

AI ASSISTED CODING

ASSIGNMENT-3.4

Name: V. Vyshnavi

Hallticket No.:2303A51968

Batch-14

Lab 4: Advanced Prompt Engineering – Zero-shot, One-shot, and Few-shot Techniques

Task 1: Zero-shot Prompt – Fibonacci Series Generator

Task 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.

Expected Output #1

- A complete Python function generated by Copilot without any example provided.
- Correct output for sample input $N = 7 \rightarrow 0\ 1\ 1\ 2\ 3\ 5\ 8$
- Observation on how Copilot understood the instruction with zero context.

Prompt:

Write a Python program that takes an integer N from user input and prints the first N Fibonacci numbers starting from 0

Code and output:

```
1 # Write a Python program that takes an integer N from user input and prints the
2
3 n = int(input("Enter the number of Fibonacci numbers to print: "))
4 a, b = 0, 1
5 for i in range(n):
6     print(a)
7     a, b = b, a + b
```

```
PS C:\Users\Vyshnavi\AppData\Local\Programs\Microsoft VS Code> & C:\Users\Vyshnavi\AppData\Local\Programs\Python\Python313\python.exe c:/Users/Vyshnavi/AppData/Local/Programs/Thonny/Lib/site-packages/jedi/third_party/typeshed/stdlib/3/2303a51968.py
Enter the number of Fibonacci numbers to print: 5
0
1
1
2
3
PS C:\Users\Vyshnavi\AppData\Local\Programs\Microsoft VS Code>
```

Explanation:

The Fibonacci series is a sequence of numbers where each number is the sum of the two preceding numbers, starting from 0 and 1.

In this program, the user enters a number **N**, and the program generates and prints the first **N Fibonacci numbers**.

The program initializes the first two Fibonacci values and uses a loop to calculate subsequent numbers and display them in a single line separated by spaces.

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.

Expected Output #2

- Copilot-generated function to reverse a list using slicing or loop.
- Output: [3, 2, 1] for input [1, 2, 3]
- Observation on how adding a single example improved Copilot's accuracy.

Prompt:

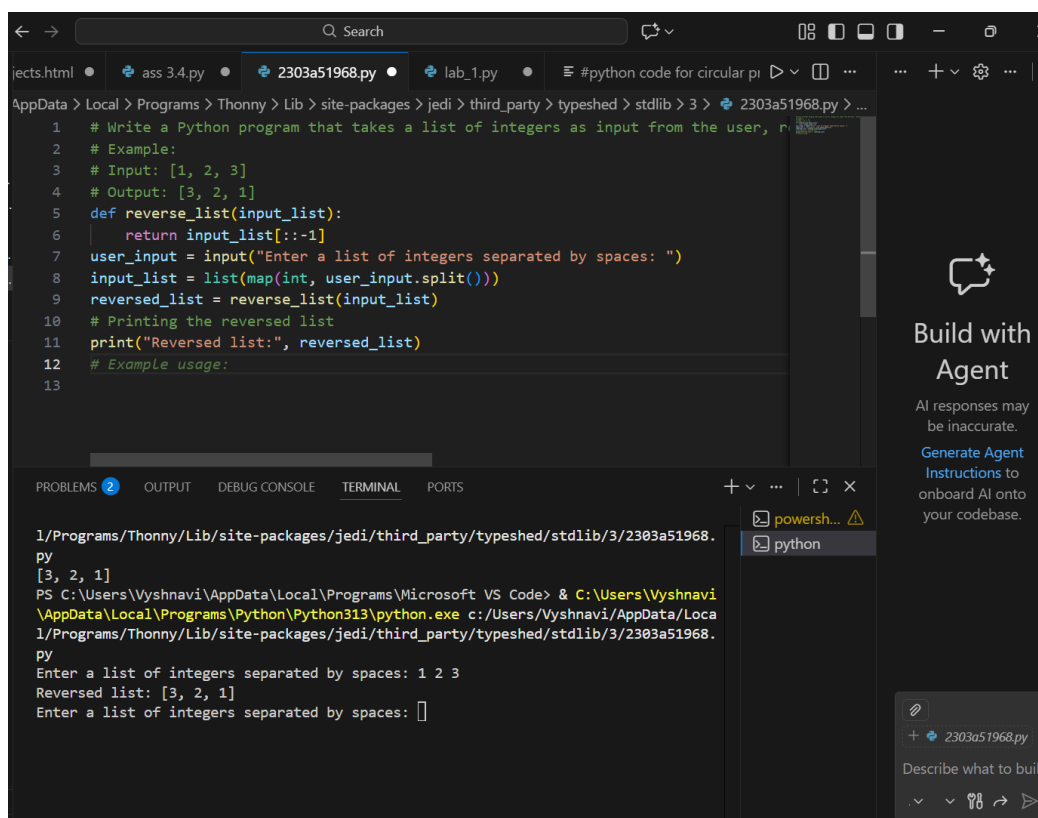
#Write a Python program that takes a list of integers as input from the user, reverses the list using a function (slicing or loop), and prints the reversed list.

