Lab Assignment 4.2

Course: AI - Assisted Coding

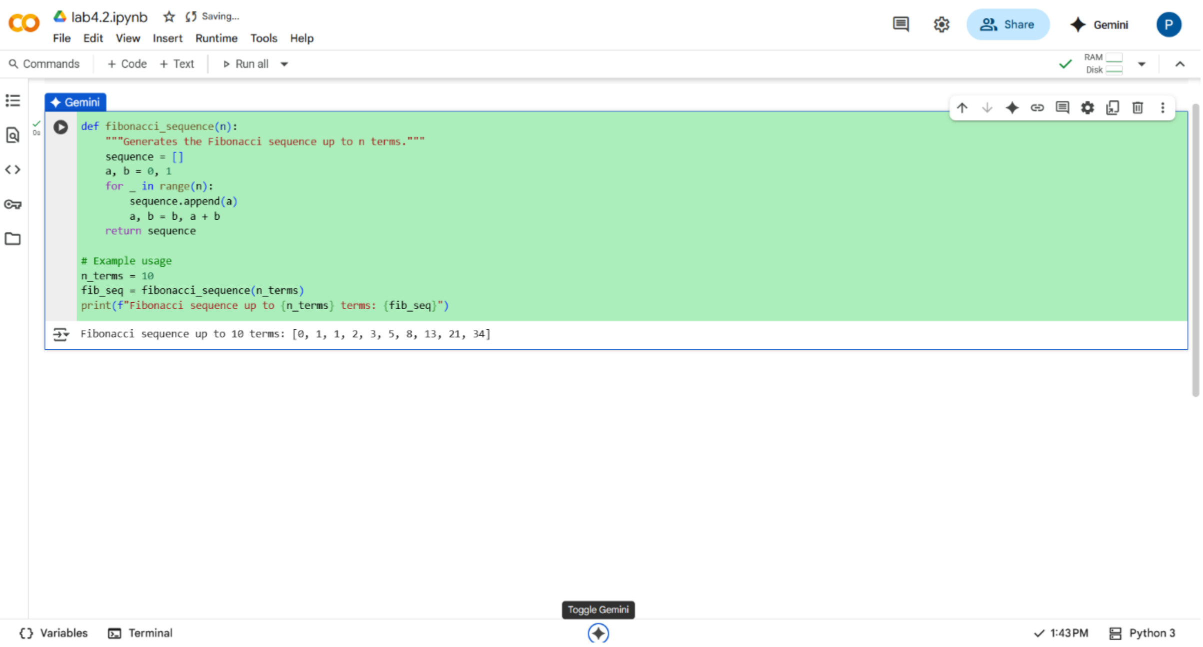
Name :P.Srikar

Htno :2403A52002

Task-1:

Prompt: Generate a Python function to generate the Fibonacci sequence up to n terms .The number should be taken as user input.

Code&Output:



```python
def fibonacci_sequence(n):
    """Generates the Fibonacci sequence up to n terms."""
    sequence = []
    a, b = 0, 1
    for _ in range(n):
        sequence.append(a)
        a, b = b, a + b
    return sequence

# Example usage
n_terms = 10
fib_seq = fibonacci_sequence(n_terms)
print(f"Fibonacci sequence up to {n_terms} terms: {fib_seq}")
```

