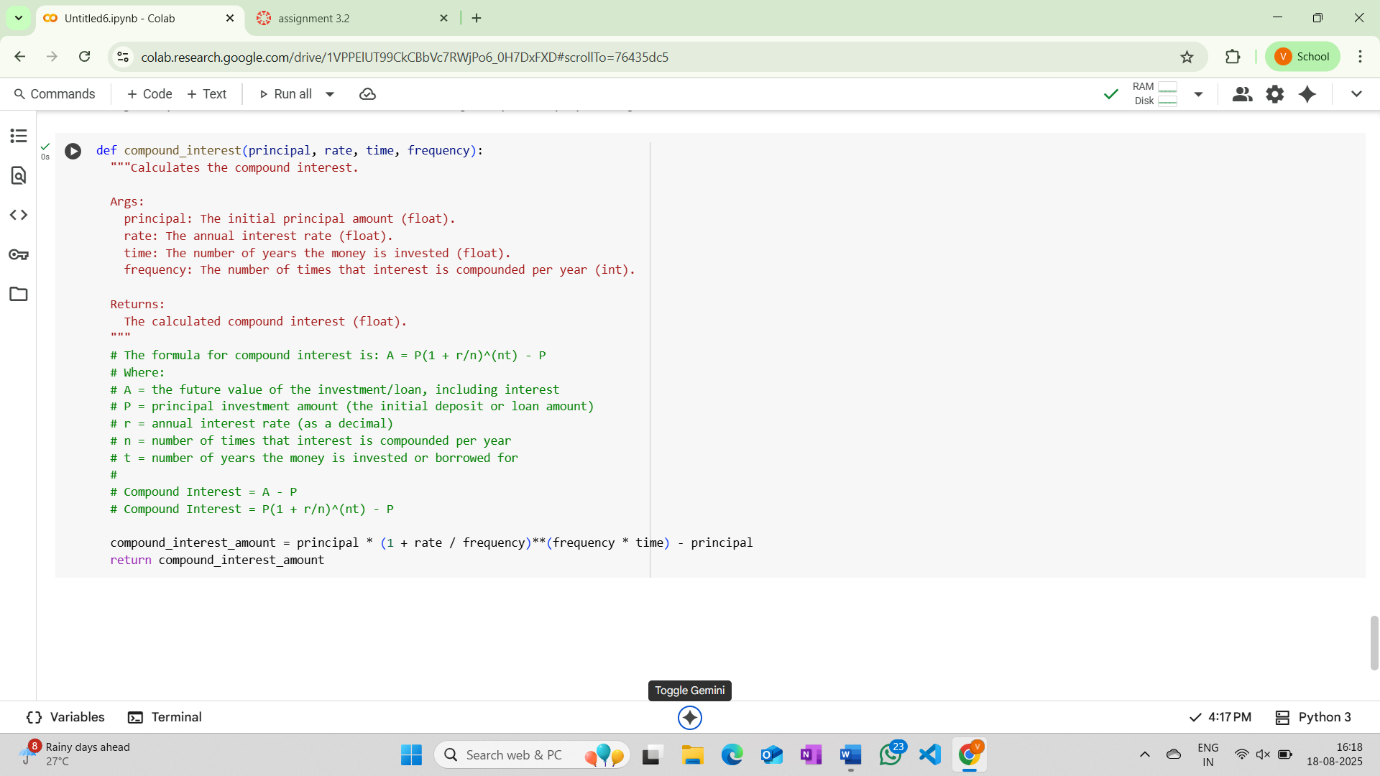
Assingment-3.2

Task-1

Prompt: Using python give me a code using functions to find compound interest



Explaination:

principal = 1000

This line assigns the value 1000 to the variable principal. This represents the initial principal amount for the compound interest calculation.

rate = 0.05

This line assigns the value 0.05 to the variable rate. This represents the annual interest rate as a decimal (5%).

time = 10

This line assigns the value 10 to the variable time. This represents the number of years the money is invested.

frequency = 4

This line assigns the value 4 to the variable frequency. This represents the number of times the interest is compounded per year (quarterly).

