

# AI-Assisted Coding

## Name:Peravena Navadeep

## Htno:2403A52019

### Task-1:

**Prompt:** Generate Fibonacci sequence up to n terms with user input.

### Code&Output:

The screenshot displays a Google Colab notebook titled 'Untitled6.ipynb'. The notebook contains a Python function `fibonacci_sequence(n)` that generates a Fibonacci sequence up to `n` terms. The function uses a loop to calculate the sequence and returns it. Below the function, there is a `try-except` block that prompts the user to enter the number of terms. If the input is a positive integer, it calls the `fibonacci_sequence` function and prints the result. If the input is invalid, it prints an error message.

```
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  
  
try:  
    n_terms = int(input("Enter the number of terms for the Fibonacci sequence: "))  
    if n_terms <= 0:  
        print("Please enter a positive integer.")  
    else:  
        fib_sequence = fibonacci_sequence(n_terms)  
        print("Fibonacci sequence:")  
        print(fib_sequence)  
except ValueError:  
    print("Invalid input. Please enter an integer.")
```

The output of the code shows the user entering '5' and the resulting Fibonacci sequence: `[0, 1, 1, 2, 3]`.

On the right side of the notebook, there is a Gemini chat interface. The chat history shows a prompt: 'Generate Fibonacci sequence up to n terms with user input.' and a response from Gemini: 'Sure, I can help you with that! I'll generate the code to generate the Fibonacci sequence. I've generated the Fibonacci sequence based on your input. What would you like to do next? Perhaps we could: Calculate the sum of the sequence? Find a specific term in the sequence? Visualize the sequence? Explore another mathematical concept? Let me know what you have in mind!'.

**Observation:** The `fibonacci_sequence(n)` function builds the sequence by starting with 0 and 1, then repeatedly adding the last two numbers to get the next.

## Task-2:

**Prompt:** Convert Fahrenheit to Celsius using a function with user input, print result rounded to 2 decimals.

## Code&Output:

The screenshot displays a Google Colab notebook environment. The top browser tabs include 'LAB TEST-4: AI Assisted Coding', 'assignment 4.2', 'Untitled6.ipynb - Colab', and 'Fibonacci sequence explanation'. The notebook's address bar shows a Google Drive link. The Colab interface includes a menu bar (File, Edit, View, Insert, Runtime, Tools, Help) and a toolbar with icons for commands, code, text, and running. The code editor contains a Python function `fahrenheit_to_celsius` that takes a Fahrenheit value and returns the Celsius equivalent, rounded to two decimal places. It also includes a try-except block for user input validation. The output area shows the function being called with 250, resulting in 121.11°C. To the right, a Gemini AI chat window is open, displaying a prompt about converting Fahrenheit to Celsius and a helpful response with code and suggestions for further exploration. The bottom status bar shows system information like temperature (26°C), time (2:04 PM), and language (English India).

```
def fahrenheit_to_celsius(fahrenheit):  
    """Converts Fahrenheit to Celsius."""  
    celsius = (fahrenheit - 32) * 5/9  
    return celsius  
  
try:  
    fahrenheit_input = float(input("Enter temperature in Fahrenheit: "))  
    celsius_output = fahrenheit_to_celsius(fahrenheit_input)  
    print(f"{fahrenheit_input}°F is equal to {celsius_output:.2f}°C")  
except ValueError:  
    print("Invalid input. Please enter a number.")
```

Enter temperature in Fahrenheit: 250  
250.0°F is equal to 121.11°C

**Gemini** X

Convert Fahrenheit to Celsius using a function with user input, print result rounded to 2 decimals.

I can certainly help with that! Here's the code to convert Fahrenheit to Celsius with user input and round the result.

I have converted the temperature from Fahrenheit to Celsius for you. What would you like to explore next? We could:

- Convert Celsius back to Fahrenheit?
- Work with other temperature units like Kelvin?
- Calculate the average of a list of temperatures?
- Perform a different unit conversion?

Let me know what you'd like to do!

What can I help you build?

Gemini can make mistakes so double-check it and use code with caution. [Learn more](#)

**Observation:** The gemini generates a function which converts a Fahrenheit to the Celsius , as I give the example , it may checks the example and give the exact suitable function to it.

**Task-3:**

**Prompt:** Write a Python program with a function to extract the domain from email addresses. Take multiple emails as dynamic user input and print only the domain part for each.

**Code&Output:**

```
def extract_domain(email):  
    """Extracts the domain from an email address."""  
    try:  
        return email.split('@')[1]  
    except IndexError:  
        return "Invalid email format"  
  
print("Enter email addresses one by one. Type 'done' when you are finished.")  
emails = []  
while True:  
    email_input = input("Enter an email address: ")  
    if email_input.lower() == 'done':  
        break  
    emails.append(email_input)  
  
print("\nExtracted domains:")  
for email in emails:  
    domain = extract_domain(email)  
    print(f"Email: {email}, Domain: {domain}")
```

Enter email addresses one by one. Type 'done' when you are finished.  
Enter an email address: koushik5511@gmail.com  
Enter an email address: yamshi1432@outlook.com  
Enter an email address: pardhu2001@email.com  
Enter an email address: done

Extracted domains:  
Email: koushik5511@gmail.com, Domain: gmail.com  
Email: yamshi1432@outlook.com, Domain: outlook.com  
Email: pardhu2001@email.com, Domain: email.com

**Gemini**

Write a Python program with a function to extract the domain from email addresses. Take multiple emails as dynamic user input and print only the domain part for each.