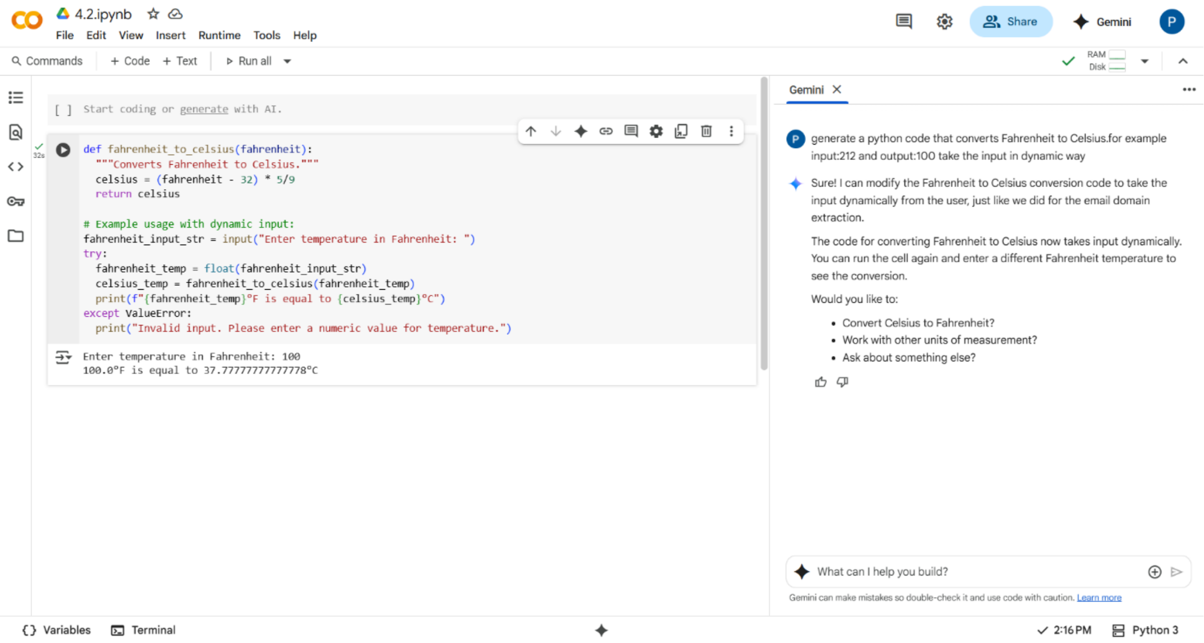
```
Fibonacci sequence up to 10 terms: [0, 1, 1, 2, 3, 5, 8, 13, 21, 34]
```

Observation: The zero-shot prompt asked the AI to generate a Python function for the Fibonacci sequence using only the instruction. The code works correctly and gives the sequence up to n terms. This shows that zero-shot prompting is useful for simple and well-known problems. However, for more complex tasks, giving examples (few-shot prompting) would guide the AI better and produce more accurate results.

Task-2:

Prompt: Generate a Python function to converts Fahrenheit to Celsius. Provide one example: Input: 100, Output: 37.78.

Code&Output:



```
[ ] Start coding or generate with AI.
```

```python
def fahrenheit_to_celsius(fahrenheit):
  """Converts Fahrenheit to Celsius."""
  celsius = (fahrenheit - 32) * 5/9
  return celsius

# Example usage with dynamic input:
fahrenheit_input_str = input("Enter temperature in Fahrenheit: ")
try:
  fahrenheit_temp = float(fahrenheit_input_str)
  celsius_temp = fahrenheit_to_celsius(fahrenheit_temp)
  print(f"{fahrenheit_temp}°F is equal to {celsius_temp}°C")
except ValueError:
  print("Invalid input. Please enter a numeric value for temperature.")
```

```
Enter temperature in Fahrenheit: 100
100.0°F is equal to 37.77777777777778°C
```

generate a python code that converts Fahrenheit to Celsius.for example input:212 and output:100 take the input in dynamic way

Sure! I can modify the Fahrenheit to Celsius conversion code to take the input dynamically from the user, just like we did for the email domain extraction.

The code for converting Fahrenheit to Celsius now takes input dynamically. You can run the cell again and enter a different Fahrenheit temperature to see the conversion.

Would you like to:

- Convert Celsius to Fahrenheit?
- Work with other units of measurement?
- Ask about something else?

What can I help you build?