compound\_interest\_amount = compound\_interest(principal, rate, time, frequency)

This line calls the compound\_interest function with the previously defined variables as arguments and assigns the returned value (the calculated compound interest) to the variable compound\_interest\_amount.

print(f"Principal: ${principal}")

This line prints the value of the principal variable, formatted as a currency amount.

print(f"Annual Interest Rate: {rate\*100}%")

This line prints the annual interest rate. It multiplies the rate by 100 to display it as a percentage.

print(f"Time (in years): {time}")

This line prints the value of the time variable, indicating the number of years.

print(f"Compounding Frequency: {frequency} times per year")

This line prints the value of the frequency variable, indicating how many times per year the interest is compounded.

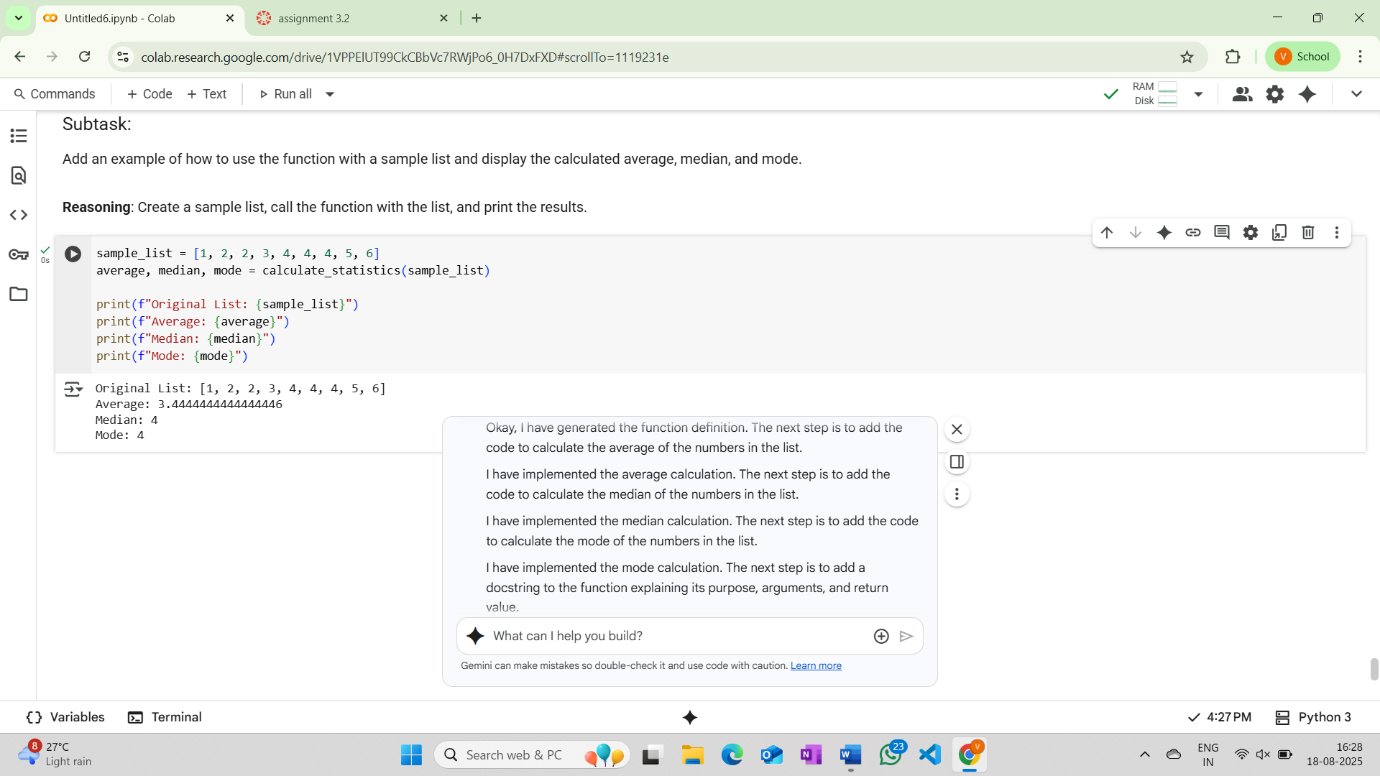
print(f"Compound Interest: ${compound\_interest\_amount:.2f}")

This line prints the calculated compound interest amount.

