**NAME – PRIYANSHI ASATI**

**BATCH – 4 (AIML-NON HONS.)**

**SAP ID – 500108525**

**ROLL NO – R2142220982**

**PREDICTIVE ANALYSIS LAB-1**

**Q1. Write a Python program to perform basic arithmetic operations (addition, subtraction, multiplication, division, and modulus) on two numbers.**

* + Prompt the user to enter two numbers.
  + Perform the arithmetic operations and print the results.
  + Use appropriate operators and print formatting for clear output.
* CODE:

def arithmetic\_operations(num1, num2):

addition = num1 + num2

subtraction = num1 - num2

multiplication = num1 \* num2

if num2 != 0:

division = num1 / num2

else:

division = "cannot divide by zero"

if num2 != 0:

modulus = num1 % num2

else:

modulus = "cannot modulus by zero"

print(f"\nResults of arithmetic operations between {num1} and {num2}:")

print(f"Addition: {num1} + {num2} = {addition}")

print(f"Subtraction: {num1} - {num2} = {subtraction}")

print(f"Multiplication: {num1} \* {num2} = {multiplication}")

print(f"Division: {num1} / {num2} = {division}")

print(f"Modulus: {num1} % {num2} = {modulus}")

try:

number1 = float(input("Enter the first number: "))

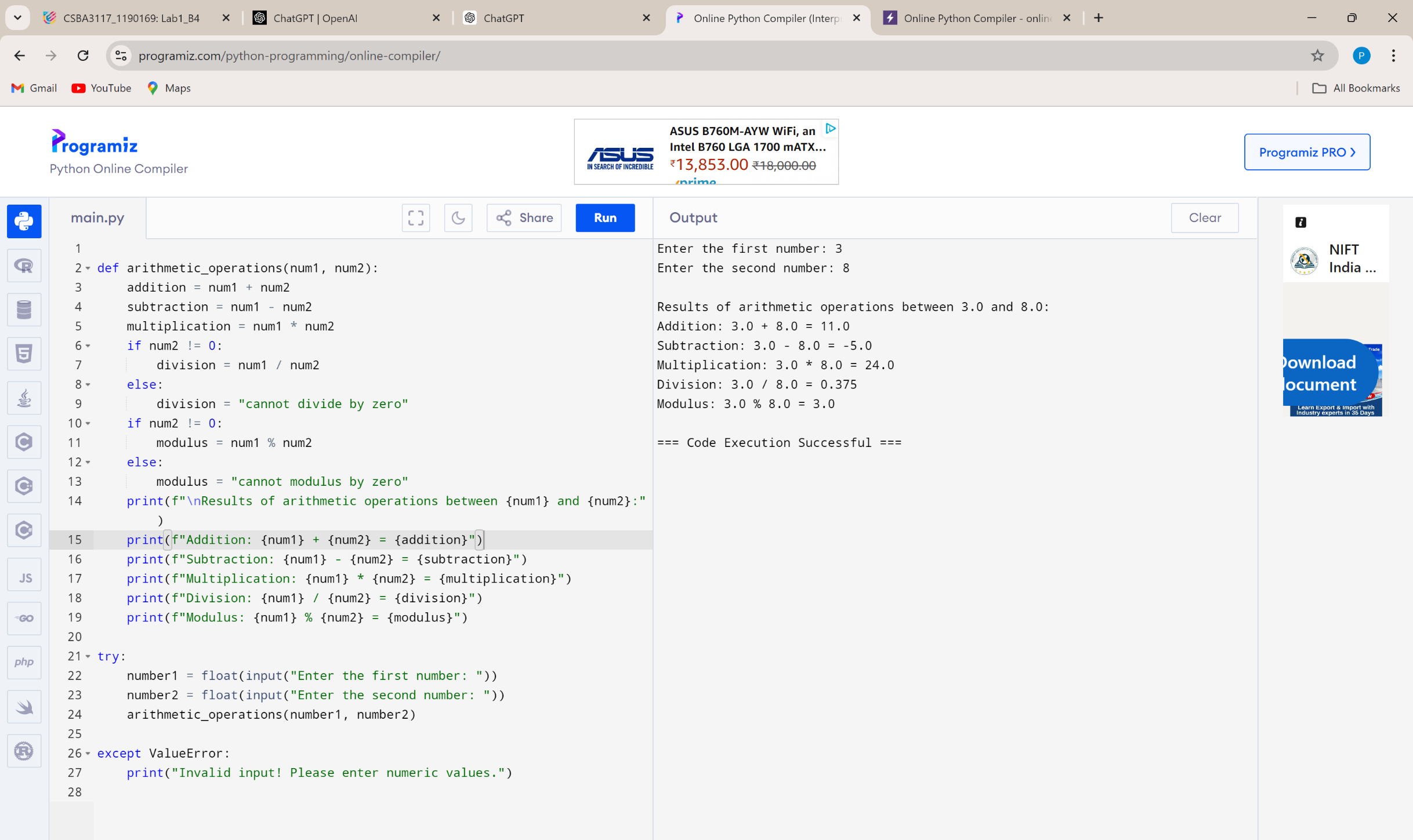
number2 = float(input("Enter the second number: "))

arithmetic\_operations(number1, number2)

except ValueError:

print("Invalid input! Please enter numeric values.")

