1. What is a lambda function in Python, and how does it differ from a regular function?

Ans:- A lambda function in Python is a small, anonymous, and inline function that can have any number of arguments, but can only have one expression. It is defined using the lambda keyword, followed by the arguments and the expression. Lambda functions are often used when we need a simple function for a short period of time .

Here are some key points about lambda functions in Python:

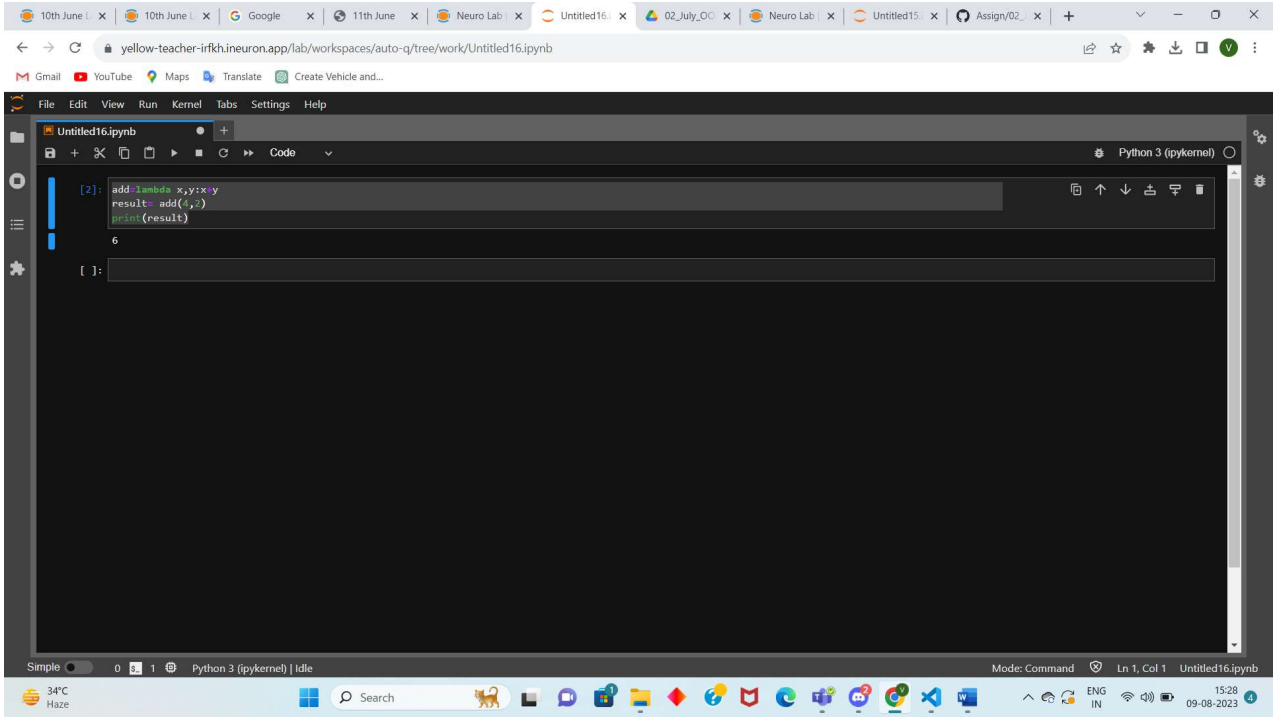
1. Lambda functions are small, anonymous functions defined with the lambda keyword.
2. They can take any number of arguments but can only have one expression.
3. The expression is evaluated and returned when the function is called.
4. Lambda functions do not require a return statement, the expression is implicitly returned.
5. You can't include statements like loops, if or else in lambda functions; only expressions are allowed.
6. Lambda functions are useful for small tasks that are not reused throughout your code.
7. They can be assigned to variables and used like regular functions.

Difference :- Lambda functions are typically more concise & are defined in a single line . Regular functions have more structured syntax with a formal header & the use of return keywords.

Lambda functions can only contain a single expression, while regular functions can have multiple statement & more complex logic.