* compound\_interest\_amount:.2f formats the compound\_interest\_amount to two decimal places, which is standard for currency.
* The f-string includes a dollar sign before the formatted amount

Task-2

#using python write code for calculating average ,median and mode of a list of numbers using functions



Explaination:

sample\_list = [1, 2, 2, 3, 4, 4, 4, 5, 6]

This line creates a list named sample\_list and initializes it with the numbers 1, 2, 2, 3, 4, 4, 4, 5, and 6. This list will be used as input for the calculate\_statistics function.

average, median, mode = calculate\_statistics(sample\_list)

This line calls the calculate\_statistics function with sample\_list as the argument. The function is expected to return three values (the calculated average, median, and mode), which are then unpacked and assigned to the variables average, median, and mode respectively.

print(f"Original List: {sample\_list}")

This line prints the content of the sample\_list to the console. The f-string is used to include the list directly in the output string.

print(f"Average: {average}")

This line prints the value stored in the average variable to the console. The f-string is used to include the average value in the output string.

print(f"Median: {median}")

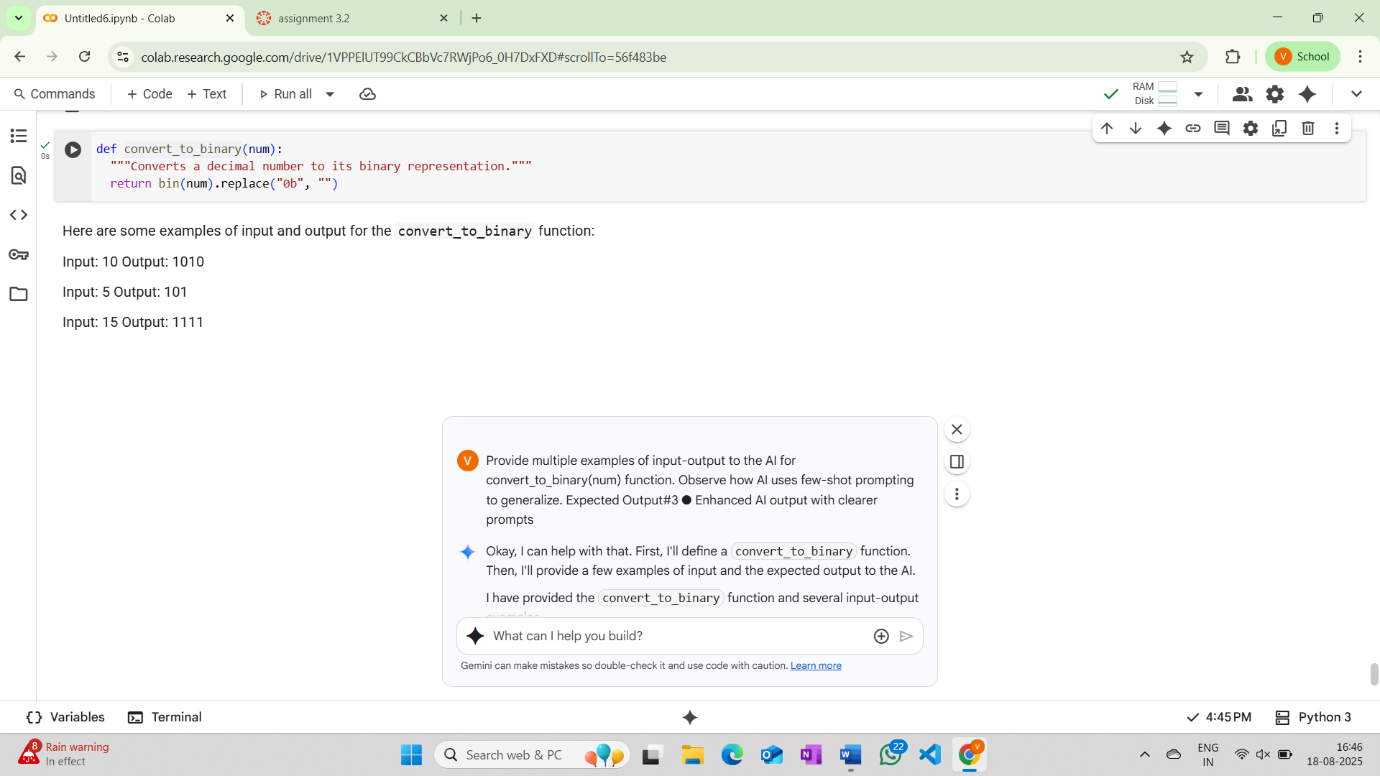
This line prints the value stored in the median variable to the console. The f-string is used to include the median value in the output string.