* OUTPUT:



**Q2.  Create variables of different data types (integer, float, string, boolean) and perform basic operations on them.**

* + Assign values to variables of different data types.
  + Perform arithmetic operations on numeric data types.
  + Concatenate strings using the + operator.
  + Use logical operators to evaluate boolean expressions.
* CODE:

# Assign values to different data types

int\_var = 15

float\_var = 20.5

str\_var = "Predictive "

bool\_var = True

# Perform arithmetic operations on numeric data types

sum\_var = int\_var + float\_var

product\_var = int\_var \* float\_var

# Concatenate strings using the + operator

greeting = str\_var + "Analytics"

# Use logical operators to evaluate boolean expressions

logical\_and = bool\_var and (int\_var > 5)

logical\_or = bool\_var or (int\_var < 5)

# Print results

print(f"Sum: {int\_var} + {float\_var} = {sum\_var}")

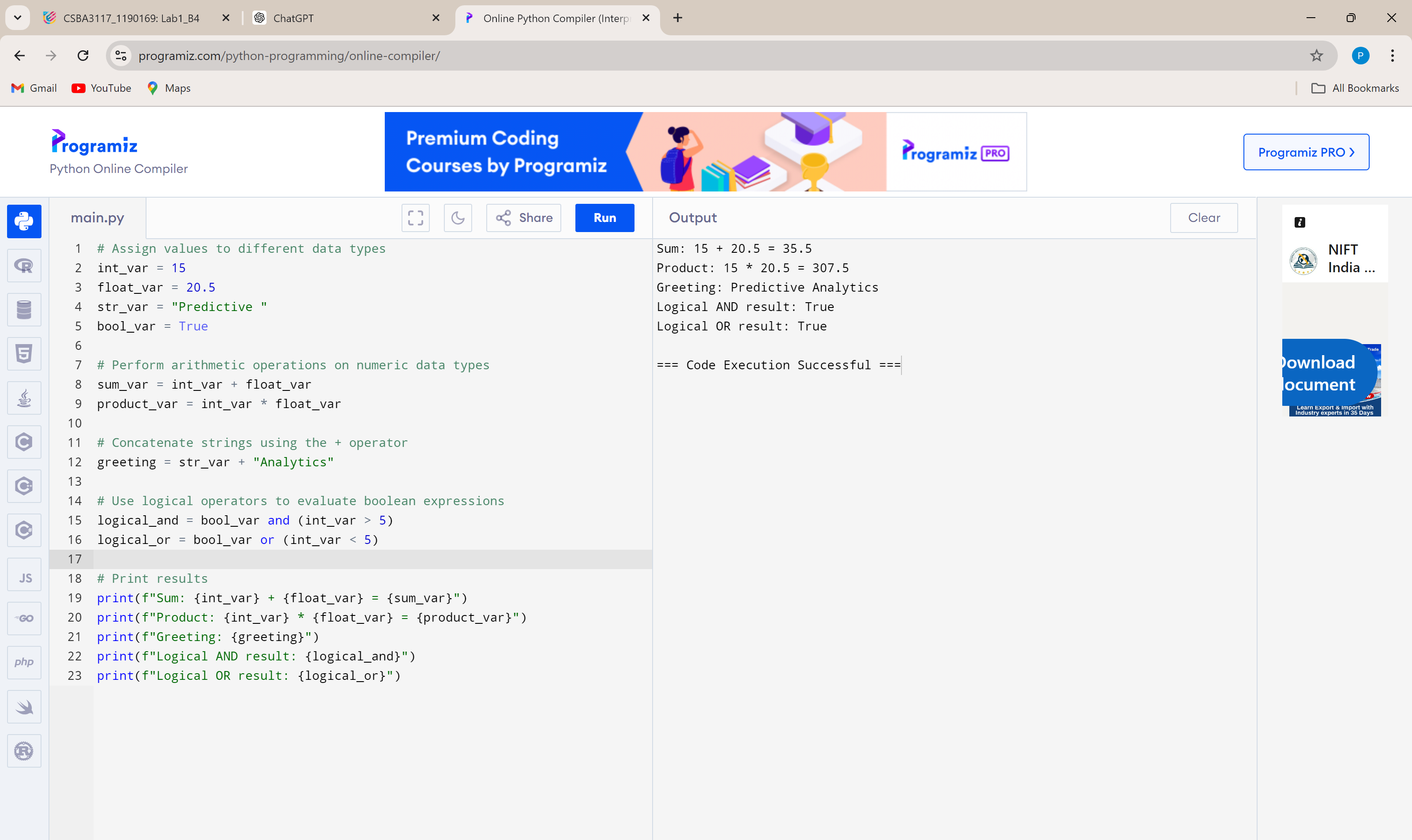
print(f"Product: {int\_var} \* {float\_var} = {product\_var}")

