AI ASSI CODING

ASSIGNMENT-3.4

H Ti.NO:2303A510B4

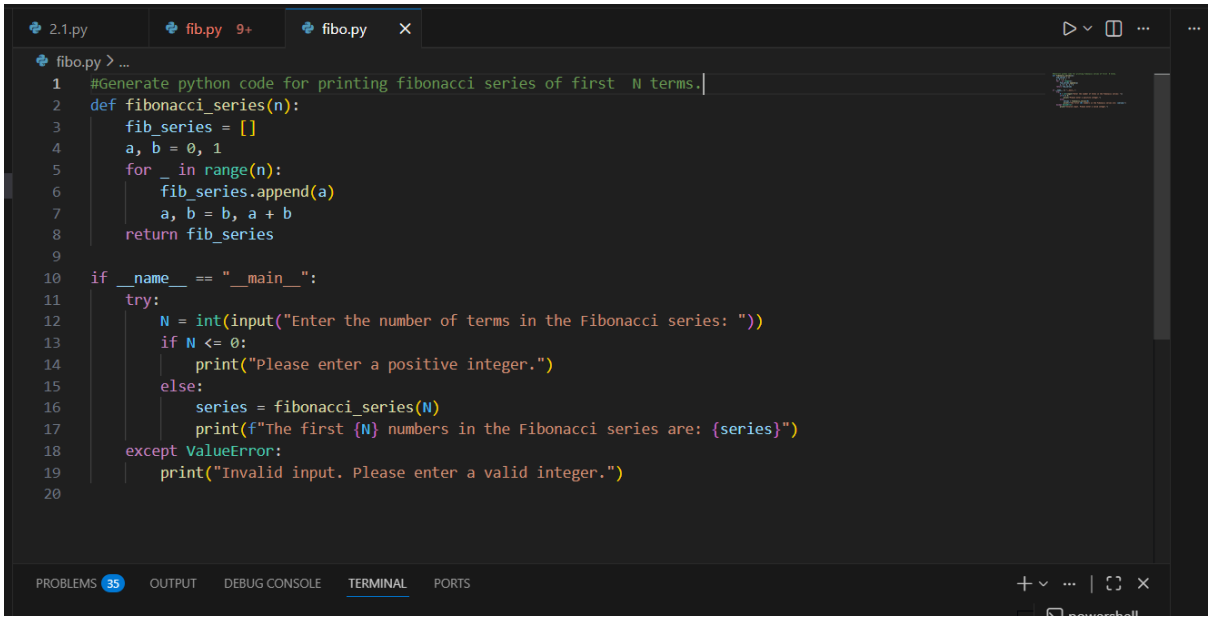
Batch:14

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

Prompt

#Generate python code for printing fibonacci series of first N terms.

Code:



Output:

Justification:

Copilot correctly interpreted the term “Fibonacci numbers” without any examples.

It generated an efficient iterative solution using variable swapping.

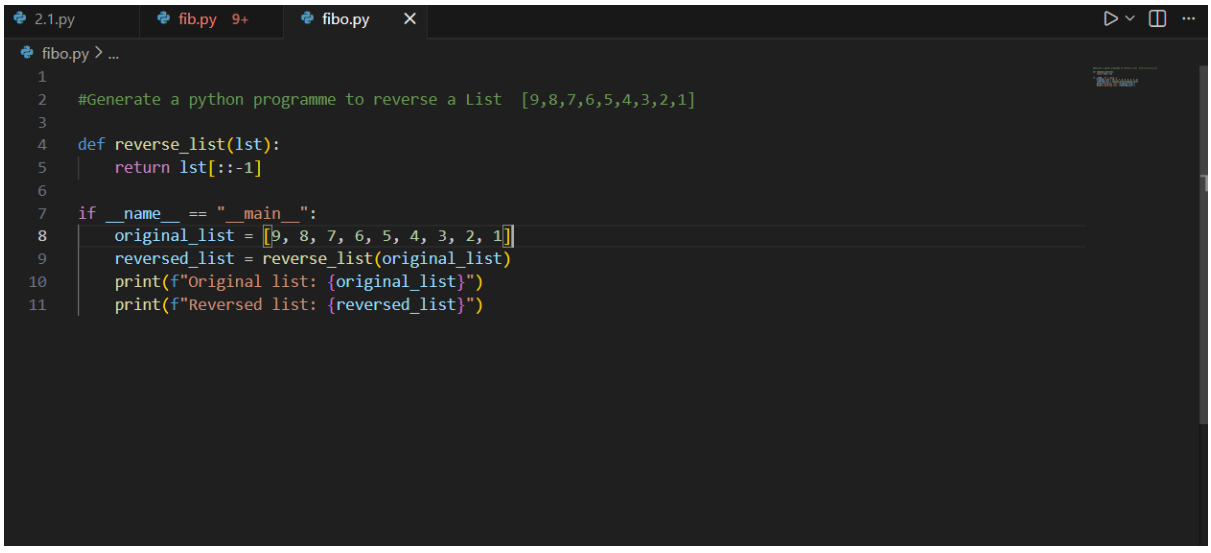
This demonstrates Copilot’s ability to rely on pretrained knowledge in a zero-shot prompting scenario.

