

AI ASSISTED CODING-LAB ASSIGNMENT

ASSIGNMENT – 3.4

Name: K.Harish

Batch:13

H.T No:2303A51858

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 :

The screenshot shows a Microsoft Visual Studio Code (VS Code) interface. The left sidebar includes the Explorer, File, Edit, Selection, View, and Chat sections. The main area displays a Python script named `ASS3.py`. The code defines a function `print_Fibonacci()` that prints the first `N` Fibonacci numbers. It includes validation for positive integers and handles invalid input. The terminal at the bottom shows the execution of the script and its output.

```
File Edit Selection View < > Q LAB ASS
EXPLORER > LAB ASS > OUTLINE > TIMELINE
... > ASS3.py
  > ASS3.py >
    > #!/usr/bin/env python3
    > #<-- [ ] Add Comment />
    1 def print_Fibonacci(n):
    2     """
    3         Print the first N Fibonacci numbers in order (one line).
    4
    5     Args:
    6         n (int): Number of Fibonacci numbers to print.
    7     """
    8     if n < 0:
    9         print("Please enter a positive number!")
    10    return
    11
    12    fib_numbers = []
    13
    14    if n >= 1:
    15        fib_numbers.append(0)
    16    if n >= 2:
    17        fib_numbers.append(1)
    18
    19    # Generate Fibonacci numbers
    20    for i in range(2, n):
    21        fib_numbers.append(fib_numbers[i-1] + fib_numbers[i-2])
    22
    23    print(*join(map(str, fib_numbers)))
    24
    25
    26    # Get user input and execute
    27    if __name__ == "__main__":
    28        try:
    29            num = int(input("n: "))
    30            print_Fibonacci(num)
    31        except ValueError:
    32            print("Invalid input! Please enter a valid integer.")
    33
PS C:\Users\chunc_yh\lab\ASS1> cd "C:\Users\chunc_yh\OneDrive\Documents\AT ASS COD\LAB ASS" & C:\Users\chunc_yh\lab\ASS1> codegees\matlab\envs\codegees-agent\python.exe "c:\users\chunc_yh\lab\ASS1\OneDrive\Documents\AT ASS COD\LAB ASS\ASS3.py"
KeyboardInterrupt
PS C:\Users\chunc_yh\lab\ASS1> cd "C:\Users\chunc_yh\OneDrive\Documents\AT ASS COD\LAB ASS" & C:\Users\chunc_yh\lab\ASS1> codegees\matlab\envs\codegees-agent\python.exe "c:\users\chunc_yh\lab\ASS1\OneDrive\Documents\AT ASS COD\LAB ASS\ASS3.py"
N: 7
0 1 1 2 3 5 8
PS C:\Users\chunc_yh\lab\ASS1>
```

Output from the terminal:

```
PS C:\Users\chunc_yh\lab\ASS1> cd "C:\Users\chunc_yh\OneDrive\Documents\AT ASS COD\LAB ASS" & C:\Users\chunc_yh\lab\ASS1> codegees\matlab\envs\codegees-agent\python.exe "c:\users\chunc_yh\lab\ASS1\OneDrive\Documents\AT ASS COD\LAB ASS\ASS3.py"
Describe what to build next
Agent > Auto > >
```

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 :

```

File Edit Selection View ... < > Q LAB ASS
EXPLORER > LAB ASS > OUTLINE > TIMELINE
ASS3.4.py
# Explain | Add Comment | ...
1 def reverse_list(lst):
2     """Reverse a list using the reverse() method."""
3     lst.reverse()
4     return lst
5
6     # Take user input
7     try:
8         user_input = input("Enter list elements separated by spaces: ")
9         lst = list(map(int, user_input.split()))
10
11     # Reverse and print
12     reversed_lst = reverse_list(lst)
13     print(reversed_lst)
14 except ValueError:
15     print("Error: Please enter valid numbers separated by spaces")
16

```

CHAT

i will provide once 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 → ASS3.4.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.

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

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

PS C:\Users\chunc_yhjtd63\OneDrive\Documents\AI ASS COD\LAB ASS\ASS.3.4.py>
Reversed list: [3, 2, 1]
PS C:\Users\chunc_yhjtd63\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>

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 :

```

File Edit Selection View ... LAB ASS ASS3App X
ASS3App.py [Add Comment] X
1 def is_valid(s):
2     """
3         Check whether a string starts with a capital letter and ends with a period.
4
5     Args:
6         s (str): The string to validate
7
8     Returns:
9         bool: True if starts with capital letter and ends with period, False otherwise
10
11    if len(s) < 2:
12        return False
13
14    return s[0].isupper() and s[-1] == '.'
15
16
17 # Test cases
18 if __name__ == "__main__":
19     tests = [
20         "Hello.",
21         "Hello",
22         "Hello!",
23         ".",
24         "python",
25         "python.",
26         "a",
27         "a."
28     ]
29
30     print("Testing is_valid() function!")
31     for test in tests:
32         result = is_valid(test)
33         print(f"Input: {test} -> Output: {result}")
34

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```

PS C:\Users\chunc\OneDrive\Documents\AT ASS CODILAB ASS> & C:/Users/chunc_jh/t063/.codegen/media/envs/codigos-agent/python.exe "c:/users/chunc_jh/t063/IsValid.py"
Input: "Hello." -> Output: False
Input: "Hello" -> Output: False
Input: "Hello!" -> Output: False
Input: "." -> Output: False
Input: "python" -> Output: True
Input: "python." -> Output: False
Input: "a" -> Output: False
Input: "a." -> Output: False
PS C:\Users\chunc_jh/t063\OneDrive\Documents\AT ASS CODILAB ASS>

```

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 ... < > Q LAB ASS
ASS3Assy X
ASS3Assy.py ->
def is_valid_email(email):
    # Check if domain part has at least one dot and parts are not empty
    if '.' not in email:
        return False
    domain_parts = email.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",
        "user@exampl.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}")

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```

PS C:\Users\chunc_yh\1083\OneDrive\Documents\AT ASS\CODE\LAB ASS> & C:\Users\chunc_yh\1083\codegen\agents\envs\college-agent\python.exe "c:/Users/chunc_yh\1083\OneDrive\Documents\AT ASS\CODE\LAB ASS\ASS3Assy.py"
Input: "user@example.co.uk" -> Output: True
Input: "john.doe@company.org" -> Output: True
Input: "user@domain.co.uk" -> Output: True
Input: "user@" -> Output: False
Input: "user@.name@example.com" -> Output: True
Input: "user.name@example.com" -> Output: True
Input: "valid_email@test.com" -> Output: True

```

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:

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

Example:

Input: 123 → Output: 6

Code and output :