print(f"Mode: {mode}")

This line prints the value stored in the mode variable to the console. The f-string is used to include the mode value in the output string. Note that mode can be a single value or a list of values depending on the input list and the logic within the calculate\_statistics function.

Task-3

#simple python code for the convert\_to\_binary(num) function along with example



Explaination:

def convert\_to\_binary(num):

This line defines a Python function named convert\_to\_binary that takes one argument, num.

  """Converts a decimal number to its binary representation."""

This is a docstring, which explains the purpose of the function. It tells us that this function converts a decimal number to its binary representation.

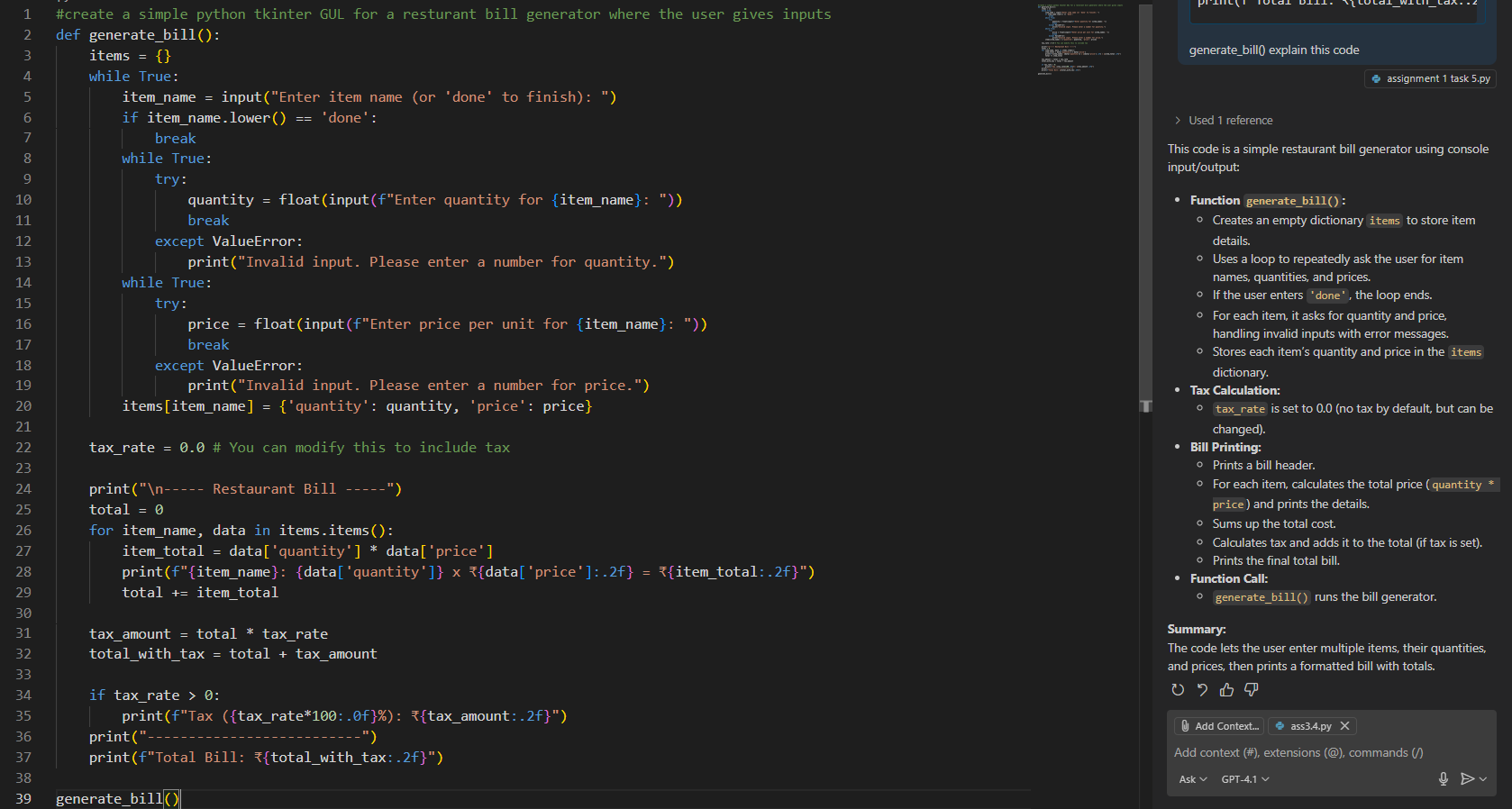
  return bin(num).replace("0b", "")

