

ASSIGNMENT-4.

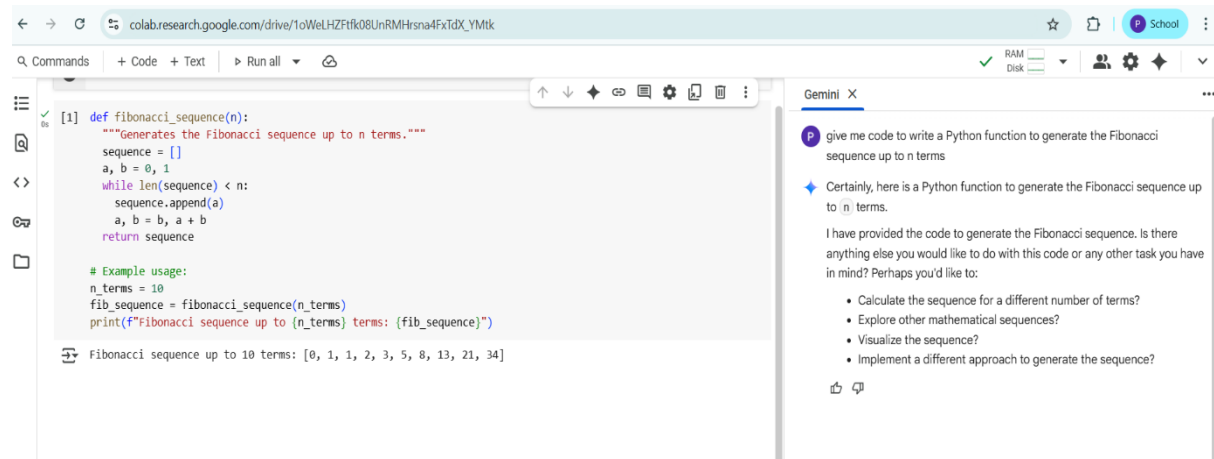
NAME: HANSINI REDDY

ROLL NO: 2403A510D5

BATCH:01

COURSE: AI ASSISTED CODING

TASK 1:



The screenshot shows a Google Colab notebook with a Python function to generate the Fibonacci sequence. The function is defined as follows:

```
[1] def fibonacci_sequence(n):  
    """Generates the Fibonacci sequence up to n terms."""  
    sequence = []  
    a, b = 0, 1  
    while len(sequence) < n:  
        sequence.append(a)  
        a, b = b, a + b  
    return sequence  
  
# Example usage:  
n_terms = 10  
fib_sequence = fibonacci_sequence(n_terms)  
print(f"Fibonacci sequence up to {n_terms} terms: {fib_sequence}")
```

The output of the code is: Fibonacci sequence up to 10 terms: [0, 1, 1, 2, 3, 5, 8, 13, 21, 34].

The Gemini chat interface on the right shows the following conversation:

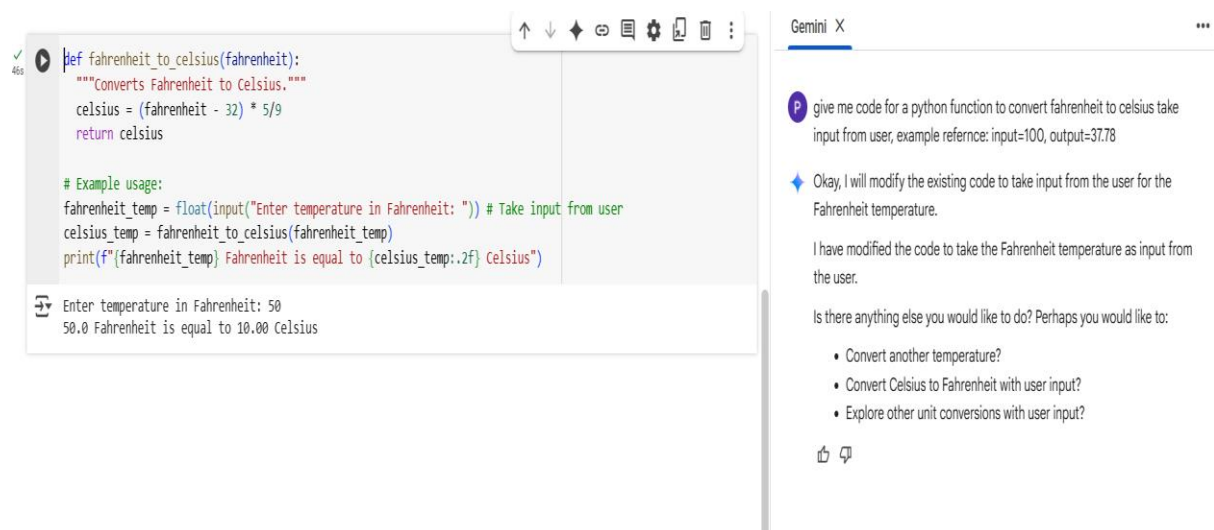
P: give me code to write a Python function to generate the Fibonacci sequence up to n terms

◆: Certainly, here is a Python function to generate the Fibonacci sequence up to `n` terms.

I have provided the code to generate the Fibonacci sequence. Is there anything else you would like to do with this code or any other task you have in mind? Perhaps you'd like to:

- Calculate the sequence for a different number of terms?
- Explore other mathematical sequences?
- Visualize the sequence?
- Implement a different approach to generate the sequence?

