

AI ASSISTED CODING-LAB ASSIGNMENT

ASSIGNMENT – 3.4

Name: .D.Nithin
H NO: 2303A51845

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 :

```
ASS3.4.py
# Write a Python function that takes an integer N and prints the first N Fibonacci numbers in order
def print_fibonacci(n):
    """Print the first n Fibonacci numbers in order (on one line)."""
    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)
    for i in range(2, n):
        fib_numbers.append(fib_numbers[i-1] + fib_numbers[i-2])
    print(" ".join(map(str, fib_numbers)))
if __name__ == "__main__":
    try:
        num = int(input("N: "))
        print_fibonacci(num)
    except ValueError:
        print("Invalid input! Please enter a valid integer.")
```

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

Done! I've created a Python function `print_fibonacci()` 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 (e.g., 10) to generate that many Fibonacci terms.

In one line output

Read Python file

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

Start JSON Server | Le 28, Col 1 | Spaces: 4 | LF-F | Python | 3.11.0 (codegen-agent) | CODEGENX | Go Live

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 ... 🔍 LAB ASS ASS3.4.py ... 🔍 Explain | Add Comment | X
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()))
except ValueError:
    print("Error: Please enter valid numbers separated by spaces")

# Reverse and print
reversed_lst = reverse_list(lst)
print(reversed_lst)
```

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

✓ Read Python file

Describe what to build next

Agent ✓ Auto ➔

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 ... Q LAB ASS
File Explorer Search Outline Timeline
ASS34.py X
ASS34.py + Add Comment
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",
        "Python",
        "python",
        ".",
        "a",
        ""
    ]
    print("Testing is_valid() function")
    for test in test_cases:
        result = is_valid(test)
        print(f"Input: '{test}' Output: {result}")

```

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 and ends with a period.

- 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
ASS34.py ...
def is_valid_email(email):
    # Check if email has at least one dot and parts are not empty
    if '.' not in email:
        return False

    # Check if domain has valid structure (e.g., example.com)
    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@example.com",
        "user@com",
        "user@gmail.co.uk",
        "john.doe@company.org",
        "user@.co.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_01\OneDrive\Documents\AI ASS CD\LAB ASS-> C:\Users\chunc_01\OneDrive\Documents\mathieu\envs\codegen-agent\python.exe "c:/users/chunc_01\OneDrive\Documents\AI ASS CD\LAB ASS\ASS34.py"

Input: "user@domain.co.uk" -> Output: True
Input: "john.doe@company.org" -> Output: True
Input: "user@.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_01\OneDrive\Documents\AI ASS CD\LAB ASS->

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 :

The screenshot shows a code editor interface with a Python script named `ASS3.4.py` open in the main pane. The script contains a function `sum_of_digits(n)` that calculates the sum of digits of a given number. It includes handling for negative numbers and user input validation. The terminal below shows the execution of the script and its output for various inputs.

```
File Edit Selection View ... Q LAB ASS
EXPLORER > LAB ASS > OUTLINE > TIMELINE
ASS3.4.py
# 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

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
Python + - | ↻
PS C:\Users\chunc_yh\OneDrive\Documents\AI ASS COD\LAB ASS> & C:/Users/chunc_yh/Downloads/codegeex-agent/python.exe "c:/Users/chunc_yh/OneDrive/Documents/AI ASS COD/LAB ASS/ASS3.4.py"
Input: 12345 + Output: 15
Input: -12345 + Output: 15
Input: 999 + Output: 27
Input: -456 + Output: 15
PS C:\Users\chunc_yh\OneDrive\Documents\AI ASS COD\LAB ASS> & C:/Users/chunc_yh/Downloads/codegeex-agent/python.exe "c:/Users/chunc_yh/OneDrive/Documents/AI ASS COD/LAB ASS/ASS3.4.py"
Enter a number: 123
Input: 123 + Output: 6
PS C:\Users\chunc_yh\OneDrive\Documents\AI ASS COD\LAB ASS>
```