print(f"Greeting: {greeting}")

print(f"Logical AND result: {logical\_and}")

print(f"Logical OR result: {logical\_or}")

* OUTPUT:



**Q3. Write a program to take user input, process it, and display the result.**

* + Prompt the user to enter their name.
  + Greet the user using their name.
  + Calculate and print the user's age based on their birth year.
* CODE:

name = input("Enter your name: ")

print(f"Good Evevning, {name}!")

try:

birth\_year = int(input("Enter your birth year: "))

current\_year = 2024

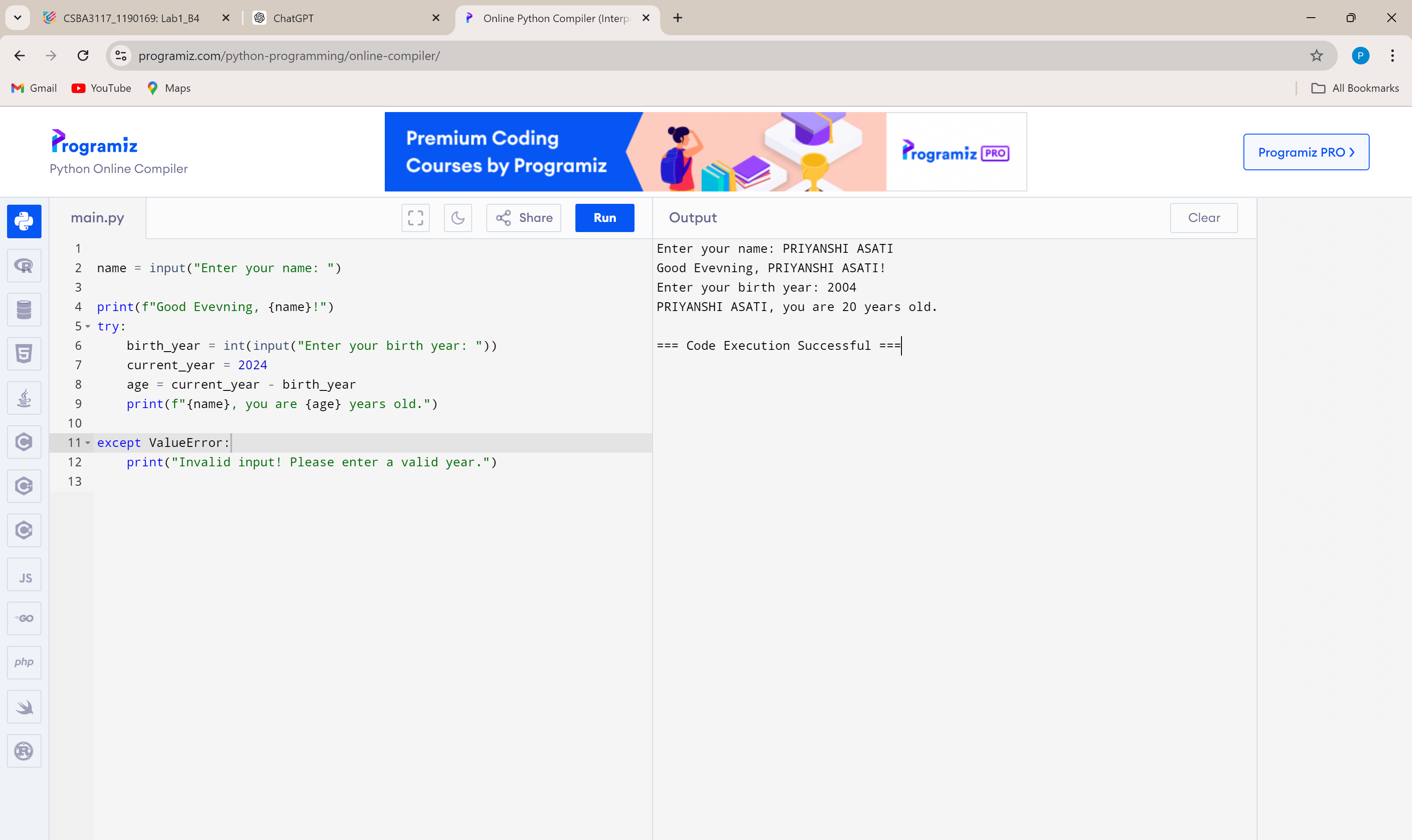
age = current\_year - birth\_year

print(f"{name}, you are {age} years old.")

except ValueError:

print("Invalid input! Please enter a valid year.")

* OUTPUT:



**Q4.  Write a program to check if a number is even or odd.**

* + Prompt the user to enter a number.
  + Use the modulus operator to determine if the number is even or odd.
  + Print the appropriate message.
* CODE:

try:

number = int(input("Enter a number: "))

if number % 2 == 0:

print(f"{number} is an even number.")

