True
Input: "john.doe@company.org" → Output: True
Input: "@example.com" → Output: False
Input: "user@" → Output: False
Input: "user..name@example.com" → Output: True
Input: "user@example" → Output: False
Input: "valid.email@test.com" → Output: True
PS C:\Users\chunc_yhjtd63\OneDrive\Documents\AI ASS COD\LAB ASS>
```

"user@example.com" → Output: True

Input: "userexample.com" → Output: False

Input: "user@com" → Output: False

✓ Replaced is_valid() function with email validator

✓ Edited ASS.3.4.py +0 -0

Done! Created the is_valid_email() function that validates email addresses by checking:

1. **Exactly one "@" symbol**
2. **Non-empty local part** (before @)
3. **Domain contains at least one dot** (after @)
4. **All parts are non-empty**

Examples:

- `"user@example.com"` → `True` ✓
- `"userexample.com"` → `False` ✓
- `"user@com"` → `False` ✓
- `"john.doe@company.org"` → `True` ✓

Run the script to see all test cases.

Task 5: Prompt Tuning – Summing Digits of a Number

Task Description #5

- Experiment with 2 different prompt styles to generate a function that returns the sum of digits of a number.

Style 1: Generic task prompt

Style 2: Task + Input/Output example

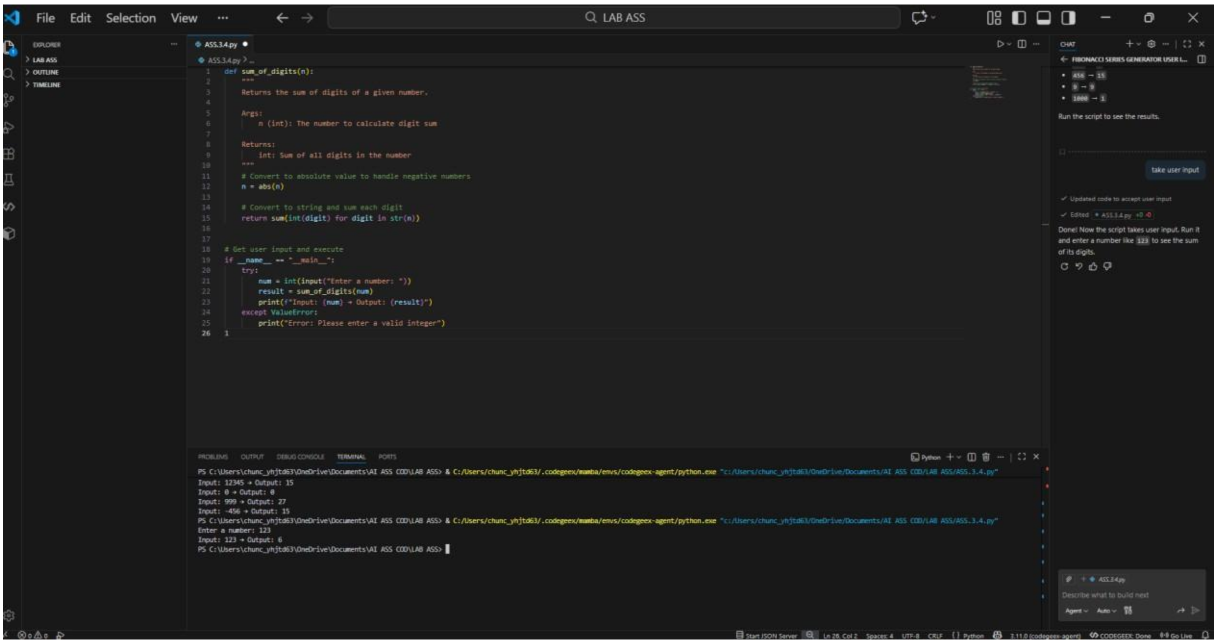
Prompt:

1) # Write a Python function that returns the sum of digits of a given number

2)# Write a Python function that returns the sum of digits of a number # Example:

# Input: 123 → Output: 6

Code and output :



```python
def sum_of_digits(n):
    """
    Returns the sum of digits of a given number.

    Args:
        n (int): The number to calculate digit sum

    Returns:
        int: Sum of all digits in the number
    """
    # Convert to absolute value to handle negative numbers
    n = abs(n)

    # Convert to string and sum each digit
    return sum(int(digit) for digit in str(n))


# Get user input and execute
if __name__ == "__main__":
    try:
        num = int(input("Enter a number: "))
        result = sum_of_digits(num)
        print(f"Input: {num} → Output: {result}")
    except ValueError:
        print("Error: Please enter a valid integer")
1
```

```
PS C:\Users\chunc_yhjtd63\OneDrive\Documents\AI ASS COD\LAB ASS> & C:/Users/chunc_yhjtd63/.codegeex/mamba/envs/codegeex-agent/python.exe "c:/Users/chunc_yhjtd63/OneDrive/Documents/AI ASS COD/LAB ASS/ASS.3.4.py"
Input: 12345 → Output: 15
Input: 0 → Output: 0
Input: 999 → Output: 27
Input: -456 → Output: 15
PS C:\Users\chunc_yhjtd63\OneDrive\Documents\AI ASS COD\LAB ASS> & C:/Users/chunc_yhjtd63/.codegeex/mamba/envs/codegeex-agent/python.exe "c:/Users/chunc_yhjtd63/OneDrive/Documents/AI ASS COD/LAB ASS/ASS.3.4.py"
Enter a number: 123
Input: 123 → Output: 6
PS C:\Users\chunc_yhjtd63\OneDrive\Documents\AI ASS COD\LAB ASS> 
```

- 456 → 15
- 9 → 9
- 1000 → 1

Run the script to see the results.

take user input

✓ Updated code to accept user input

✓ Edited ASS.3.4.py +0 -0

Done! Now the script takes user input. Run it and enter a number like 123 to see the sum of its digits.

ASS.3.4.py

Describe what to build next

Agent Auto