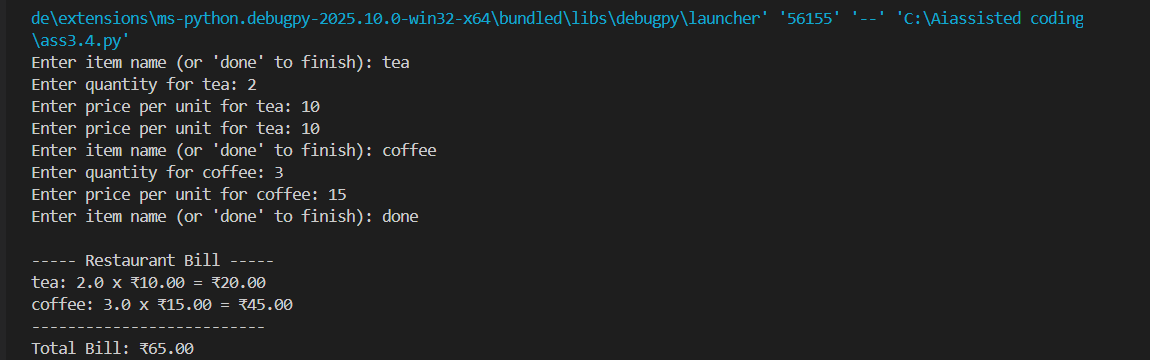
This line calculates and returns the binary representation of the input number.

* bin(num) is a built-in Python function that converts an integer num to its binary string representation. The result will have a "0b" prefix (e.g., bin(10) returns "0b1010").
* .replace("0b", "") is a string method that removes the "0b" prefix from the binary string generated by bin(num).
* return sends the resulting binary string back as the output of the function.

Task-4:

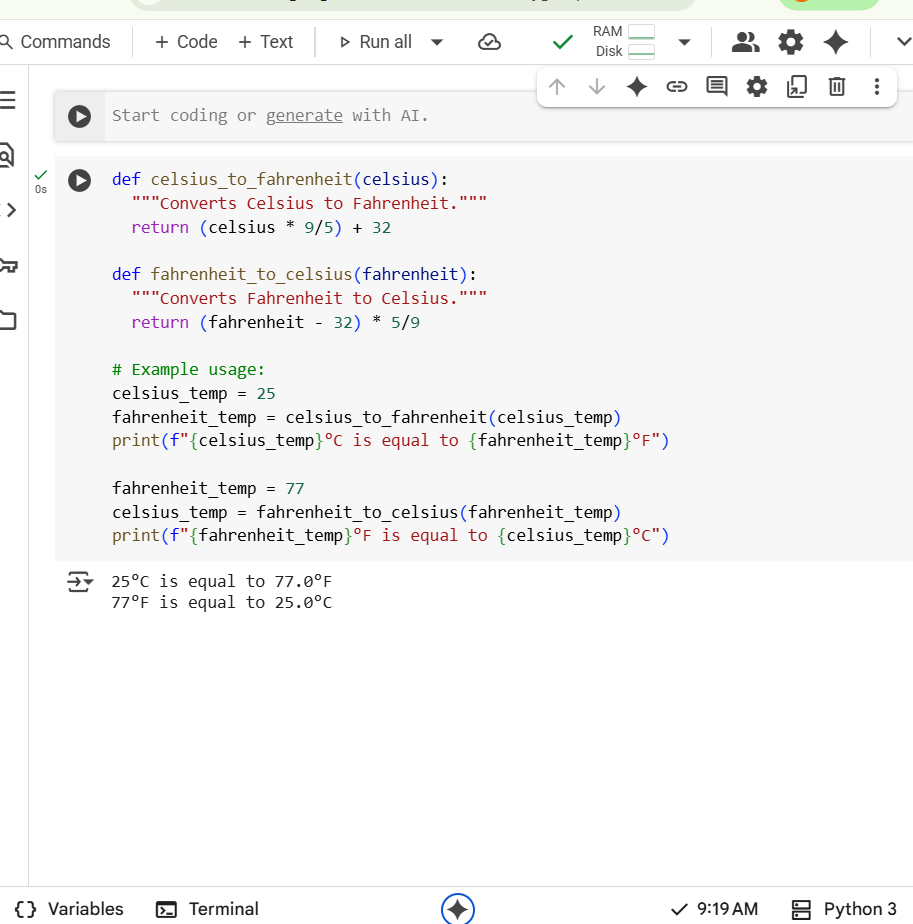
# using python generate restutant bill in text format





Task-5:

#using simple python code write code for conversion of temperature using function



Explaination:

def celsius\_to\_fahrenheit(celsius):

This line defines a function named celsius\_to\_fahrenheit that takes one argument, celsius.

  """Converts Celsius to Fahrenheit."""

This is a docstring, which explains what the function does. It's good practice to include these in your functions.

  return (celsius \* 9/5) + 32

This line performs the conversion from Celsius to Fahrenheit using the formula (Celsius \* 9/5) + 32 and returns the result.

def fahrenheit\_to\_celsius(fahrenheit):

This line defines another function named fahrenheit\_to\_celsius that takes one argument, fahrenheit.

  """Converts Fahrenheit to Celsius."""

This is the docstring for the fahrenheit\_to\_celsius function, explaining its purpose.

  return (fahrenheit - 32) \* 5/9

This line performs the conversion from Fahrenheit to Celsius using the formula (Fahrenheit - 32) \* 5/9 and returns the result.

# Example usage:

This is a comment indicating that the following lines demonstrate how to use the functions.

celsius\_temp = 25

This line assigns the value 25 to the variable celsius\_temp.

fahrenheit\_temp = celsius\_to\_fahrenheit(celsius\_temp)

This line calls the celsius\_to\_fahrenheit function with celsius\_temp (which is 25) as input and stores the returned Fahrenheit value in the variable fahrenheit\_temp.

print(f"{celsius\_temp}°C is equal to {fahrenheit\_temp}°F")

This line prints a formatted string to the console showing the original Celsius temperature and its converted Fahrenheit equivalent. The f before the string indicates a f-string, allowing you to embed variables directly within the string using curly braces {}.

fahrenheit\_temp = 77

This line assigns the value 77 to the variable fahrenheit\_temp.

celsius\_temp = fahrenheit\_to\_celsius(fahrenheit\_temp)

This line calls the fahrenheit\_to\_celsius function with fahrenheit\_temp (which is 77) as input and stores the returned Celsius value in the variable celsius\_temp.

print(f"{fahrenheit\_temp}°F is equal to {celsius\_temp}°C")

This line prints a formatted string to the console showing the original Fahrenheit temperature and its converted Celsius equivalent.

Let me know if you have any other questions!