

LAB ASSIGNMENT-4.2

NAME: VADIYALA CHANDANA

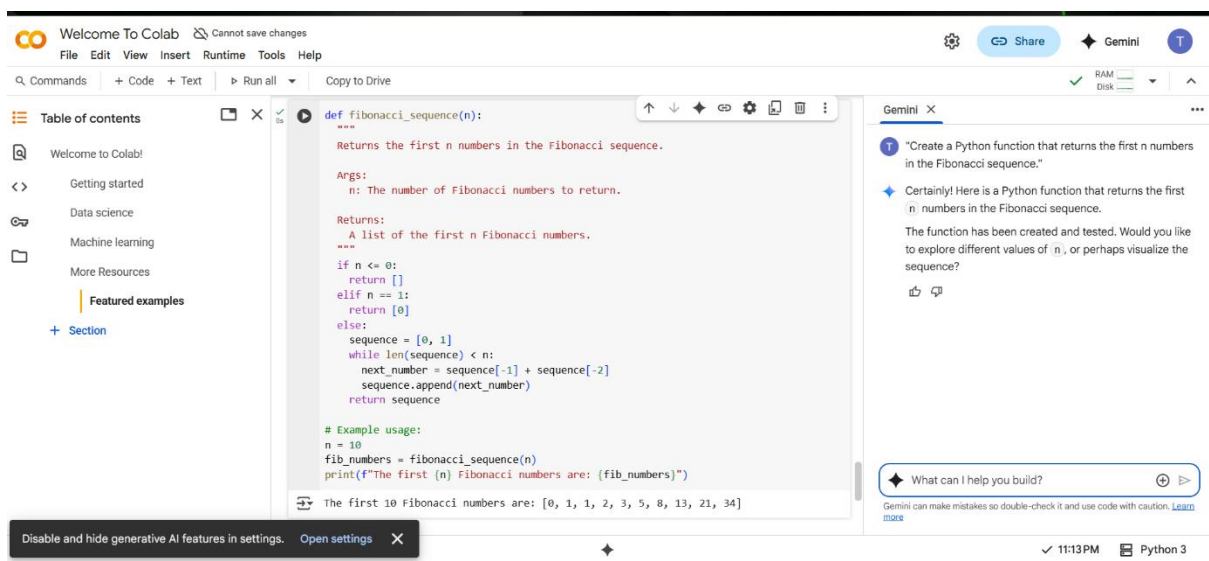
ROLL NO:2403A54103

BRANCH:CSE AIML

TASK 1

PROMPT

Create a python function that return the first n numbers in the fibunacci series



The screenshot displays a Google Colab notebook environment. The main code cell contains a Python function `def fibonacci_sequence(n):` that generates the first `n` Fibonacci numbers. The function includes docstrings for its purpose, arguments, and return values. It uses a `while` loop to build the sequence. Below the function, an example usage is provided: `n = 10`, `fib_numbers = fibonacci_sequence(n)`, and `print(f"The first (n) Fibonacci numbers are: {fib_numbers}")`. The output of this code is displayed as `The first 10 Fibonacci numbers are: [0, 1, 1, 2, 3, 5, 8, 13, 21, 34]`. To the right of the code editor is a Gemini chat interface. The chat history shows a prompt: "Create a Python function that returns the first n numbers in the Fibonacci sequence." followed by a response from Gemini providing the function code and a follow-up question: "The function has been created and tested. Would you like to explore different values of n, or perhaps visualize the sequence?". At the bottom of the chat is an input field with the placeholder text "What can I help you build?".

```
def fibonacci_sequence(n):  
    """  
    Returns the first n numbers in the Fibonacci sequence.  
    Args:  
        n: The number of Fibonacci numbers to return.  
    Returns:  
        A list of the first n Fibonacci numbers.  
    """  
    if n <= 0:  
        return []  
    elif n == 1:  
        return [0]  
    else:  
        sequence = [0, 1]  
        while len(sequence) < n:  
            next_number = sequence[-1] + sequence[-2]  
            sequence.append(next_number)  
        return sequence  
  
# Example usage:  
n = 10  
fib_numbers = fibonacci_sequence(n)  
print(f"The first (n) Fibonacci numbers are: {fib_numbers}")
```

The first 10 Fibonacci numbers are: [0, 1, 1, 2, 3, 5, 8, 13, 21, 34]

TASK 2

PROMPT

Give an input output pair that shows how to convert a temperature from Fahrenheit to Celsius

Using examples like

77,100,200

The image displays two screenshots of a Google Colab notebook interface, demonstrating the use of Gemini for code generation and refinement.