Task-2: One-shot Prompt – List Reversal Function

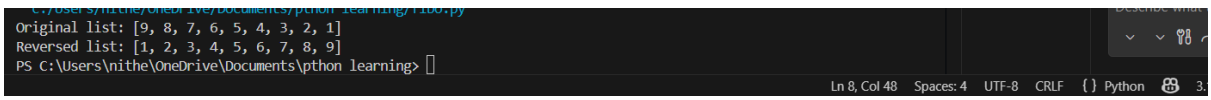
Prompt:

#Generate a python programme to reverse a
List [9,8,7,6,5,4,3,2,1

Code:



Output:



Justification:

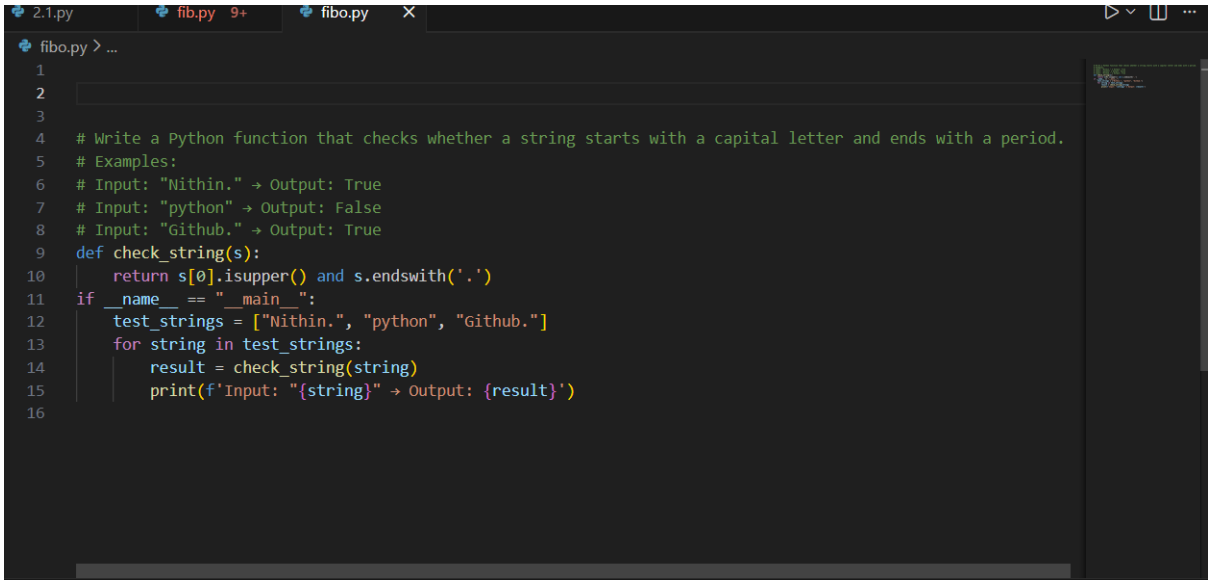
The single example clearly defined the expected input–output behavior.

Copilot accurately inferred the requirement to reverse the list order.

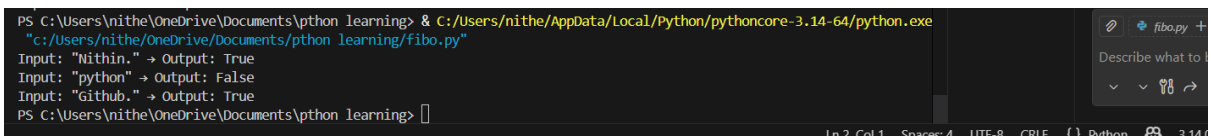
Providing one example improved precision compared to zero-shot prompting.

Task-3: Few-shot Prompt – String Pattern Matching

Prompt:



Output:



Justification:

Multiple examples clearly defined both valid and invalid string patterns.

Copilot inferred the exact conditions using capitalization and ending character.