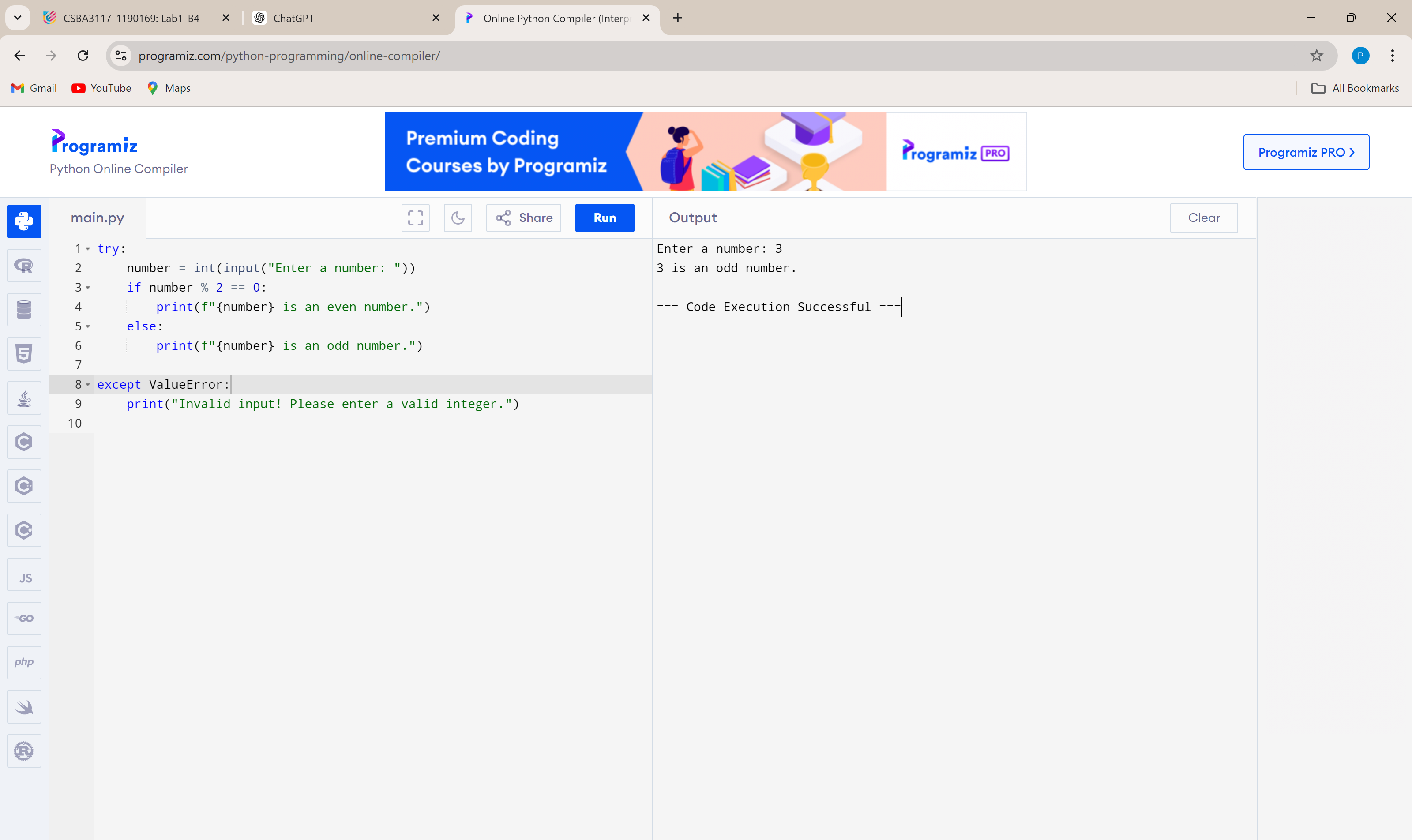
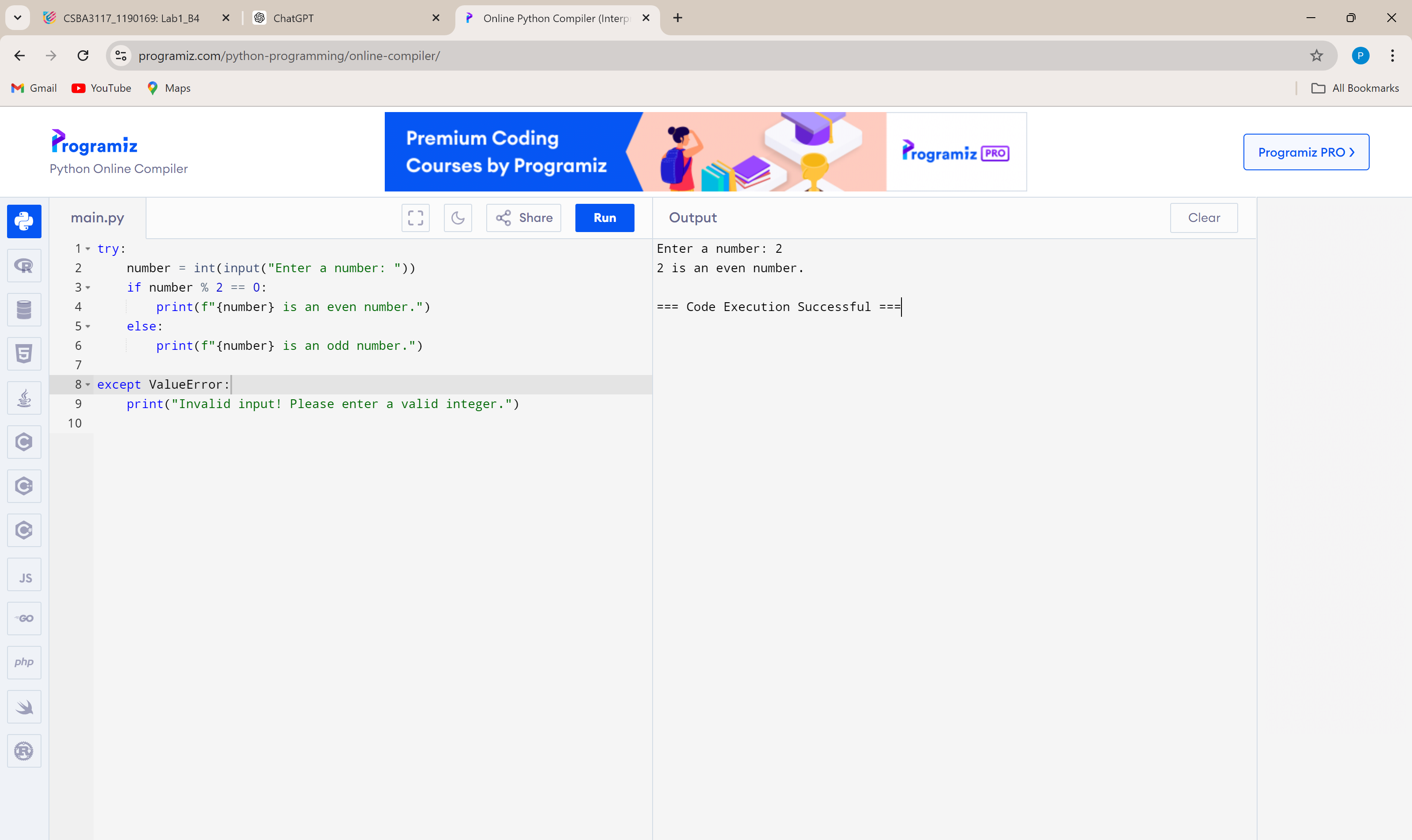
else:

print(f"{number} is an odd number.")

except ValueError:

print("Invalid input! Please enter a valid integer.")