TASK 2:



The screenshot shows a Google Colab notebook with a Python function to convert Fahrenheit to Celsius. The function is defined as follows:

```
def fahrenheit_to_celsius(fahrenheit):  
    """Converts Fahrenheit to Celsius."""  
    celsius = (fahrenheit - 32) * 5/9  
    return celsius  
  
# Example usage:  
fahrenheit_temp = float(input("Enter temperature in Fahrenheit: ")) # Take input from user  
celsius_temp = fahrenheit_to_celsius(fahrenheit_temp)  
print(f"{fahrenheit_temp} Fahrenheit is equal to {celsius_temp:.2f} Celsius")
```

The output of the code is: Enter temperature in Fahrenheit: 50
50.0 Fahrenheit is equal to 10.00 Celsius

The Gemini chat interface on the right shows the following conversation:

P: give me code for a python function to convert fahrenheit to celsius take input from user, example reference: input=100, output=37.78

◆: Okay, I will modify the existing code to take input from the user for the Fahrenheit temperature.

I have modified the code to take the Fahrenheit temperature as input from the user.

Is there anything else you would like to do? Perhaps you would like to:

- Convert another temperature?
- Convert Celsius to Fahrenheit with user input?
- Explore other unit conversions with user input?

TASK 3:

```
def extract_domain(email):
    """Extracts the domain name from an email address."""
    try:
        domain = "@".split("@")[1]
        return domain
    except IndexError:
        return "Invalid email format"

# Example usage:
email_address = input("Enter an email address: ")
domain_name = extract_domain(email_address)
print(f"The domain name is: {domain_name}")
```

Enter an email address: "honey@gmail.com"
The domain name is: @gmail.com

Gemini

Give me code for a Python function that extracts the domain name from an email address. Take input from user Examples: Input: "hansi@gmail.com", Output: "@gmail.com" Input: "sunanditha@yahoo.com", Output: "@yahoo.com" Input: "hasini@company.org", Output: "@company.org"

Certainly, here is a Python function that extracts the domain name from an email address, taking input from the user.

I have provided the code to extract the domain name from an email address and take user input.

Is there anything else you would like to do? Perhaps you would like to:

- Test the function with a different email address?
- Modify the function to handle different edge cases (e.g., multiple "@" symbols)?
- Extract other parts of an email address (e.g., the username)?
- Work on a different task?

TASK 4:

ZERO SHOT:

```
[7] 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:
word1 = "Racecar"
word2 = "A man, a plan, a canal: Panama."
word3 = "hello"

print(f'{word1} is a palindrome: {is_palindrome(word1)}')
print(f'{word2} is a palindrome: {is_palindrome(word2)}')
print(f'{word3} is a palindrome: {is_palindrome(word3)}')
```

'Racecar' is a palindrome: True
'A man, a plan, a canal: Panama.' is a palindrome: False
'hello' is a palindrome: False

give me code for generating a function that checks whether a word is a palindrome, ignoring punctuation and case

Certainly, here is a Python function that checks whether a word is a palindrome, ignoring punctuation and case.

I have provided the code to check if a word is a palindrome, ignoring punctuation and case.

Is there anything else you would like to do? Perhaps you would like to:

- Test the function with different words or phrases?
- Modify the function to handle other types of inputs (e.g., sentences)?
- Explore different ways to check for palindromes?
- Work on a different task?

FEW SHOT:

```
import string

def is_palindrome(word):
    """
    Checks if a word is a palindrome, ignoring punctuation and case.

    Args:
        word: The input word (string).

    Returns:
        True if the word is a palindrome, False otherwise.
    """
    # Remove punctuation and convert to lowercase
    cleaned_word = "".join(char.lower() for char in word if char not in string.punctuation and char.isprintable())
    # Check if the cleaned word is a palindrome
    return cleaned_word == cleaned_word[::-1]

# Get input from the user
user_input = input("Enter a word or phrase to check if it's a palindrome: ")

# Check and print the result
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: Level
"Level" is a palindrome.

Give me code for generating a function that checks whether a word is a palindrome, ignoring punctuation and case. Take the input from user Examples reference: Input: "Racecar" Output: True, Input: "A man, a canal: Panama" Output: True, Input: "Hello!" Output: False, Input: "No lemon, no melon" Output: True

Okay, I can provide you with Python code to create a function that checks if a word is a palindrome, ignoring punctuation and case, and takes input from the user.

Working...

TASK 5:

```
def find_maximum(num1, num2, num3):
    """Finds the maximum of three numbers without using max()."""
    maximum = num1
    if num2 > maximum:
        maximum = num2
    if num3 > maximum:
        maximum = num3
    return maximum

# Example usage:
try:
    num1 = float(input("Enter the first number: "))
    num2 = float(input("Enter the second number: "))
    num3 = float(input("Enter the third number: "))
    maximum_number = find_maximum(num1, num2, num3)
    print(f"The maximum of {num1}, {num2}, and {num3} is: {maximum_number}")
except ValueError:
    print("Invalid input. Please enter valid numbers.")
```

Enter the first number: 89
Enter the second number: 66
Enter the third number: 90
The maximum of 89.0, 66.0, and 90.0 is: 90.0

Gemini X

Give me code to write a Python function that returns the maximum of three numbers, without using built-in max(). take input from user Examples: Input: 5, 1, 9, Output: 9 Input: -15, -39, -8, Output: -8 Input: 1000, 10, 100, Output: 1000

Certainly, here is a Python function that finds the maximum of three numbers without using the built-in max() function, and takes input from the user.

I have provided the code to find the maximum of three numbers and take user input.

Is there anything else you would like to do? Perhaps you would like to:

- Test the function with different numbers?
- Modify the function to find the maximum of more than three numbers?
- Compare this approach to using the built-in max() function?
- Work on a different task?