Example:

Input: [1, 2, 3]

Output: [3, 2, 1]

Code and output:



The screenshot shows a Python IDE with a file named `2303a51968.py`. The code defines a function `reverse_list` that takes a list and returns its reverse using slicing. It then prompts the user for input, converts it to a list of integers, calls the function, and prints the result. The terminal output shows the program running successfully, taking the input `1 2 3` and outputting `[3, 2, 1]`.

```
1 # Write a Python program that takes a list of integers as input from the user, r
2 # Example:
3 # Input: [1, 2, 3]
4 # Output: [3, 2, 1]
5 def reverse_list(input_list):
6     return input_list[::-1]
7 user_input = input("Enter a list of integers separated by spaces: ")
8 input_list = list(map(int, user_input.split()))
9 reversed_list = reverse_list(input_list)
10 # Printing the reversed list
11 print("Reversed list:", reversed_list)
12 # Example usage:
13
```

```
1/Programs/Thonny/Lib/site-packages/jedi/third_party/typeshed/stdlib/3/2303a51968.
py
[3, 2, 1]
PS C:\Users\Vyshnavi\AppData\Local\Programs\Microsoft VS Code> & C:\Users\Vyshnavi
\AppData\Local\Programs\Python\Python313\python.exe c:/Users/Vyshnavi/AppData/Loca
1/Programs/Thonny/Lib/site-packages/jedi/third_party/typeshed/stdlib/3/2303a51968.
py
Enter a list of integers separated by spaces: 1 2 3
Reversed list: [3, 2, 1]
Enter a list of integers separated by spaces: 
```

Explanation:

- input() takes the list from the user as a string.
- eval() converts the input string into a Python list.
- The list is stored in the variable lst.
- lst[::-1] reverses the list using slicing.
- 1 as the step value means the list is read from last to first.
- print() displays the reversed list as output.

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.

Expected Output #3

- A function `is_valid()` that checks the pattern.
- Output: True or False based on input.
- Students reflect on how multiple examples guide Copilot to generate more accurate code.

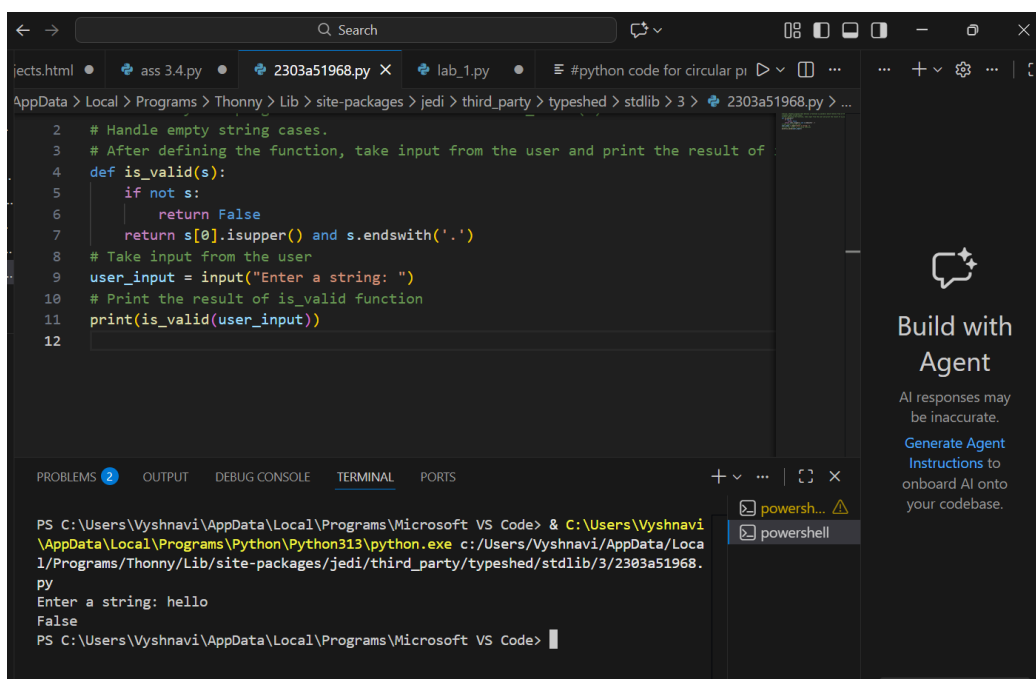
Prompt:

Write a Python program that defines a function `is_valid(s)` which returns True if the string starts with a capital letter and ends with a period.

Handle empty string cases.

After defining the function, take input from the user and print the result of `is_valid(s)`.

Code and output:



The screenshot shows a VS Code editor window with a Python file named `2303a51968.py`. The code defines a function `is_valid(s)` that checks if a string starts with a capital letter and ends with a period. It also takes user input and prints the result of the function.

```
2 # Handle empty string cases.
3 # After defining the function, take input from the user and print the result of
4 def is_valid(s):
5     if not s:
6         return False
7     return s[0].isupper() and s.endswith('.')
8 # Take input from the user
9 user_input = input("Enter a string: ")
10 # Print the result of is_valid function
11 print(is_valid(user_input))
12
```

The terminal output shows the execution of the script. The user enters "hello", and the output is "False".

```
PS C:\Users\Vyshnavi\AppData\Local\Programs\Microsoft VS Code> & C:\Users\Vyshnavi\AppData\Local\Programs\Python\Python313\python.exe c:/Users/Vyshnavi/AppData/Local/Programs/Thonny/Lib/site-packages/jedi/third_party/typeshed/stdlib/3/2303a51968.py
Enter a string: hello
False
PS C:\Users\Vyshnavi\AppData\Local\Programs\Microsoft VS Code>
```

Explanation:

- The function `is_valid(s)` checks whether a given string satisfies two conditions.
- First, it checks if the string is empty. If it is empty, the function returns `False` because an empty string cannot start with a capital letter or end with a period.
- The function then checks the first character of the string using `isupper()` to verify that it starts with a capital letter.
- Next, it uses the `endswith('.')` method to check whether the string ends with a period.
- Both conditions are combined using the logical `and` operator.
- If both conditions are true, the function returns `True`; otherwise, it returns `False`.
- After defining the function, the program takes input from the user.
- Finally, it prints the result returned by the `is_valid(s)` function.

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.

Expected Output #4

- Compare both outputs:

Zero-shot may result in basic or generic validation.

Few-shot gives detailed and specific logic (e.g., `@` and domain checking).

- Submit both code versions and note how few-shot improves reliability

Prompt:

Write a Python function to validate an email address.

Use regular expressions to check the email format.

The function should return `True` if the email is valid, otherwise `False`.

Take the email address as input from the user and print the result.

Code and output:

```

1 # Write a Python function to validate an email address.
2 # Use regular expressions to check the email format.
3 # The function should return True if the email is valid, otherwise False.
4 # Take the email address as input from the user and print the result.
5 import re
6 def validate_email(email):
7     # Define the regular expression for validating an email
8     email_regex = r'^[a-zA-Z0-9._%+-]+@[a-zA-Z0-9.-]+\.[a-zA-Z]{2,}$'
9
10    # Use re.match to check if the email matches the regex
11    if re.match(email_regex, email):
12        return True
13    else:
14        return False
15
16    # Take email input from the user
17    user_email = input("Enter an email address to validate: ")
18    # Validate the email and print the result
19    if validate_email(user_email):
20        print("The email address is valid.")
21    else:
22        print("The email address is invalid.")

```

```

PS C:\Users\Vyshnavi\AppData\Local\Programs\Microsoft VS Code> & C:\Users\Vyshnavi\AppData\Local\Programs\Python\Python313\python.exe c:/Users/Vyshnavi/AppData/Local/Programs/Thonny/Lib/site-packages/jedi/third_party/typeshed/stdlib/3/2303a51968.py
PS C:\Users\Vyshnavi\AppData\Local\Programs\Microsoft VS Code> & C:\Users\Vyshnavi\AppData\Local\Programs\Python\Python313\python.exe c:/Users/Vyshnavi/AppData/Local/Programs/Thonny/Lib/site-packages/jedi/third_party/typeshed/stdlib/3/2303a51968.py
Enter an email address to validate: vuggevyshnavi@gmail.com
The email address is valid.
PS C:\Users\Vyshnavi\AppData\Local\Programs\Microsoft VS Code>

```

Explanation:

- The re module is used for email pattern matching.
- The function is `_valid_email()` checks whether the email format is valid.
- A regular expression defines the correct email structure.
- `re.match()` verifies the email against the pattern.
- The function returns **True** for a valid email and **False** otherwise.
- The email is taken as input and the result is printed.

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

Expected Output #5

- Two versions of the `sum_of_digits()` function.
- Example Output: `sum_of_digits(123) → 6`

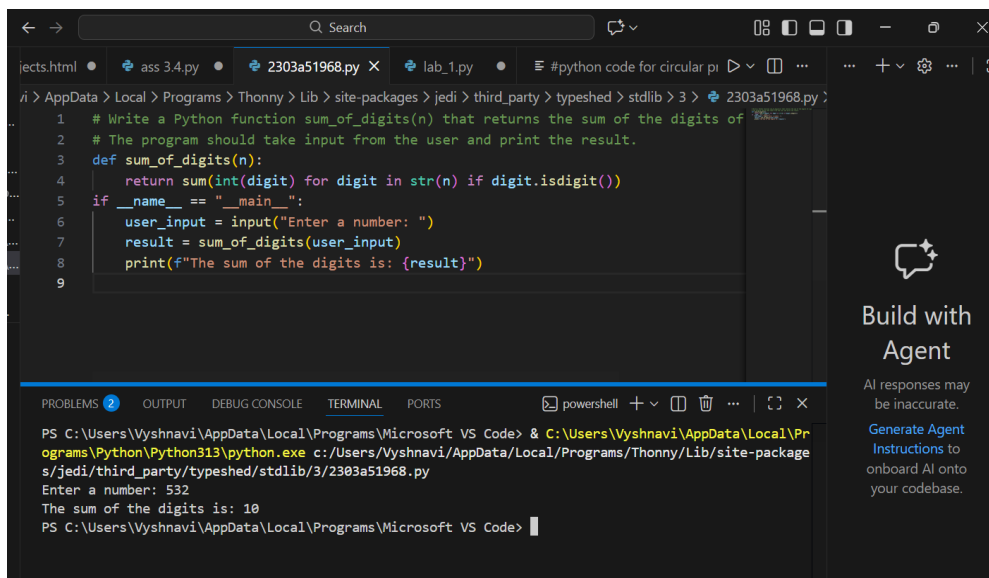
- Short analysis: which prompt produced cleaner or more optimized code and why?

Prompt:

Write a Python function sum_of_digits(n) that returns the sum of the digits of a given number.

The program should take input from the user and print the result.

Code and output:



The screenshot shows a VS Code editor window with a Python file named `2303a51968.py`. The code defines a function `sum_of_digits(n)` that calculates the sum of digits in a number. The program takes user input and prints the result. The terminal output shows the program being executed, the user entering `532`, and the program outputting `The sum of the digits is: 10`.

```
i /i > AppData > Local > Programs > Thonny > Lib > site-packages > jedi > third_party > typeshed > stdlib > 3 > 2303a51968.py
1 # Write a Python function sum_of_digits(n) that returns the sum of the digits of
2 # The program should take input from the user and print the result.
3 def sum_of_digits(n):
4     return sum(int(digit) for digit in str(n) if digit.isdigit())
5 if __name__ == "__main__":
6     user_input = input("Enter a number: ")
7     result = sum_of_digits(user_input)
8     print(f"The sum of the digits is: {result}")
9
```

PROBLEMS 2 OUTPUT DEBUG CONSOLE TERMINAL PORTS powershell + - X

PS C:\Users\Vyshnavi\AppData\Local\Programs\Microsoft VS Code> & C:\Users\Vyshnavi\AppData\Local\Programs\Python\Python313\python.exe c:/Users/Vyshnavi/AppData/Local/Programs/Thonny/Lib/site-packages/jedi/third_party/typeshed/stdlib/3/2303a51968.py

Enter a number: 532

The sum of the digits is: 10

PS C:\Users\Vyshnavi\AppData\Local\Programs\Microsoft VS Code>

Explanation:

- The function `sum_of_digits()` calculates the total of all digits in a given number.
- The number is taken as input from the user.
- Each digit is extracted using modulo (%) and added to a sum variable.
- Integer division (//) is used to remove the last digit after each step.
- Finally, the total sum of digits is printed.