* OUTPUT:



**Q5. Write a program to print the numbers from 1 to 10 using both for and while loops.**

* + Use a for loop to iterate through a range of numbers.
  + Use a while loop with a counter variable.
* CODE:

# Using a for loop

print("Using for loop:")

for i in range(1, 11):

print(i, end=" ")

print()

# Using a while loop

print("Using while loop:")

i = 1

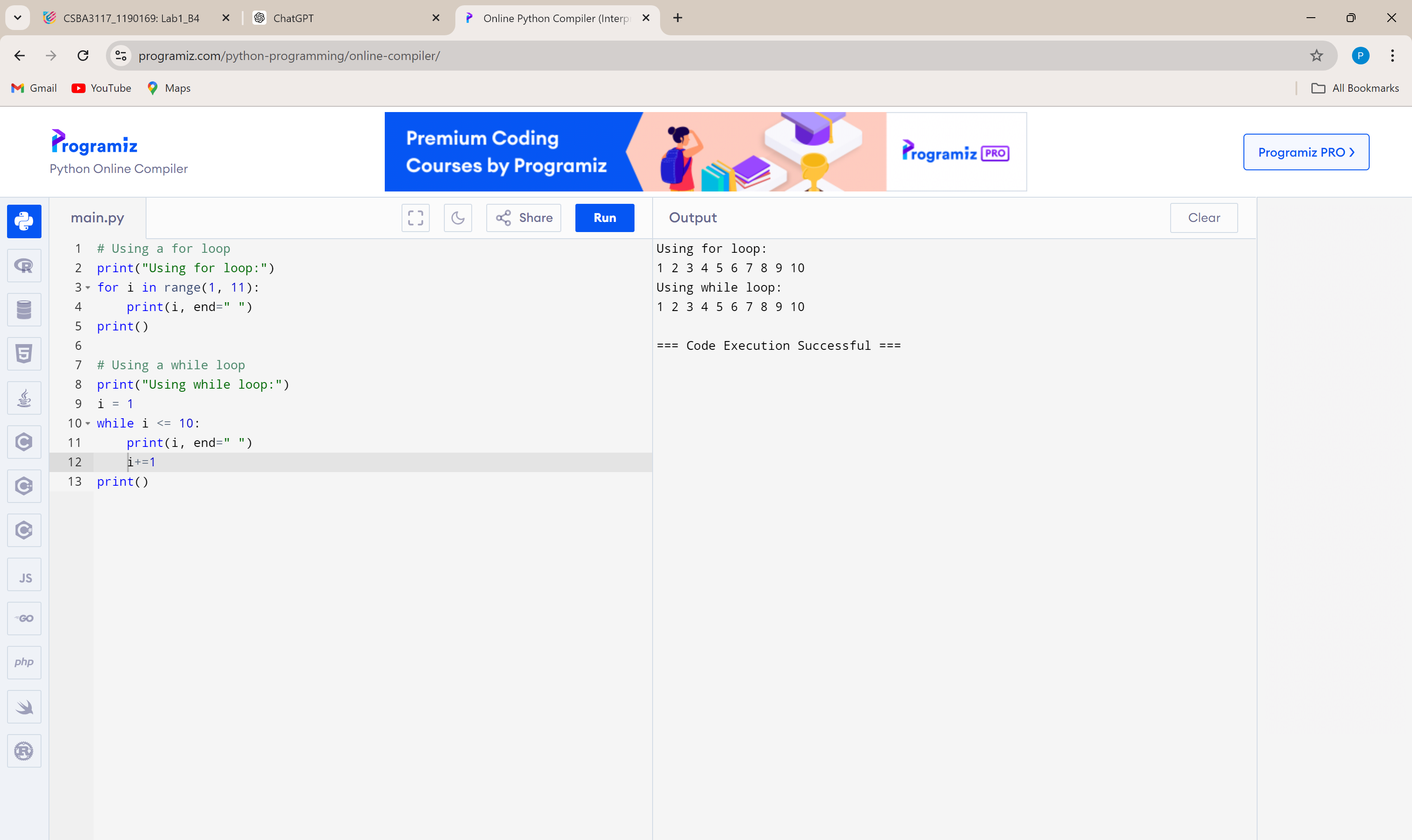
while i <= 10:

print(i, end=" ")

  i += 1

print()

* OUTPUT:



**Q6.  Create a list, access elements, modify elements, and perform list operations.**

* + Create a list of fruits.
  + Access elements using indexing.
  + Modify elements in the list.
  + Add and remove elements from the list.
  + Find the length of the list.
  + Sort the list in ascending and descending order.
* CODE:

# Create a list of fruits

fruits = ["apple", "banana", "cherry", "date"]

# Access elements using indexing

print(f"First fruit: {fruits[0]}")

print(f"Last fruit: {fruits[-1]}")

# Modify an element in the list

fruits[1] = "blueberry"

print(f"Modified list: {fruits}")