I can help you with that. I'll create a Python program that defines a function to extract the domain from email addresses, takes multiple email addresses as input from the user, and then prints the domain for each email.

I've successfully extracted the domains from the email addresses you provided. What would you like to do next? We could:

- Validate the email addresses?
- Count the number of emails for each domain?
- Extract other parts of the email address, like the username?
- Work with a list of emails from a file?

Let me know how you'd like to proceed!

What can I help you build?

Gemini can make mistakes so double-check it and use code with caution. [Learn more](#)

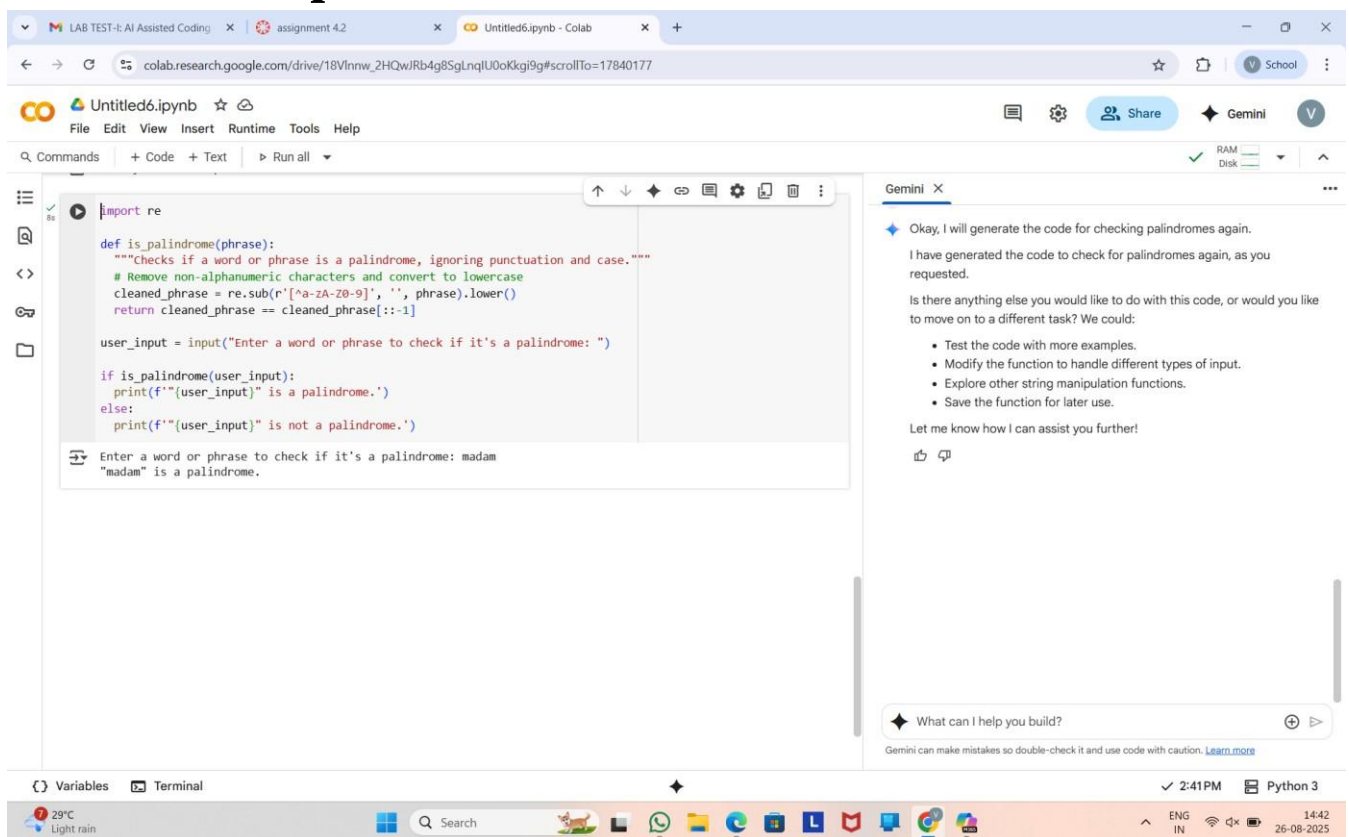
**Observation:** The code correctly extracts the domain portion of a valid email and gives an appropriate error message if the @ symbol is missing. However, it does not handle other invalid cases (like multiple @ symbols or missing domain part).