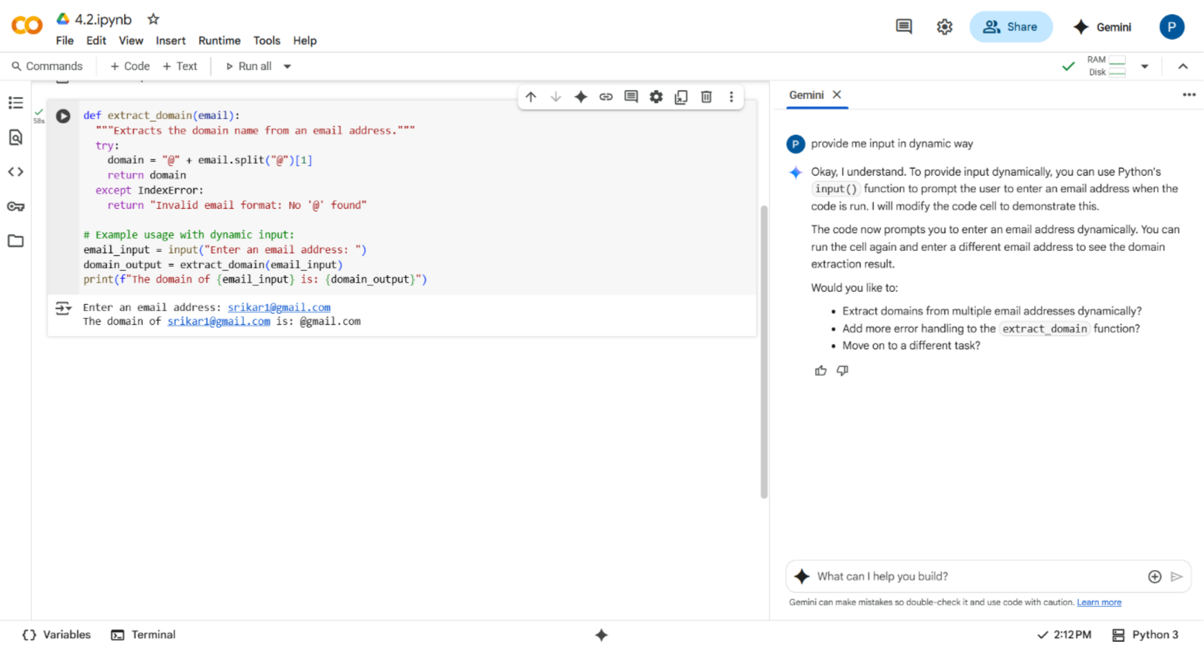
Gemini can make mistakes so double-check it and use code with caution. Learn more

Observation: Using a one-shot prompt, the AI was given a single example (Input: 100 → Output: 37.78). This guided the AI to understand both the formula and the expected output format. Compared to zero-shot prompting, one-shot makes the result more accurate and consistent because the AI learns from at least one demonstration. For simple problems, zero-shot may work, but one-shot improves reliability by showing the AI exactly what is expected.

Task-3:

Prompt: Generate a Python function to extracts the domain name from an email address. for example input:srikar1@gmail.com and output :@gmail.com , input : ram1@hotmail.com and output : @hotmail.com , input: pardhu1@yahoo.in and output : @yahoo.in.

Code&Output:



4.2.ipynb
File Edit View Insert Runtime Tools Help
Commands + Code + Text Run all
Share
Gemini
RAM
Disk

```python
def extract_domain(email):
  """Extracts the domain name from an email address."""
  try:
    domain = "@" + email.split("@")[1]
    return domain
  except IndexError:
    return "Invalid email format: No '@' found"

# Example usage with dynamic input:
email_input = input("Enter an email address: ")
domain_output = extract_domain(email_input)
print(f"The domain of {email_input} is: {domain_output}")
```

```
Enter an email address: srikar1@gmail.com
The domain of srikar1@gmail.com is: @gmail.com
```

Gemini

provide me input in dynamic way

Okay, I understand. To provide input dynamically, you can use Python's `input()` function to prompt the user to enter an email address when the code is run. I will modify the code cell to demonstrate this.

The code now prompts you to enter an email address dynamically. You can run the cell again and enter a different email address to see the domain extraction result.

Would you like to:

- Extract domains from multiple email addresses dynamically?
- Add more error handling to the `extract_domain` function?
- Move on to a different task?

What can I help you build?

Gemini can make mistakes so double-check it and use code with caution. Learn more