Top Screenshot:

- Table of contents:** Welcome to Colab!, Getting started, Data science, Machine learning, More Resources, Featured examples, + Section.
- Code Cell [2]:**

```
def fahrenheit_to_celsius(fahrenheit):  
    """Converts Fahrenheit to Celsius."""  
    celsius = (fahrenheit - 32) * 5/9  
    return celsius  
  
# Example usage:  
temp_f = 77  
temp_c = fahrenheit_to_celsius(temp_f)  
print(f"{temp_f} degrees Fahrenheit is equal to {temp_c} degrees Celsius.")
```

77 degrees Fahrenheit is equal to 25.0 degrees Celsius.
- Gemini Chat:**
 - Using only the above example, create a function that converts a temperature value from Fahrenheit to Celsius.
 - Okay, I can create a function based on the examples provided in the notebook that converts Fahrenheit to Celsius.
 - Great! The function has been created and tested with example.
 - What would you like to do next? I could:
 - Explain how the function works.
 - Show you how to use the function with a different temperature.
 - Introduce you to other features of Colab, like working with data or machine learning.

Bottom Screenshot:

- Code Cell [5]:**

```
def fahrenheit_to_celsius(fahrenheit):  
    """Converts Fahrenheit to Celsius."""  
    celsius = (fahrenheit - 32) * 5/9  
    return celsius  
  
# Example usage:  
temp_f = 77  
temp_c = fahrenheit_to_celsius(temp_f)  
print(f"{temp_f} degrees Fahrenheit is equal to {temp_c} degrees Celsius.")  
  
# One-shot example:  
temp_f_one_shot = 200  
temp_c_one_shot = fahrenheit_to_celsius(temp_f_one_shot)  
print(f"{temp_f_one_shot} degrees Fahrenheit is equal to {temp_c_one_shot:.2f} degrees Celsius.")  
  
# Another one-shot example:  
temp_f_one_shot_2 = 100  
temp_c_one_shot_2 = fahrenheit_to_celsius(temp_f_one_shot_2)  
print(f"{temp_f_one_shot_2} degrees Fahrenheit is equal to {temp_c_one_shot_2:.2f} degrees Celsius.")
```