## Task-4:

**Prompt:** Write a Python program with a function to check if a word or phrase is a palindrome, ignoring punctuation and case. Take dynamic user input and print

whether it is a palindrome or not. for example  
input: madam and output is "madam" is a palindrome  
and input: vamshi and output is "vamshi" is not a  
palindrome

## Code&Output:



The screenshot displays a Google Colab notebook titled 'Untitled6.ipynb'. The code in the notebook defines a function `is_palindrome` that checks if a word or phrase is a palindrome, ignoring punctuation and case. It uses `re.sub` to remove non-alphanumeric characters and `.lower()` to convert to lowercase. The function then compares the cleaned phrase with its reverse. Below the function definition, there is a user input prompt: 'Enter a word or phrase to check if it's a palindrome:'. The output shows the user entered 'madam', and the program correctly identifies it as a palindrome.

On the right side of the notebook, the Gemini AI chat interface is visible. It shows a conversation where the user asked for code to check palindromes, and Gemini provided the code. The chat also includes a list of suggestions for further exploration, such as testing with more examples, modifying the function for different input types, and saving the function for later use.

```
import re

def is_palindrome(phrase):
    """Checks if a word or phrase is a palindrome, ignoring punctuation and case."""
    # Remove non-alphanumeric characters and convert to lowercase
    cleaned_phrase = re.sub(r'[^a-zA-Z0-9]', '', phrase).lower()
    return cleaned_phrase == cleaned_phrase[::-1]

user_input = input("Enter a word or phrase to check if it's a palindrome: ")

if is_palindrome(user_input):
    print(f"{user_input} is a palindrome.")
else:
    print(f"{user_input} is not a palindrome.")
```

Enter a word or phrase to check if it's a palindrome: madam  
"madam" is a palindrome.

Gemini

Okay, I will generate the code for checking palindromes again.

I have generated the code to check for palindromes again, as you requested.

Is there anything else you would like to do with this code, or would you like to move on to a different task? We could:

- Test the code with more examples.
- Modify the function to handle different types of input.
- Explore other string manipulation functions.
- Save the function for later use.

Let me know how I can assist you further!

What can I help you build?

Gemini can make mistakes so double-check it and use code with caution. [Learn more](#)

**Observation:** The code given and also even the output will be the same for the both codes , but the few -shot prompt given examples gives a clarity that they guide both the code logic and the validation of correctness.

## Task-5:

**Prompt:** Generate a python function that the which gives the output the largest number from the three given input numbers , for example input:2,3,4 output : 4 is largest number , if we didn't give three correct numbers it raises an error, the input has to be taken from user and generate the largest number.

## Code&Output:

The screenshot displays a Google Colab notebook environment. The top browser bar shows the URL: `colab.research.google.com/drive/18Vlnnw_2HQwJRb4g8SgLnqLU0oKkgi9g#scrollTo=5ad09cd`. The notebook interface includes a menu bar (File, Edit, View, Insert, Runtime, Tools, Help) and a toolbar with icons for commands, code, text, and running. The code editor contains the following Python code:

```
def find_largest_number(num1, num2, num3):  
    """Finds the largest of three numbers."""  
    if not all(isinstance(i, (int, float)) for i in [num1, num2, num3]):  
        raise ValueError("All inputs must be numbers.")  
  
    return max(num1, num2, num3)  
  
try:  
    input_str = input("Enter three numbers separated by commas: ")  
    num_list = [float(num.strip()) for num in input_str.split(',')]  
  
    if len(num_list) != 3:  
        raise ValueError("Please enter exactly three numbers.")  
  
    largest = find_largest_number(num_list[0], num_list[1], num_list[2])  
    print(f"{largest} is the largest number.")  
  
except ValueError as e:  
    print(f"Error: {e}")  
except Exception as e:  
    print(f"An unexpected error occurred: {e}")
```

The output of the code execution is shown in the terminal:

```
Enter three numbers separated by commas: 50,1,5  
50.0 is the largest number.
```

On the right side, the Gemini AI chat interface is visible. It contains the following text:

generate a python function that the which gives the output the largest number from the three given input numbers , for example input:2,3,4 output : 4 is largest number , if we didn't give three correct numbers it raises an error, the input has to be taken from user and generate the largest number

I can help you with that! I will create a Python function that takes three numbers as input from the user, finds the largest among them, and handles incorrect input.

I've successfully generated the code to find the largest of three numbers and handle invalid input.

What would you like to do next? We could:

- Find the smallest number among three inputs.
- Find the largest or smallest number from a list of numbers.
- Sort the three numbers in ascending or descending order.
- Explore other numerical operations.

Let me know how else I can assist you!

What can I help you build?

Gemini can make mistakes so double-check it and use code with caution. [Learn more](#)

**Observation:** The code correctly takes three numbers as input from the user, validates that exactly three values are provided, and finds the largest among them. It also handles errors gracefully. However, it uses Python's built-in `max()` function, like the prompt asked to raises an error—so the logic works, then it fully satisfy the given requirement.