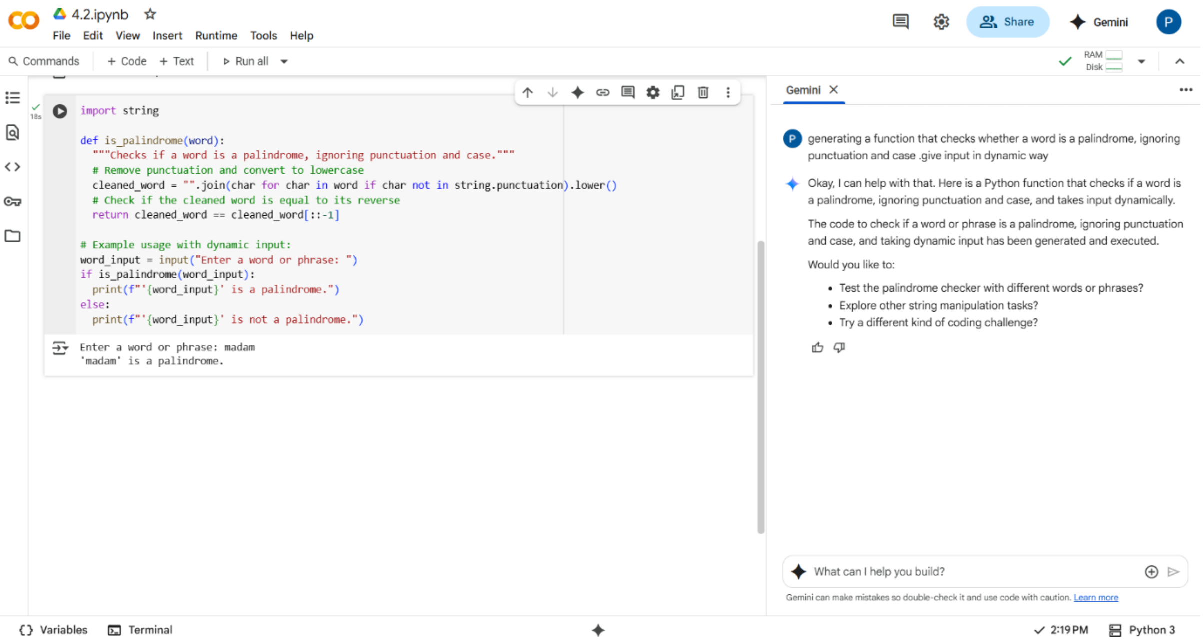
Observation: The program successfully extracts the domain name from an email address by locating the @ symbol and returning the remaining part of the string. It works correctly for different inputs like Gmail, Hotmail, and Yahoo. This shows how string manipulation in Python can be used to solve real-world problems such as processing and validating email address.

Task-4:

Prompt-1:

ZERO-SHOT: Write a Python function that checks whether a given word is a palindrome. The function should ignore punctuation, spaces, and letter case.

Code&Output:



```
import string

def is_palindrome(word):
  """Checks if a word is a palindrome, ignoring punctuation and case."""
  # Remove punctuation and convert to lowercase
  cleaned_word = "".join(char for char in word if char not in string.punctuation).lower()
  # Check if the cleaned word is equal to its reverse
  return cleaned_word == cleaned_word[::-1]

# Example usage with dynamic input:
word_input = input("Enter a word or phrase: ")
if is_palindrome(word_input):
  print(f"'{word_input}' is a palindrome.")
else:
  print(f"'{word_input}' is not a palindrome.")
```

```
Enter a word or phrase: madam
'madam' is a palindrome.
```

Gemini

generating a function that checks whether a word is a palindrome, ignoring punctuation and case .give input in dynamic way

Okay, I can help with that. Here is a Python function that checks if a word is a palindrome, ignoring punctuation and case, and takes input dynamically.

The code to check if a word or phrase is a palindrome, ignoring punctuation and case, and taking dynamic input has been generated and executed.

Would you like to:

- Test the palindrome checker with different words or phrases?
- Explore other string manipulation tasks?
- Try a different kind of coding challenge?

What can I help you build?