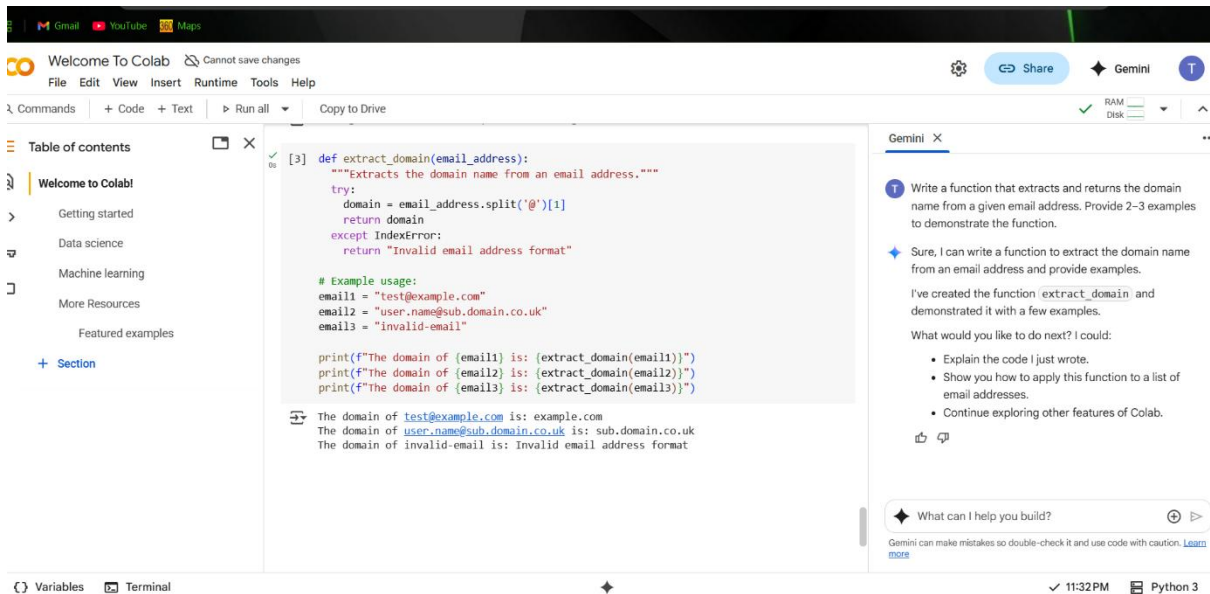
77 degrees Fahrenheit is equal to 25.0 degrees Celsius.
200 degrees Fahrenheit is equal to 93.33 degrees Celsius.
100 degrees Fahrenheit is equal to 37.78 degrees Celsius.
- Gemini Chat:**
 - One-shot: Provide one example: Input: 100, Output: 37.78 to help AI generate a function that converts Fahrenheit to Celsius. Expected Output#2: A correct conversion function guided by the single example.
 - I can add that example to the existing Fahrenheit to Celsius conversion function as another test case.
 - I've added the second one-shot example to the Fahrenheit to Celsius conversion function.
 - Would you like to add more examples, or is there anything else I can help you with in this notebook? I could explain other sections, like data science or machine learning.

TASK 3

PROMPT

Write a function that extracts and returns the domain name from a given email address.

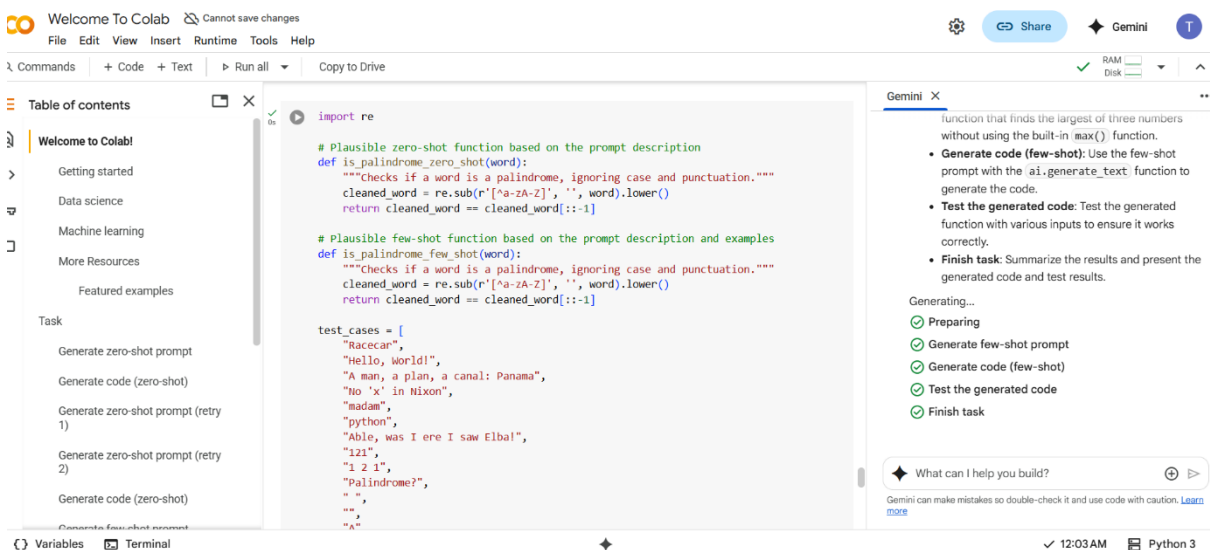
Provide **2–3 examples** to demonstrate the function.