# Add and remove elements from the list

fruits.append("elderberry")

print(f"After adding a fruit: {fruits}")

fruits.remove("cherry")

print(f"After removing a fruit: {fruits}")

# Find the length of the list

print(f"Length of the list: {len(fruits)}")

# Sort the list in ascending and descending order

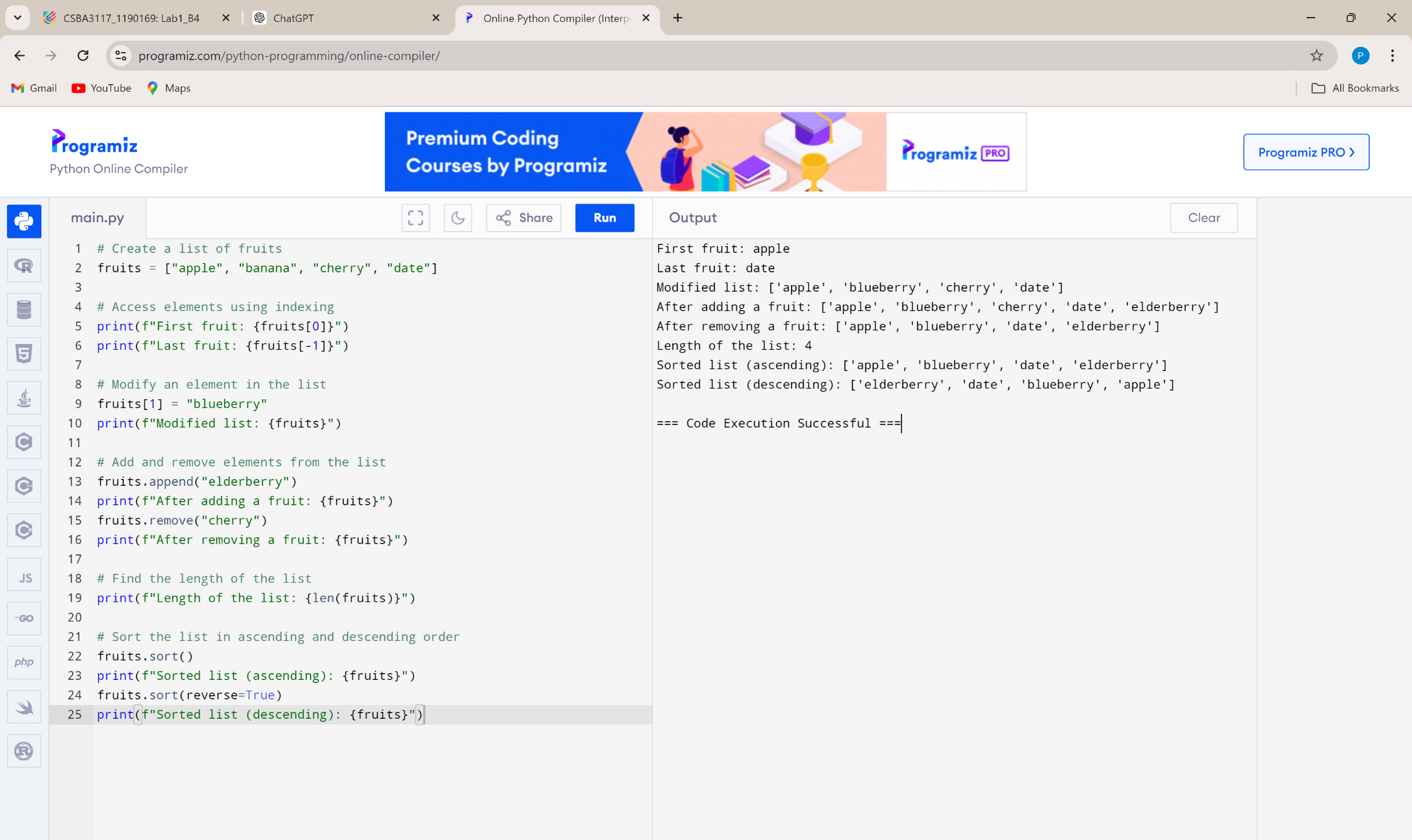
fruits.sort()

print(f"Sorted list (ascending): {fruits}")

fruits.sort(reverse=True)

print(f"Sorted list (descending): {fruits}")

* OUTPUT:



**Q7. Manipulate strings using various built-in functions.**

* + Create a string variable and find the length of the string.
  + Convert the string to uppercase and lowercase.
  + Check if a substring exists in the string.
  + Split the string into a list of words.
* CODE:

# Create a string variable

sentence = "Hello World, welcome to Python programming."