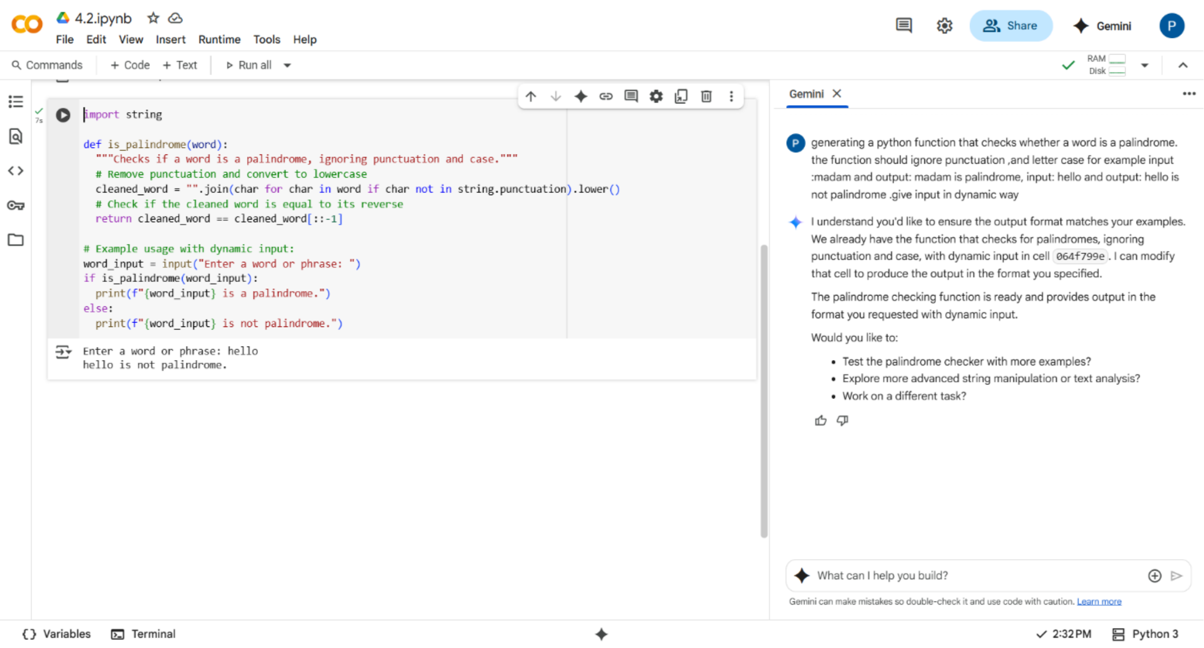
Gemini can make mistakes so double-check it and use code with caution. Learn more

Prompt-2:

FEW-SHOT:

Write a Python function that checks whether a given word is a palindrome. The function should ignore punctuation, spaces, and letter case for example input: madam and output: madam is palindrome ,input : hello and output: hello is not palindrome , input: racecar and output : racecar is palindrome.

Code&Output:



```python
import string

def is_palindrome(word):
  """Checks if a word is a palindrome, ignoring punctuation and case."""
  # Remove punctuation and convert to lowercase
  cleaned_word = "".join(char for char in word if char not in string.punctuation).lower()
  # Check if the cleaned word is equal to its reverse
  return cleaned_word == cleaned_word[::-1]

# Example usage with dynamic input:
word_input = input("Enter a word or phrase: ")
if is_palindrome(word_input):
  print(f"{word_input} is a palindrome.")
else:
  print(f"{word_input} is not palindrome.")
```

```
Enter a word or phrase: hello
hello is not palindrome.
```

generating a python function that checks whether a word is a palindrome. the function should ignore punctuation ,and letter case for example input :madam and output: madam is palindrome, input: hello and output: hello is not palindrome .give input in dynamic way

I understand you'd like to ensure the output format matches your examples. We already have the function that checks for palindromes, ignoring punctuation and case, with dynamic input in cell `064f799e`. I can modify that cell to produce the output in the format you specified.

The palindrome checking function is ready and provides output in the format you requested with dynamic input.

Would you like to:

- Test the palindrome checker with more examples?
- Explore more advanced string manipulation or text analysis?
- Work on a different task?

Gemini can make mistakes so double-check it and use code with caution. Learn more