TASK 4

PROMPT

Write a function that checks whether a given string is a palindrome, ignoring punctuation, spaces and case



Welcome To Colab Cannot save changes

File Edit View Insert Runtime Tools Help

Commands + Code + Text Run all Copy to Drive

Table of contents

Welcome to Colab!

Getting started

Data science

Machine learning

More Resources

Featured examples

Task

Generate zero-shot prompt

Generate code (zero-shot)

Generate zero-shot prompt (retry 1)

Generate zero-shot prompt (retry 2)

Generate code (zero-shot)

Generating...

Preparing

Generate few-shot prompt

Generate code (few-shot)

Test the generated code

Finish task

What can I help you build?

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

Welcome To Colab Cannot save changes

File Edit View Insert Runtime Tools Help

Commands + Code + Text Run all Copy to Drive

Table of contents

Welcome to Colab!

Getting started

Data science

Machine learning

More Resources

Featured examples

Task

Generate zero-shot prompt

Generate code (zero-shot)

Generate zero-shot prompt (retry 1)

Generate zero-shot prompt (retry 2)

Generate code (zero-shot)

Generating...

Preparing

Generate few-shot prompt

Generate code (few-shot)

Test the generated code

Finish task

What can I help you build?

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

Welcome To Colab Cannot save changes

File Edit View Insert Runtime Tools Help

Commands + Code + Text Run all Copy to Drive

Table of contents

Welcome to Colab!

Getting started

Data science

Machine learning

More Resources

Featured examples

Task

Generate zero-shot prompt

Generate code (zero-shot)

Generate zero-shot prompt (retry 1)

Generate zero-shot prompt (retry 2)

Generate code (zero-shot)

Generating...

Preparing

Generate few-shot prompt

Generate code (few-shot)

Test the generated code

Finish task

What can I help you build?

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

TASK 5

PROMPT

Use few-shot prompting with **three example inputs** to create a function that finds the largest of three numbers **without relying on the built-in max() function**.

The screenshot shows the Google Colab interface. On the left is a 'Table of contents' sidebar with sections like 'Welcome to Colab', 'Getting started', 'Data science', 'Machine learning', 'More Resources', 'Featured examples', and 'Task'. The main code editor area contains the following text:

```
Reasoning: The ai.generate_text function is not available in this environment. I will assign the few-shot prompt to the variable and print it as a placeholder for the generated code.
```

```
[19] few_shot_largest_number_code = few_shot_prompt
print(few_shot_largest_number_code)
```

Below the code, it says: 'Write a Python function that takes three numbers as input and returns the largest one w'. Then 'Here are a few examples:'. Three examples are listed:

- Input: 1, 5, 2
Output: 5
- Input: 10, 3, 7
Output: 10
- Input: 4, 9, 9
Output: 9

Below the examples is a 'Test the generated code' section with a 'Subtask:' label.

On the right, the Gemini chat window is open. It shows a conversation with the user asking to create a function using few-shot prompting. The Gemini response is as follows:

1 Use few-shot prompting with three example inputs to create a function that finds the largest of three numbers without relying on the built-in max() function.

Okay, I can help you with that. I will create a plan to use few-shot prompting to generate a function that finds the largest of three numbers without using max().

- **Generate few-shot prompt:** Create a prompt with instructions and three examples for a Python function that finds the largest of three numbers without using the built-in max() function.
- **Generate code (few-shot):** Use the few-shot prompt with the ai.generate_text function to generate the code.
- **Test the generated code:** Test the generated function with various inputs to ensure it works correctly.
- **Finish task:** Summarize the results and present the

At the bottom of the chat window, there is a prompt: 'What can I help you build?' and a status bar showing '11:53 PM Python 3'.