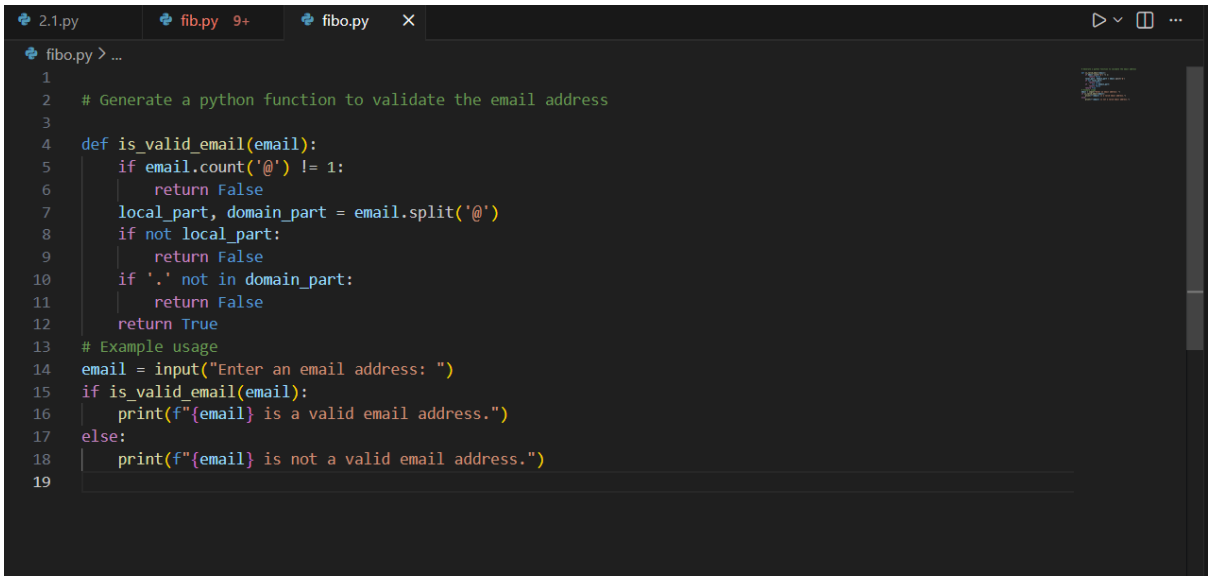
Few-shot prompting improved accuracy by reducing ambiguity compared to zero- or one-shot prompts

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

Prompt:

Zero-shot:

Code:



Output:

```
 "c:/Users/nithe/OneDrive/Documents/pthon learning/fibo.py"
Enter an email address: mohan4gmail.com
mohan4gmail.com is not a valid email address.
PS C:\Users\nithe\OneDrive\Documents\pthon learning> & C:/Users/nithe/AppData/Local/Python/pythoncore-3.14-64/python.exe
 "c:/Users/nithe/OneDrive/Documents/pthon learning/fibo.py"
Enter an email address: nithenk3@gmail.com
nithenk3@gmail.com is a valid email address.
PS C:\Users\nithe\OneDrive\Documents\pthon learning>
```

Few-shot:

Code:

```python
#Generate a python function to validate the email address.
def is_valid_email(email):
    if email.count('@') != 1:
        return False
    local_part, domain_part = email.split('@')
    if not local_part:
        return False
    if '.' not in domain_part:
        return False
    return True
# Example usage
print(is_valid_email("nithenk3@gmail.com"))      # True
print(is_valid_email("mohangmail.com"))       # False
print(is_valid_email("bala@com"))              # False
print(is_valid_email("akhil3@gmail.com"))   # True
```

Output:

```
& C:/Users/nithe/AppData/Local/Python/pythoncore-3.14-64/python.exe "c:/Users/nithe/OneDrive/Documents/pthon learning/fi
bo.py" is not a valid email address.
True
False
False
True
PS C:\Users\nithe\OneDrive\Documents\pthon learning>
```

Justification:

- The zero-shot prompt resulted in a basic validation that only checks for symbols.
- Few-shot examples clarified valid and invalid email structures explicitly.
- Providing examples improved Copilot’s reliability and produced more accurate validation logic.

Task-5: Prompt Tuning – Summing Digits of a Number

Prompt:

Style-1:Generic task Prompt

#generate a python function that returns the sum of digits of a number.