2. Can a lambda function in Python have multiple arguments? If yes, how can you define and use them?

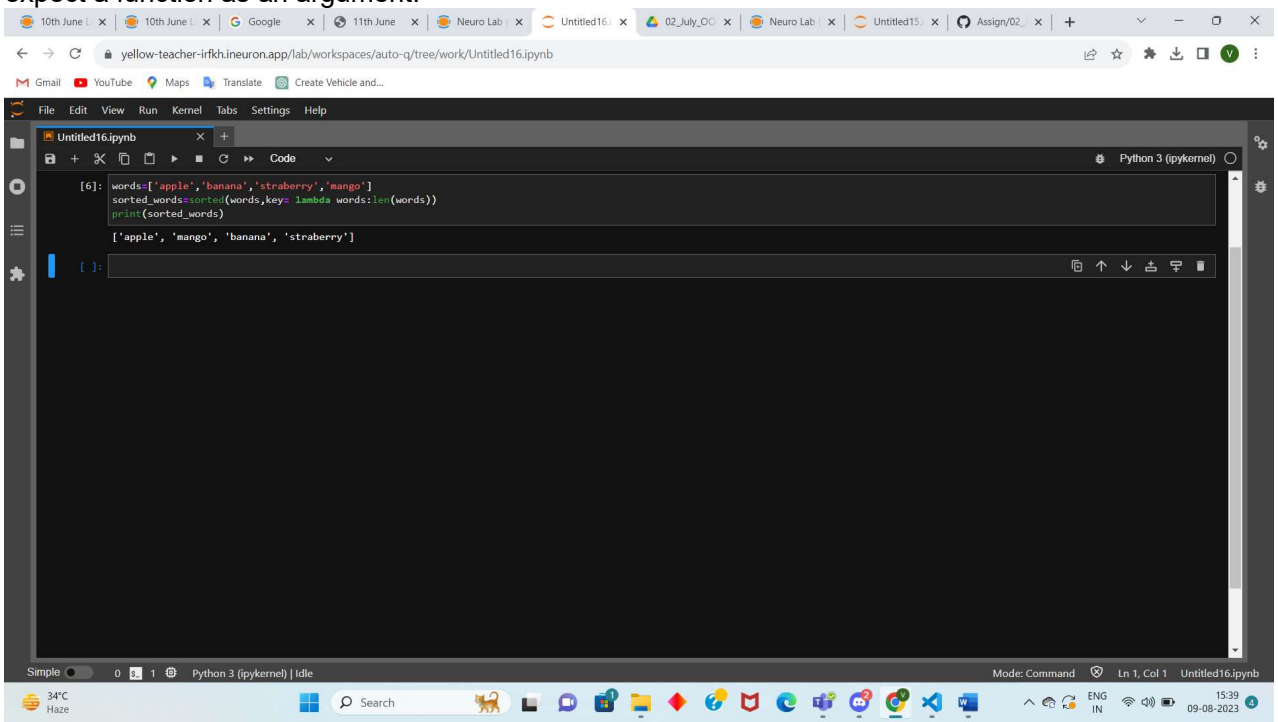
Ans:- Yes, a lambda function in Python can have multiple arguments. The syntax for defining a lambda function with multiple arguments is as follows:



3. How are lambda functions typically used in Python? Provide an example use case.

Ans:-

Lambda functions are typically used in Python for situations where we need a small, anonymous function to perform a specific task, especially when We are working with higher-order functions that expect a function as an argument.



4. What are the advantages and limitations of lambda functions compared to regular functions in Python?

Ans -Advantages of lambda functions in python:-

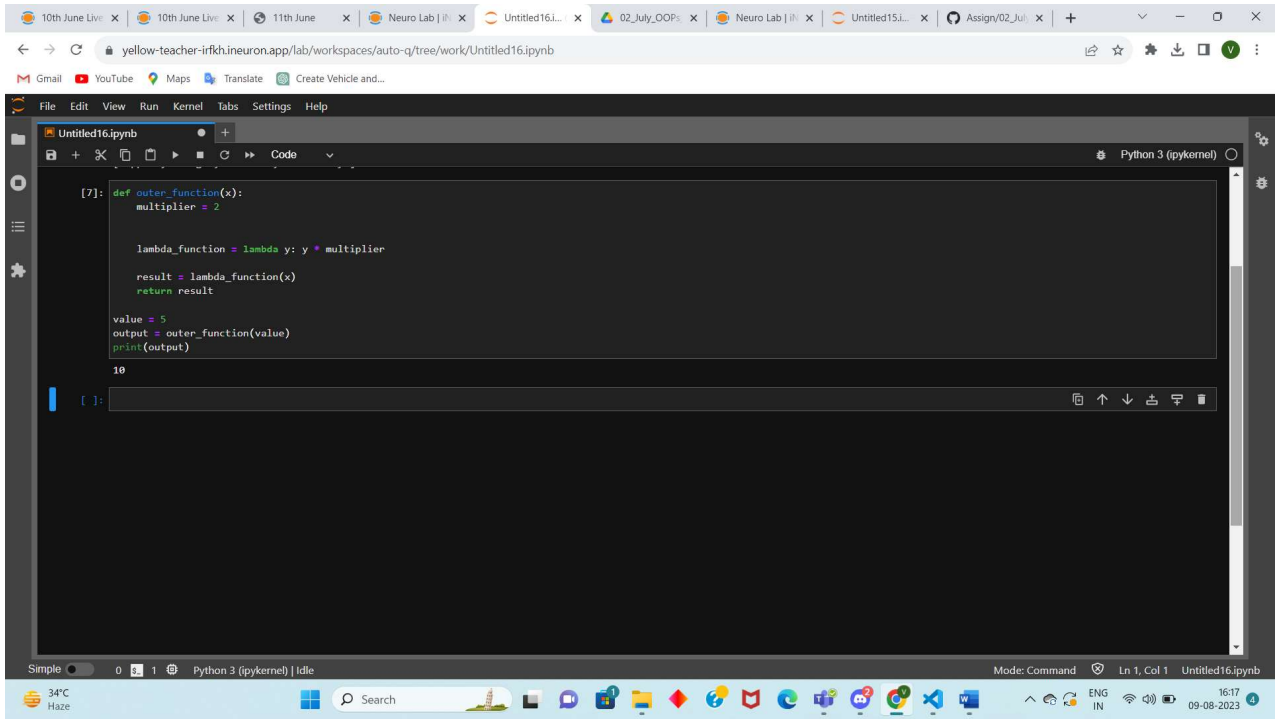
a. This is particularly useful for short and straightforward operations.
b. They return automatically.
c. They can't have a docstring and they don't have a name.

Limitations of lambda functions:-

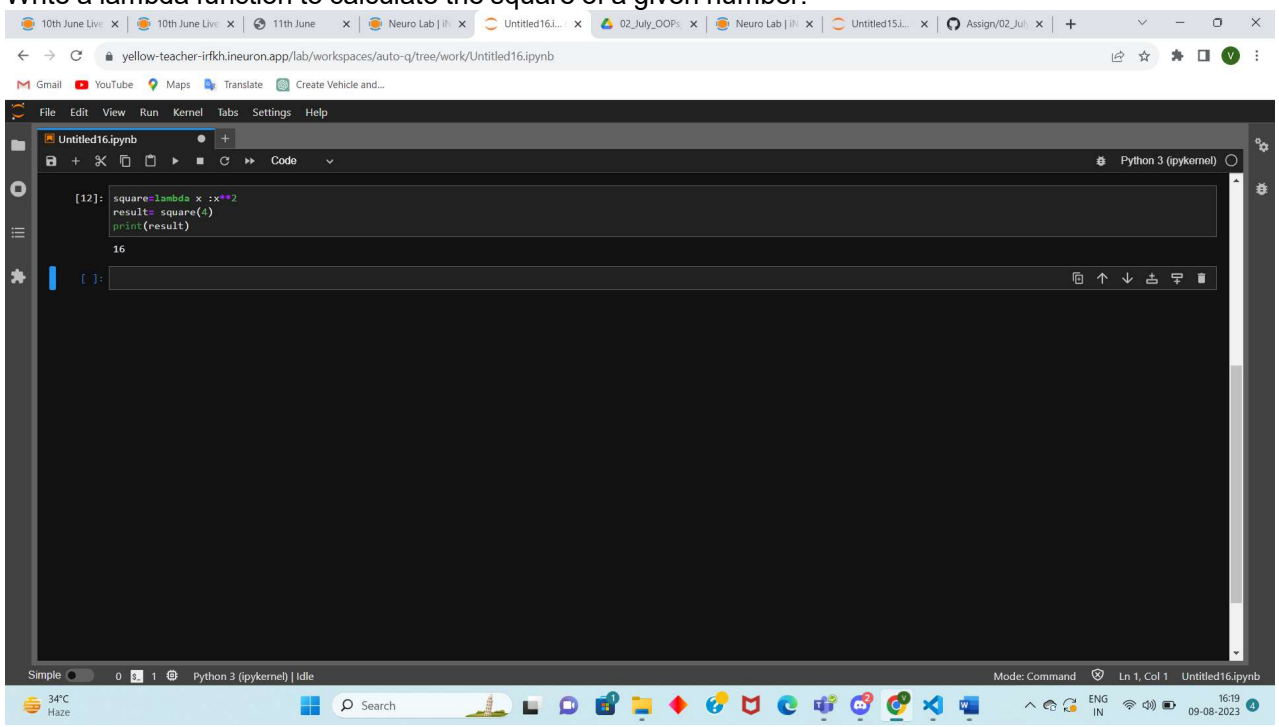
a. Lambda functions can only consist of a single expression, which means they are not suitable for more complex logic that requires multiple statements or control structures.
b. Lambda functions do not support docstrings, which are important for providing information about a function's purpose, parameters, and usage.
c. Due to their anonymous nature, lambda functions are less reusable compared to named functions.

5. Are lambda functions in Python able to access variables defined outside of their own scope? Explain with an example.

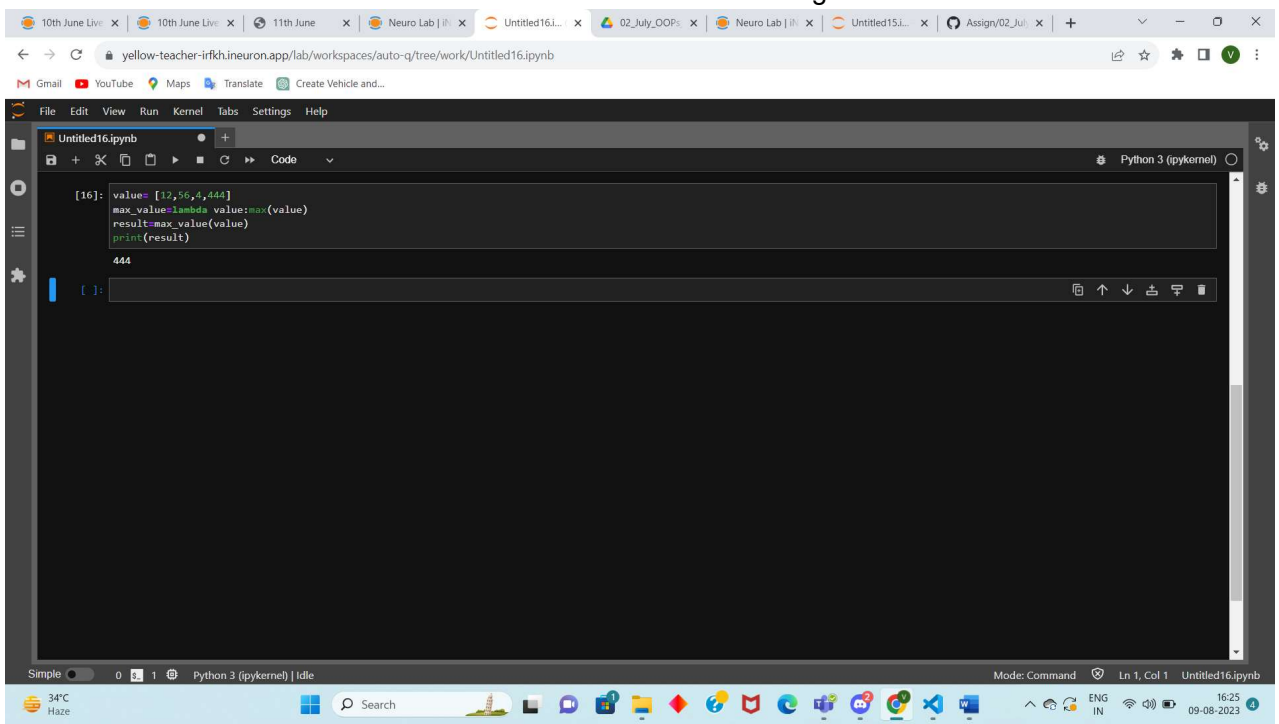
Ans:- Yes, lambda functions in Python are able to access variables defined outside of their own scope.



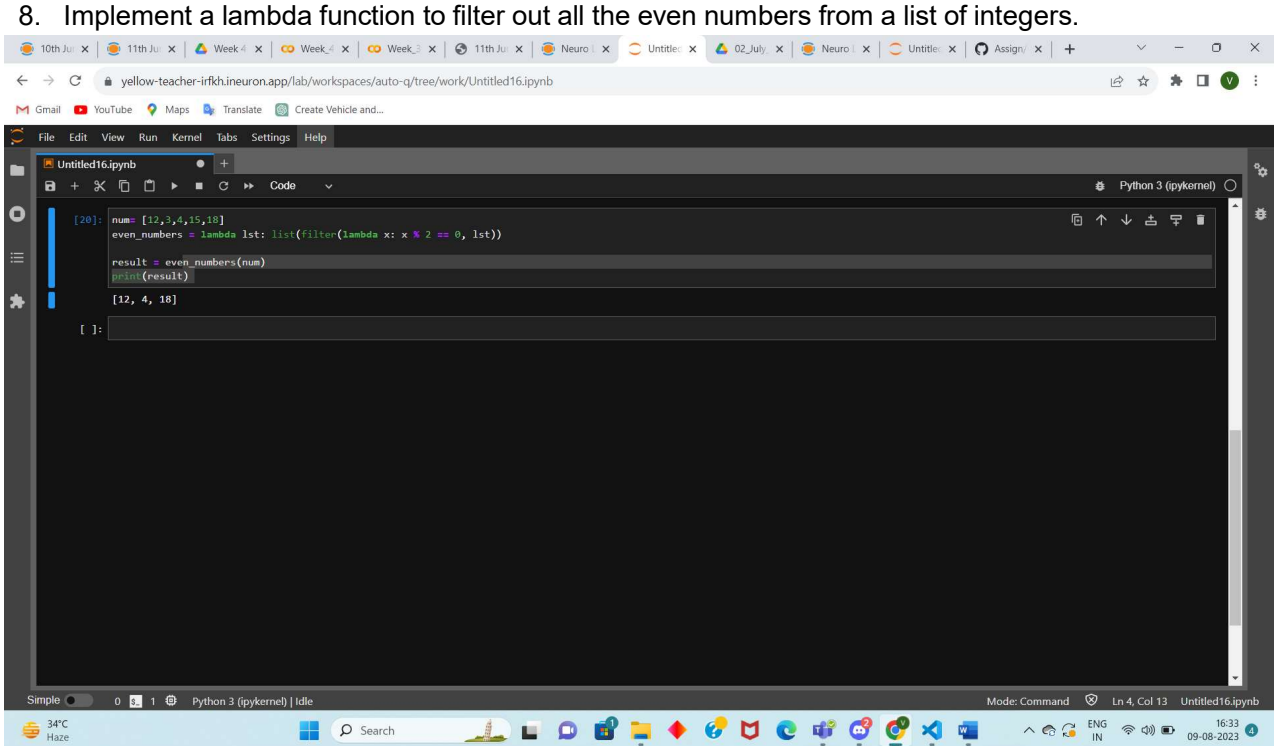
6. Write a lambda function to calculate the square of a given number.



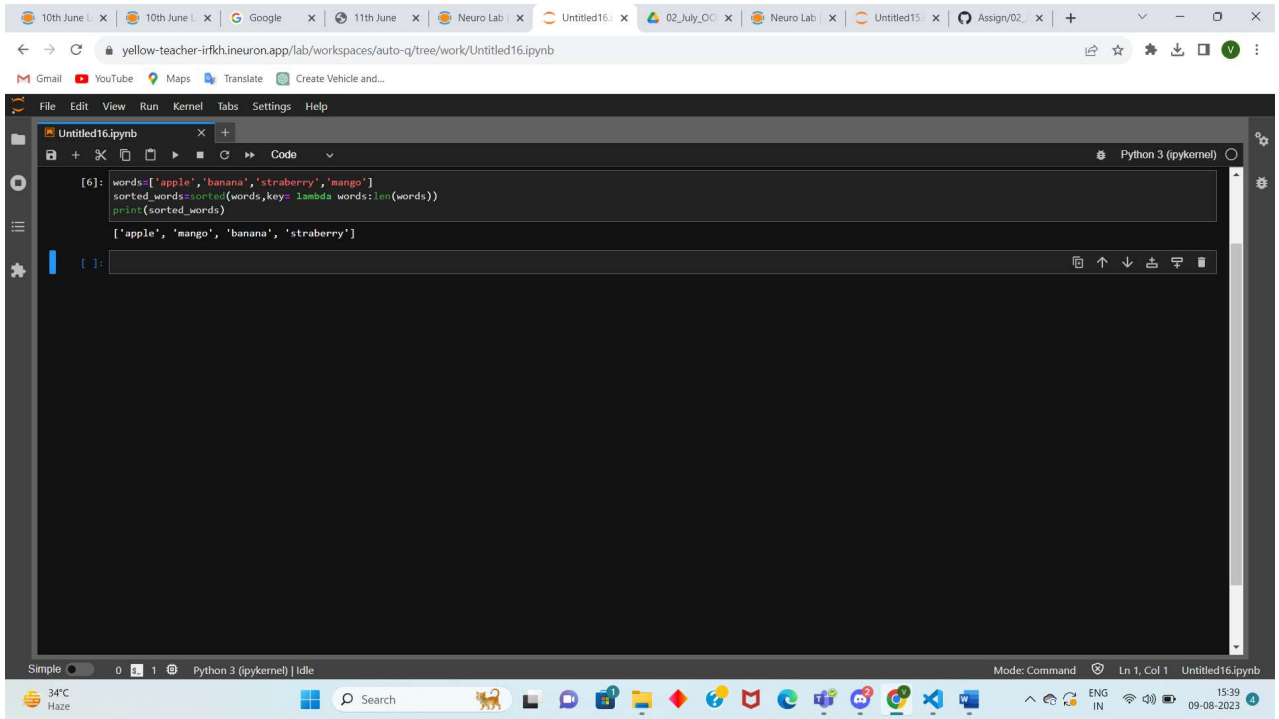
7. Create a lambda function to find the maximum value in a list of integers.



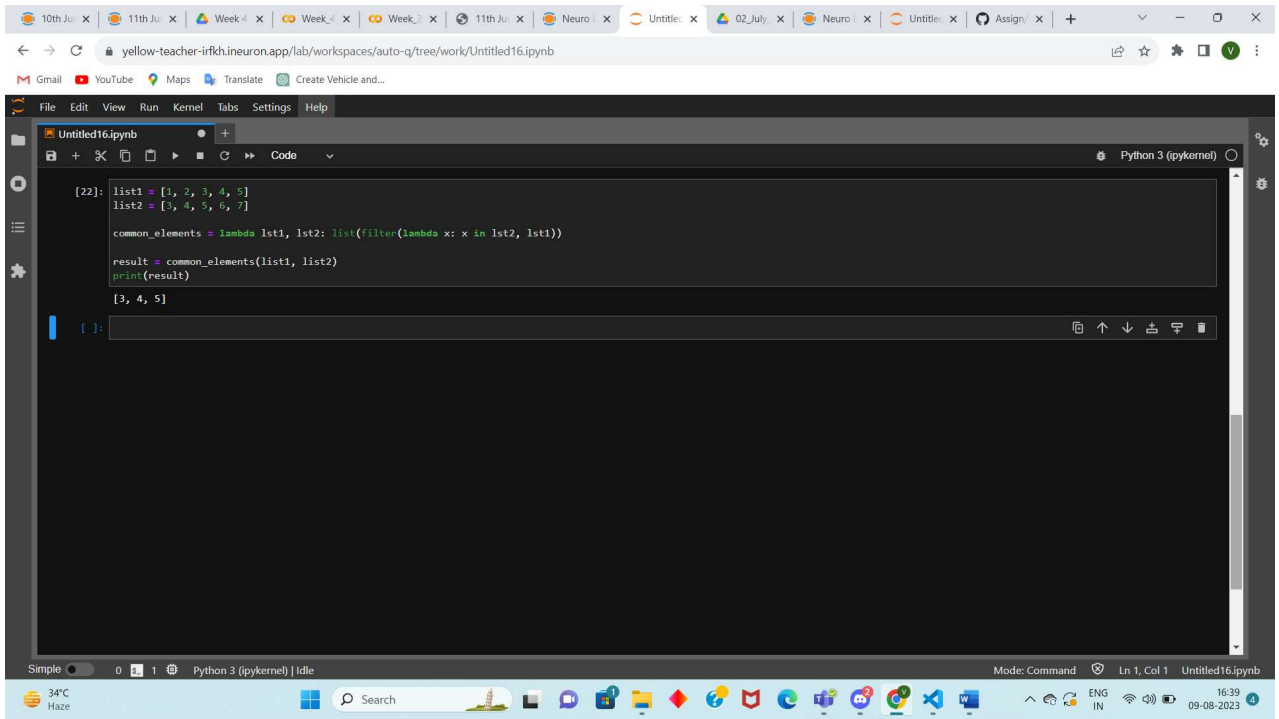
8. Implement a lambda function to filter out all the even numbers from a list of integers.



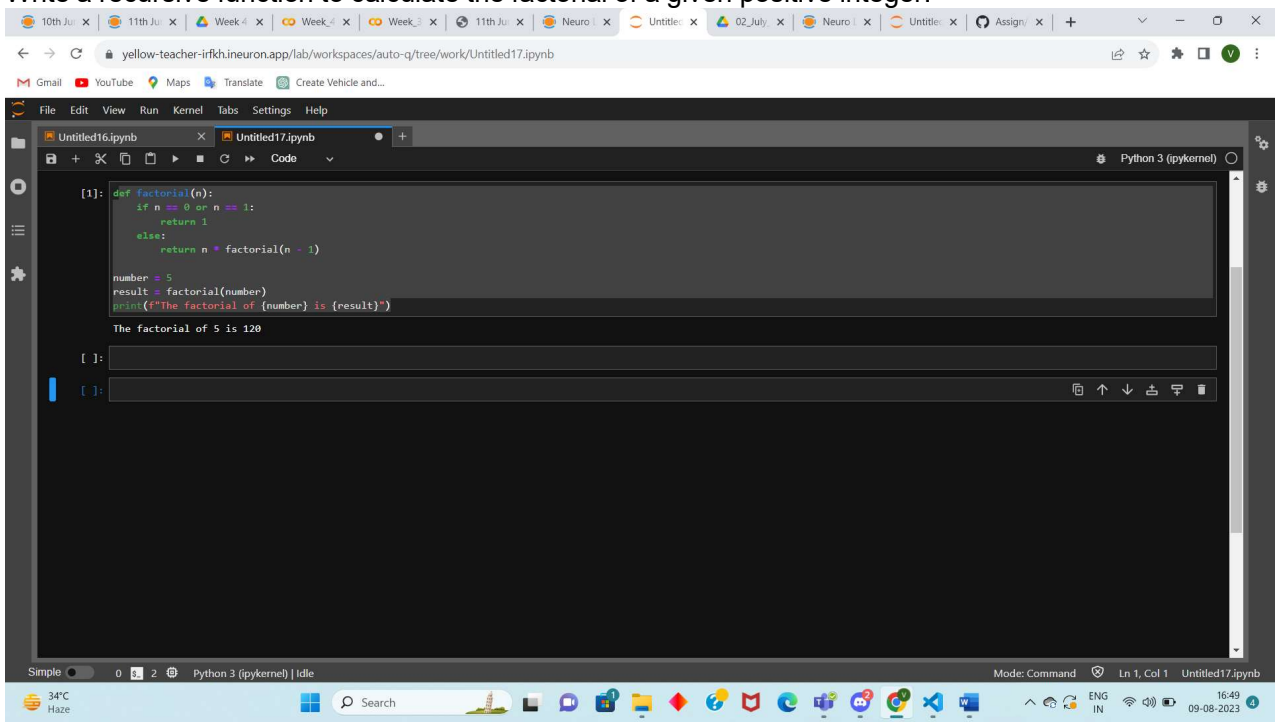
9. Write a lambda function to sort a list of strings in ascending order based on the length of each string.