# Find the length of the string

print(f"Length of the string: {len(sentence)}")

# Convert the string to uppercase and lowercase

print(f"Uppercase: {sentence.upper()}")

print(f"Lowercase: {sentence.lower()}")

# Check if a substring exists in the string

substring = "Python"

if substring in sentence:

print(f"'{substring}' found in the string.")

else:

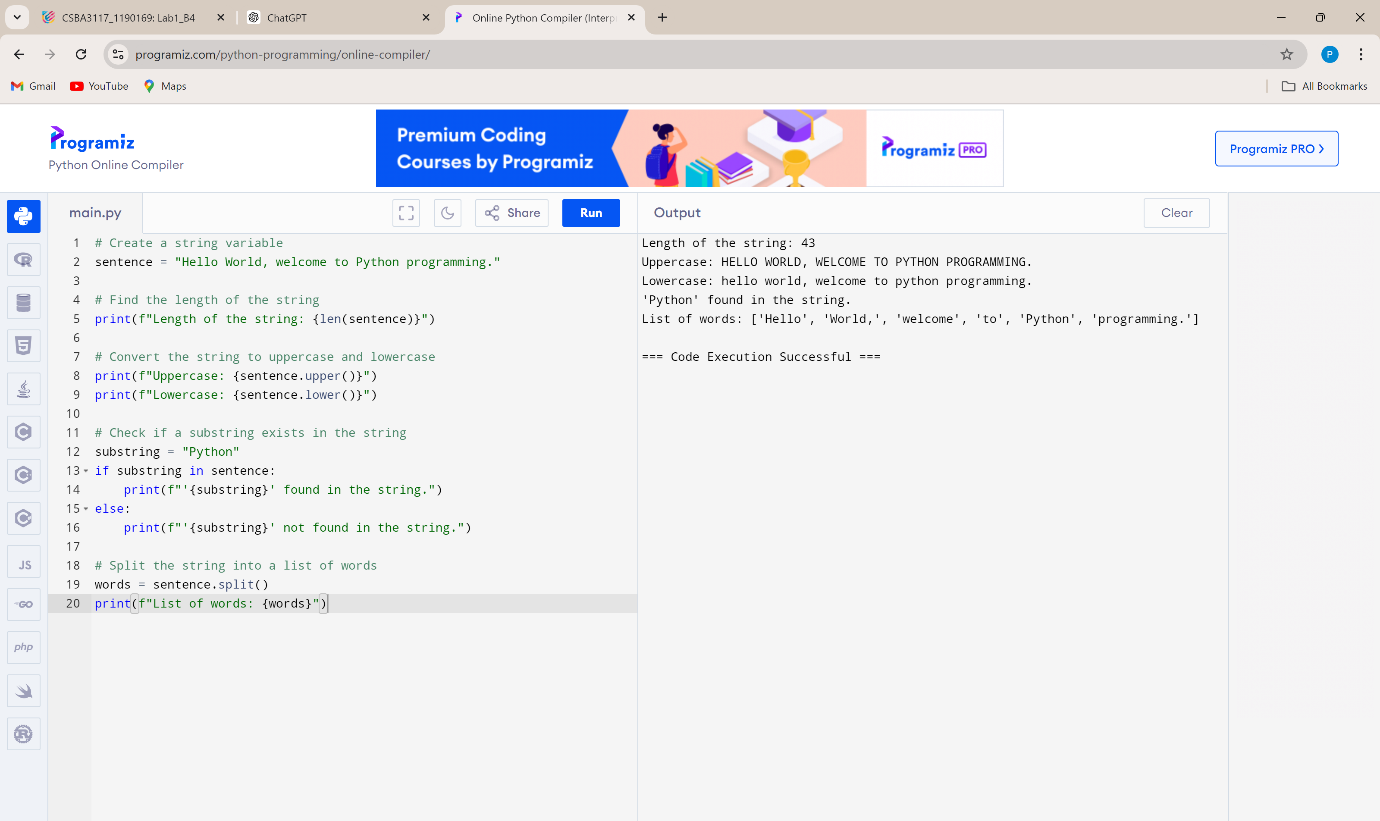
print(f"'{substring}' not found in the string.")

# Split the string into a list of words

words = sentence.split()

print(f"List of words: {words}")

* OUTPUT:



**Q8. Write a program to find the largest and smallest number in a list.**

CODE:

# Create a list of numbers

numbers = [10, 5, 20, 15, 2, 30]

# Find the largest and smallest numbers

largest = max(numbers)

smallest = min(numbers)

# Print the results

print(f"Largest number: {largest}")

print(f"Smallest number: {smallest}")

OUTPUT:

