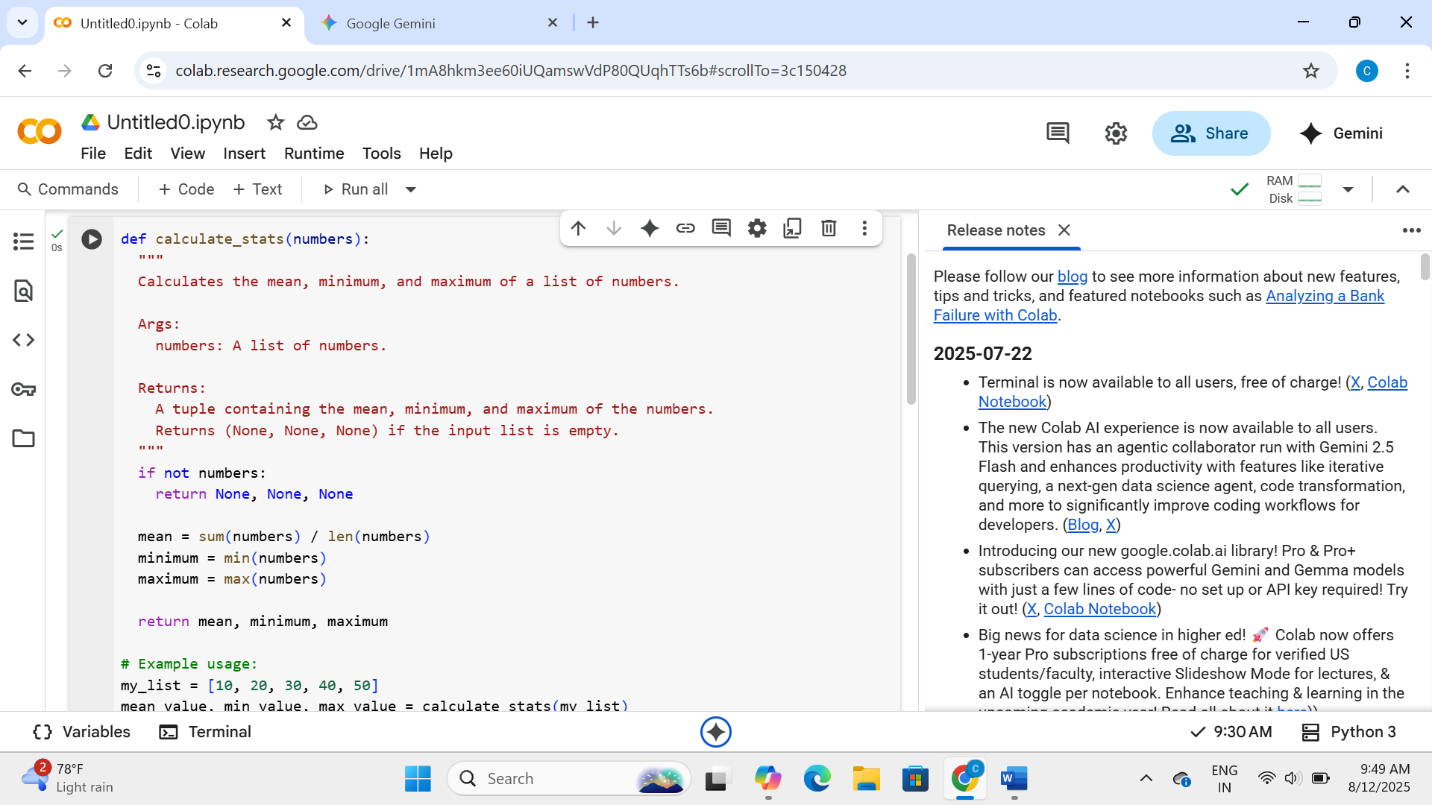
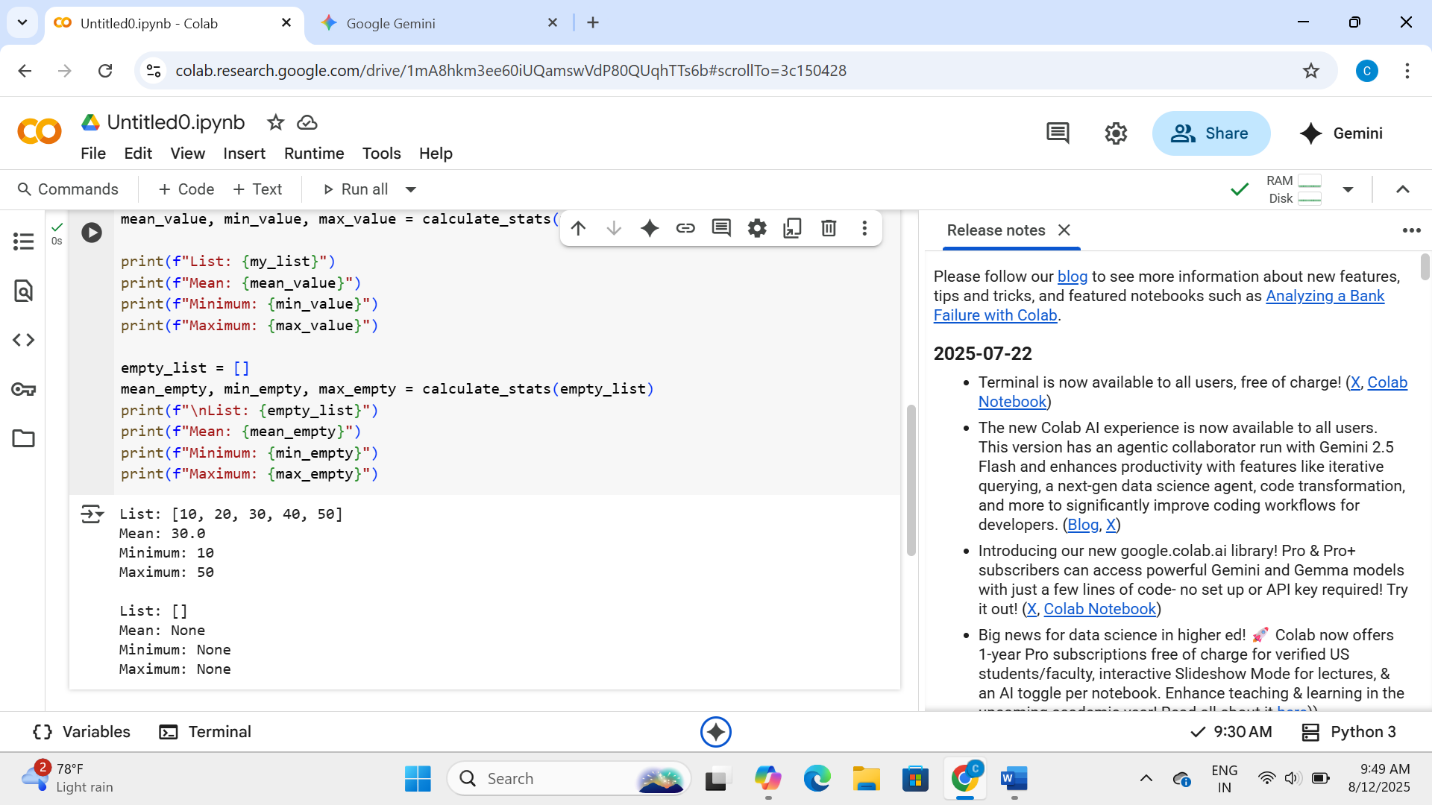
Assignment-2

Task-1

write a python function that reads a list of numbers and calculate the mean, minimum , maximum





1. **Average (Mean):** What's the typical number in the list? It adds up all the numbers and divides by how many there are.
2. **Smallest (Minimum):** What's the lowest number in the list?
3. **Largest (Maximum):** What's the highest number in the list?

The code defines something called a "function" that does these calculations for you. You give it your list of numbers, and it gives you back the average, smallest, and largest numbers.

It also includes a little check: if you give it an empty list (a list with no numbers), it knows it can't calculate anything and just tells you that (by returning None).

Think of it like this: you put a list of numbers into a box (calculate\_ stats), and the box gives you back three results: the average, the smallest, and the largest number from that list.

Task-2

Write a python function that checks whether a number is armstrong or not

def is\_armstrong(number):

"""

Check if a number is an Armstrong number.

Args:

number (int): The number to check.

Returns:

bool: True if Armstrong number, False otherwise.

"""

num\_str = str(number)

num\_digits = len(num\_str)

total = sum(int(digit) \*\* num\_digits for digit in num\_str)

return total == number

print(is\_armstrong(153)) # True

print(is\_armstrong(9474)) # True

print(is\_armstrong(123)) # False

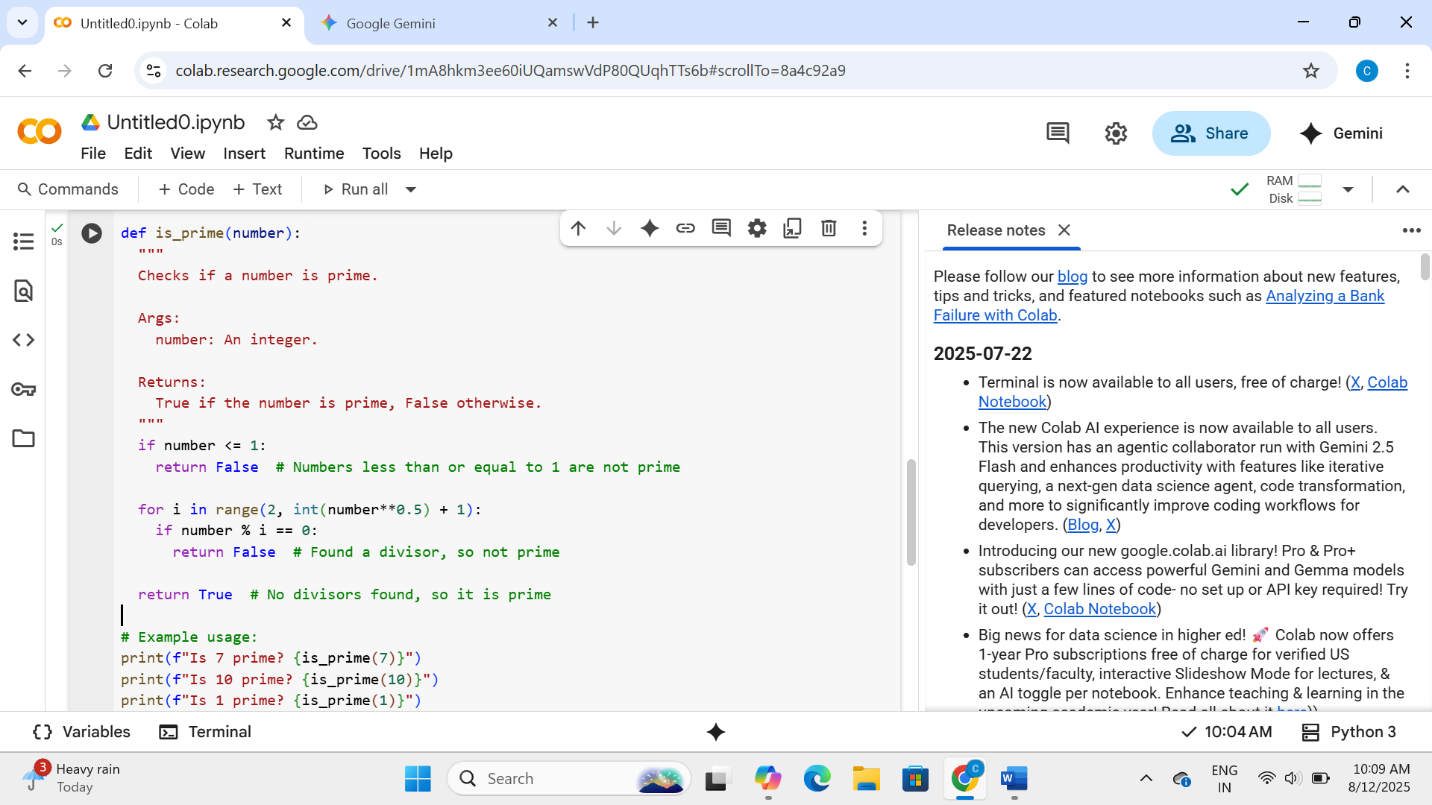
What’s an Armstrong Number?

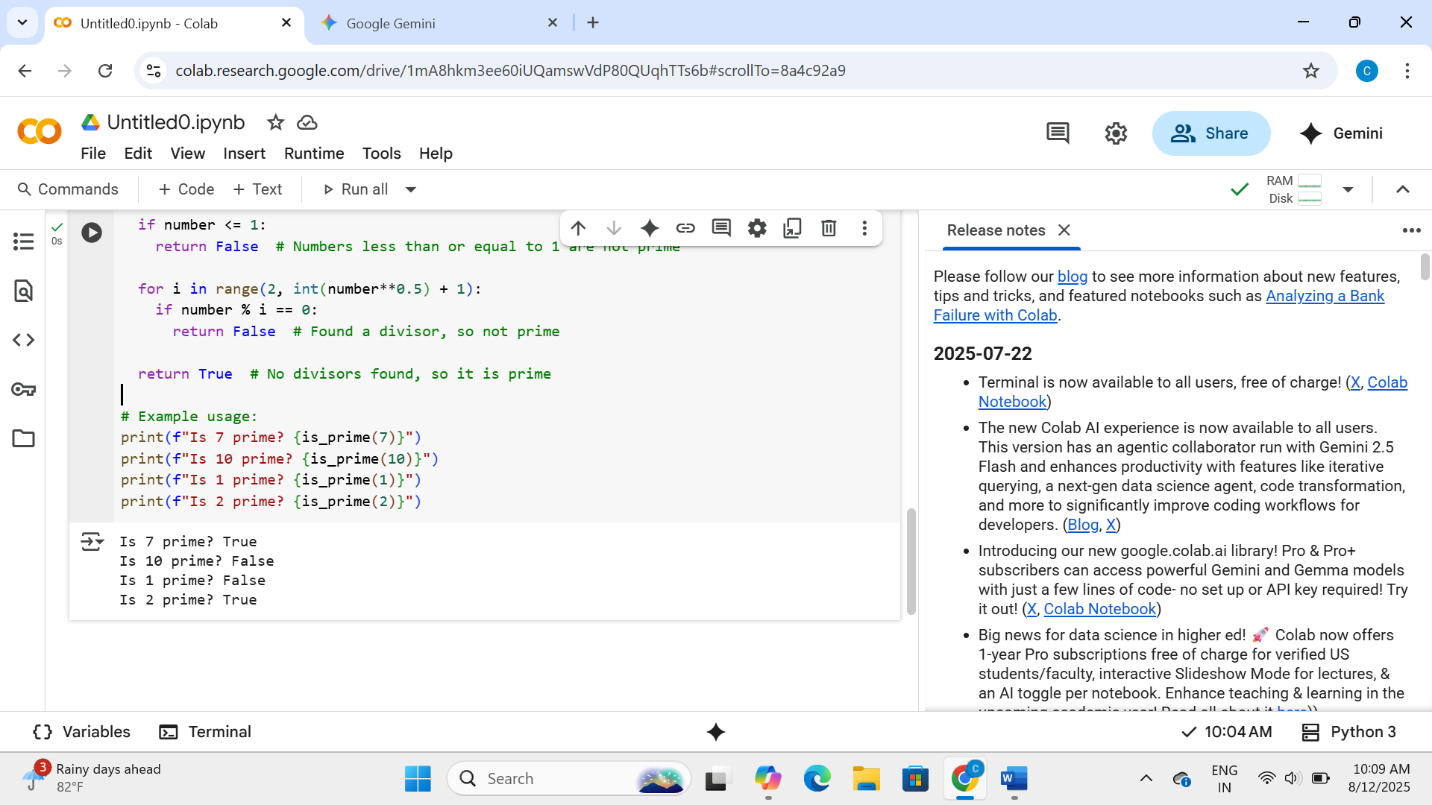
A number is an **Armstrong number** if:

The sum of its digits, each raised to the power of the number of digits, equals the number itself.

Example:  
**153** has 3 digits  
→ 1^3 + 5^3 + 3^3 = 153  
So, 153 is an Armstrong number.

Task-3





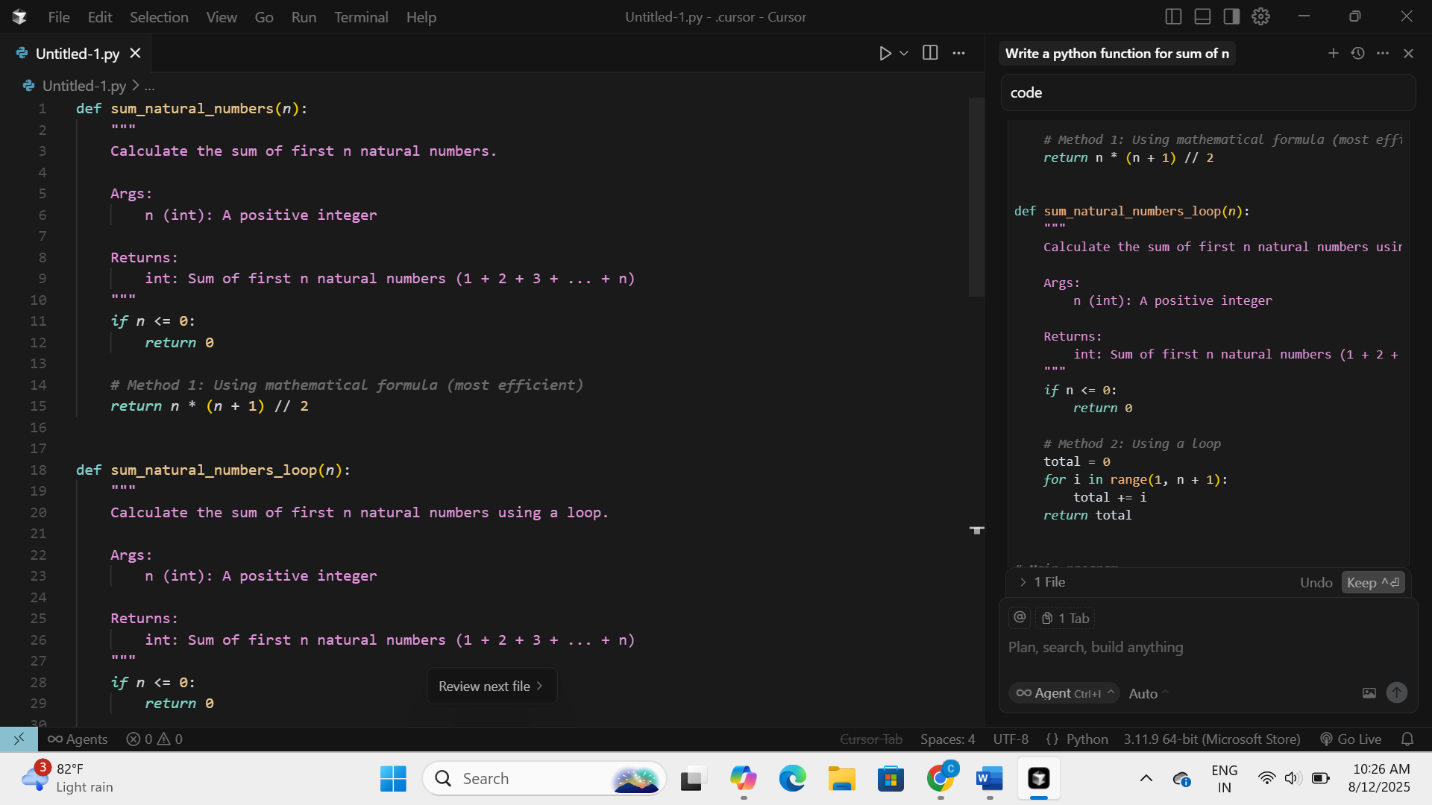
The code is a function called is\_prime. Its job is to look at a number and tell you "True" if it's a prime number or "False" if it's not.

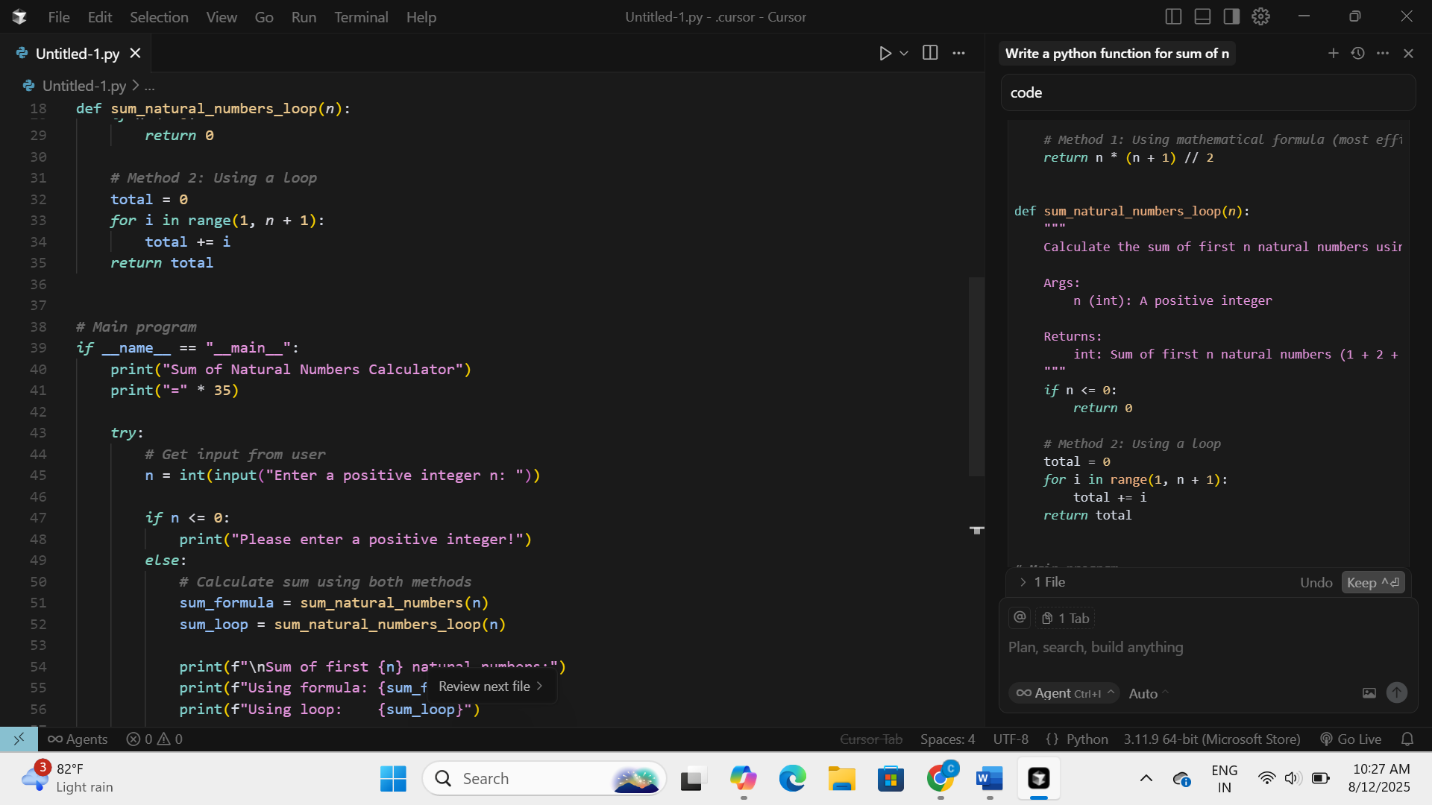
* **What is a prime number?** It's a whole number greater than 1 that can only be divided evenly by 1 and itself. (Like 2, 3, 5, 7, 11, etc.)
* **How does the code check?**
  + First, it quickly checks if the number is 1 or less. Those are definitely not prime.
  + Then, it tries dividing the number by other numbers, starting from 2.
  + It only needs to check up to a certain point (the square root) because of a math rule.
  + If it finds any number (other than 1 or the number itself) that divides it evenly, it knows it's *not* prime and says "False".
  + If it tries all the relevant numbers and doesn't find any that divide it evenly, then it knows it *is* prime and says "True".

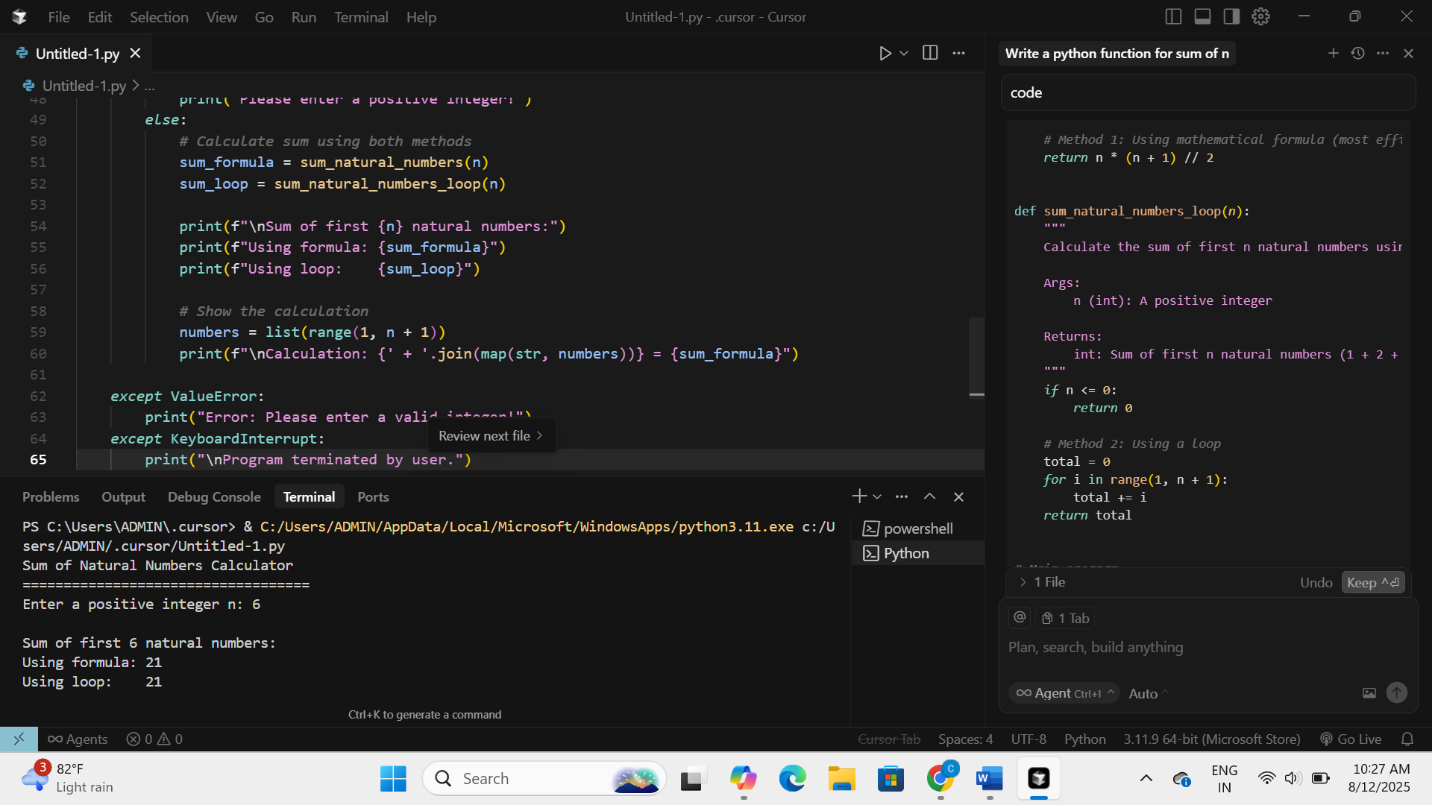
So, in short, the code is a simple test to see if a number is prime by checking if it has any divisors other than 1 and itself.

Task-4

Write a python program to calculate sum of N natural numbers







Task-5