Code:

```python
#generate a python function that returns the sum of digits of a number.
def sum_of_digits(n):
    return sum(int(digit) for digit in str(n))
# Example usage:
number = int(input("Enter a number: "))
result = sum_of_digits(number)
print(f"The sum of digits of {number} is {result}.")
```

Output:

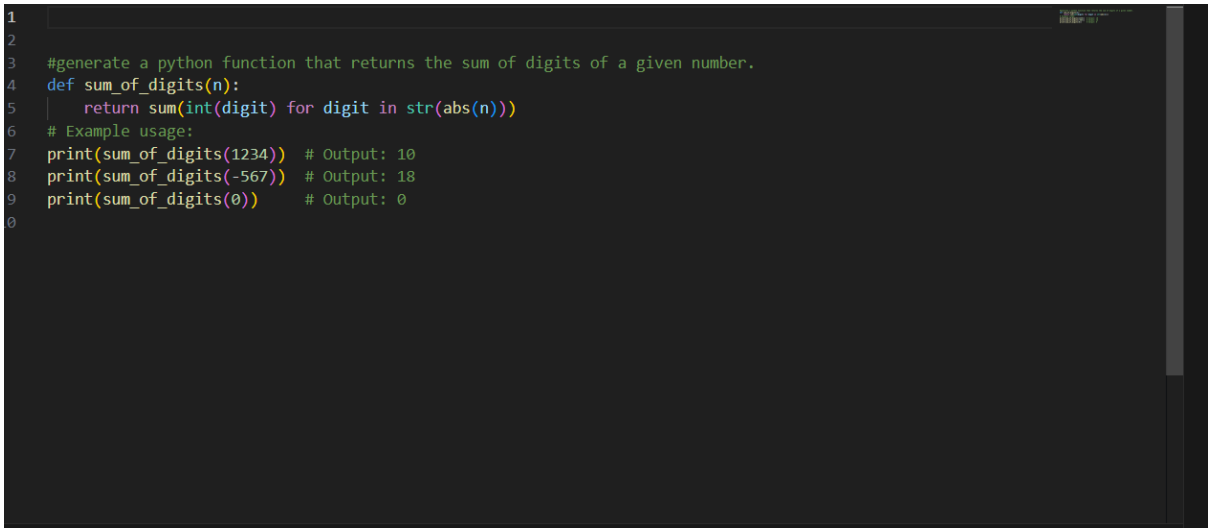
```
PS C:\Users\nithe\OneDrive\Documents\pthon learning> & C:/Users/nithe/AppData/Local/Python/pythoncore-3.14-64/python.exe
 "c:/Users/nithe/OneDrive/Documents/pthon learning/fibo.py"
Enter a number: 2345
The sum of digits of 2345 is 14.
PS C:\Users\nithe\OneDrive\Documents\pthon learning> & C:/Users/nithe/AppData/Local/Python/pythoncore-3.14-64/python.exe
 "c:/Users/nithe/OneDrive/Documents/pthon learning/fibo.py"
Enter a number: 8356
The sum of digits of 8356 is 22.
PS C:\Users\nithe\OneDrive\Documents\pthon learning>
```

Style-2:

Prompt:

generate a python function that returns the sum of digits of a given number.

Code:

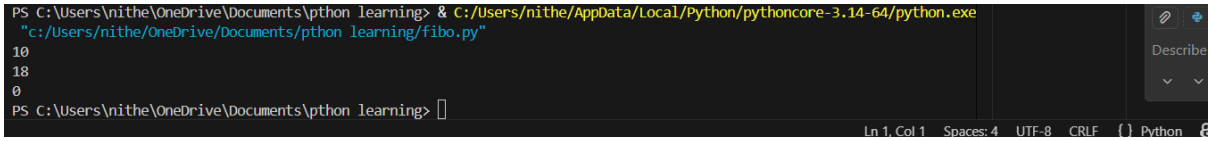


```python

#generate a python function that returns the sum of digits of a given number.
def sum_of_digits(n):
    return sum(int(digit) for digit in str(abs(n)))
# Example usage:
print(sum_of_digits(1234))  # Output: 10
print(sum_of_digits(-567))  # Output: 18
print(sum_of_digits(0))     # Output: 0

```

Output:



```
PS C:\Users\nithe\OneDrive\Documents\pthon learning> & C:/Users/nithe/AppData/Local/Python/pythoncore-3.14-64/python.exe
 "c:/Users/nithe/OneDrive/Documents/pthon learning/fibo.py"
10
18
0
PS C:\Users\nithe\OneDrive\Documents\pthon learning>
```

Justification:

- The generic prompt produced a correct but more verbose loop-based solution.
- Adding an input/output example guided the model toward a cleaner, more concise implementation.
- Prompt tuning with examples improves readability and optimization by clarifying expected behavior.