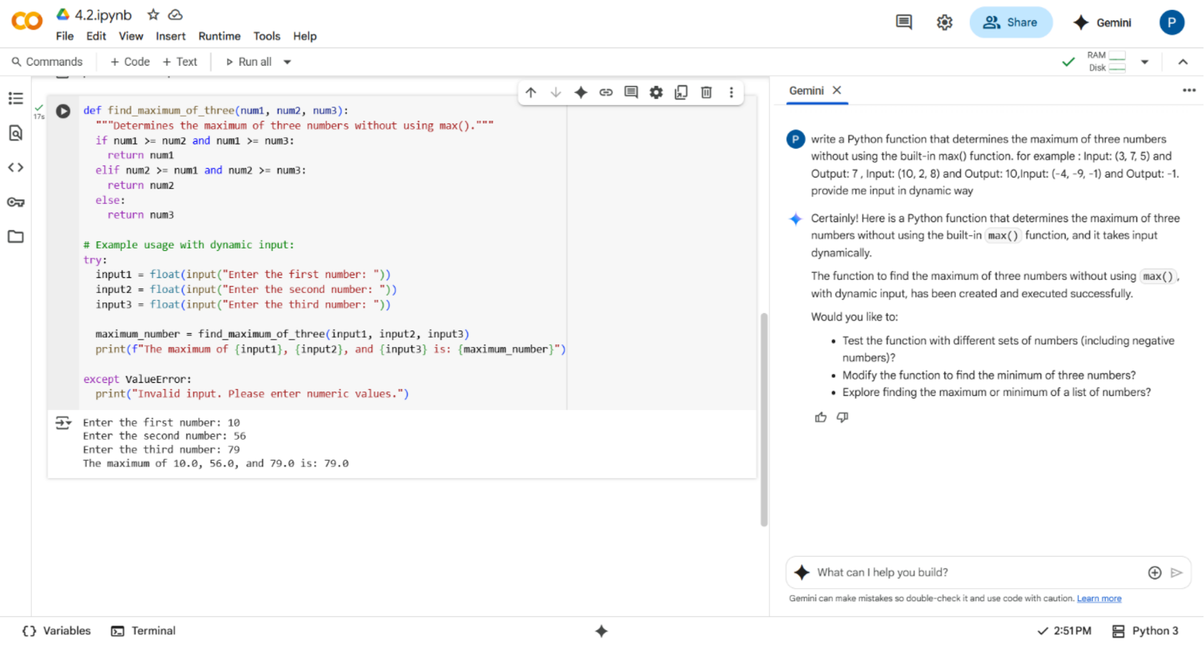
Observation:

Zero-shot prompting is faster and works if the task is simple and unambiguous, but risks misinterpretation.Few-shot prompting is more reliable because examples clarify expectations and edge cases, leading to better, more accurate code. For a task like palindrome checking (where rules like ignoring punctuation and case matter), few-shot prompting is superior.

Task-5:

Prompt: write a Python function that determines the maximum of three numbers without using the built-in max() function. for example : Input: (3, 7, 5) and Output: 7 , Input: (10, 2, 8) and Output: 10,Input: (-4, -9, -1) and Output: -1. provide me input in dynamic way

Code&Output:


4.2.ipynb
File Edit View Insert Runtime Tools Help
Commands
+ Code
+ Text
Run all
Share
Gemini
RAM
Disk
17s
Gemini
What can I help you build?
Gemini can make mistakes so double-check it and use code with caution. Learn more
Variables
Terminal
2:51 PM
Python 3

```python
def find_maximum_of_three(num1, num2, num3):
  """Determines the maximum of three numbers without using max()."""
  if num1 >= num2 and num1 >= num3:
    return num1
  elif num2 >= num1 and num2 >= num3:
    return num2
  else:
    return num3

# Example usage with dynamic input:
try:
  input1 = float(input("Enter the first number: "))
  input2 = float(input("Enter the second number: "))
  input3 = float(input("Enter the third number: "))

  maximum_number = find_maximum_of_three(input1, input2, input3)
  print(f"The maximum of {input1}, {input2}, and {input3} is: {maximum_number}")

except ValueError:
  print("Invalid input. Please enter numeric values.")
```

```
Enter the first number: 10
Enter the second number: 56
Enter the third number: 79
The maximum of 10.0, 56.0, and 79.0 is: 79.0
```

write a Python function that determines the maximum of three numbers without using the built-in max() function. for example : Input: (3, 7, 5) and Output: 7 , Input: (10, 2, 8) and Output: 10,Input: (-4, -9, -1) and Output: -1. provide me input in dynamic way

Certainly! Here is a Python function that determines the maximum of three numbers without using the built-in `max()` function, and it takes input dynamically.

The function to find the maximum of three numbers without using `max()`, with dynamic input, has been created and executed successfully.

Would you like to:

- Test the function with different sets of numbers (including negative numbers)?
- Modify the function to find the minimum of three numbers?
- Explore finding the maximum or minimum of a list of numbers?

Observation: The program correctly determines the largest of three numbers without using the built-in max() function. By comparing the numbers step by step with conditional statements, it can handle positive, negative, and mixed inputs. This shows how logical thinking and control structures in Python can replace built-in functions to solve problems efficiently.