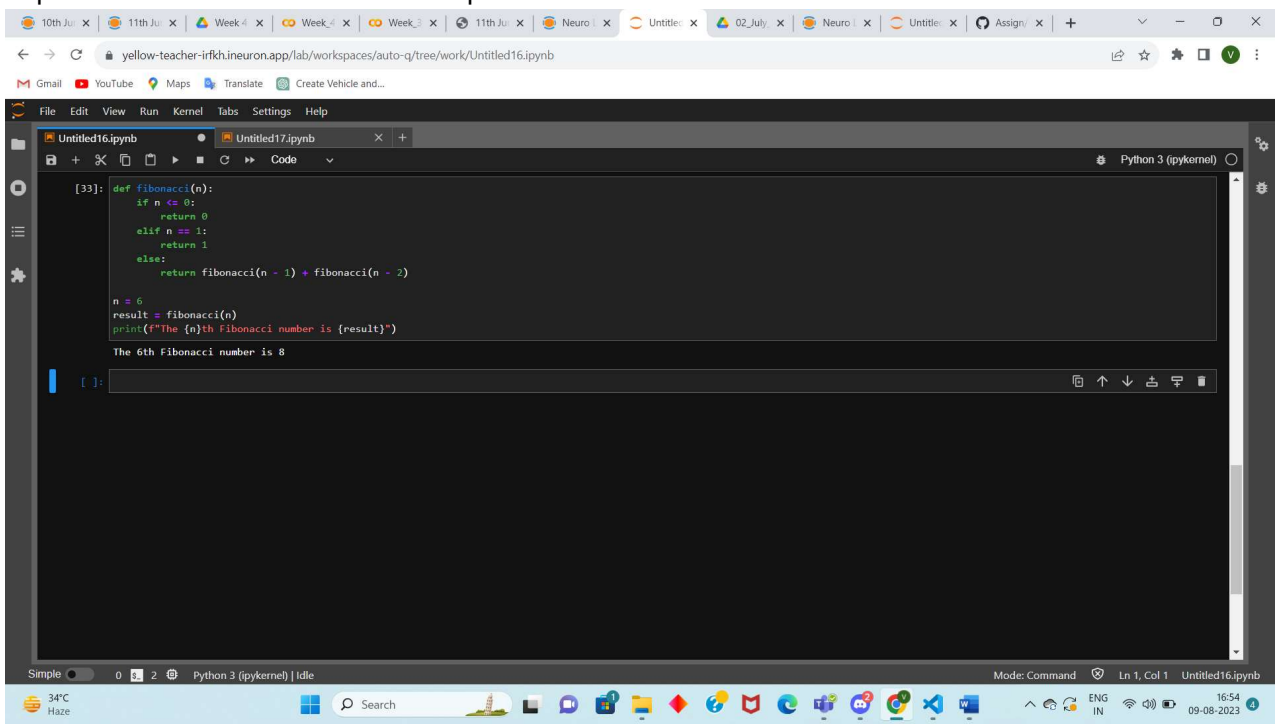
10. Create a lambda function that takes two lists as input and returns a new list containing the common elements between the two lists.



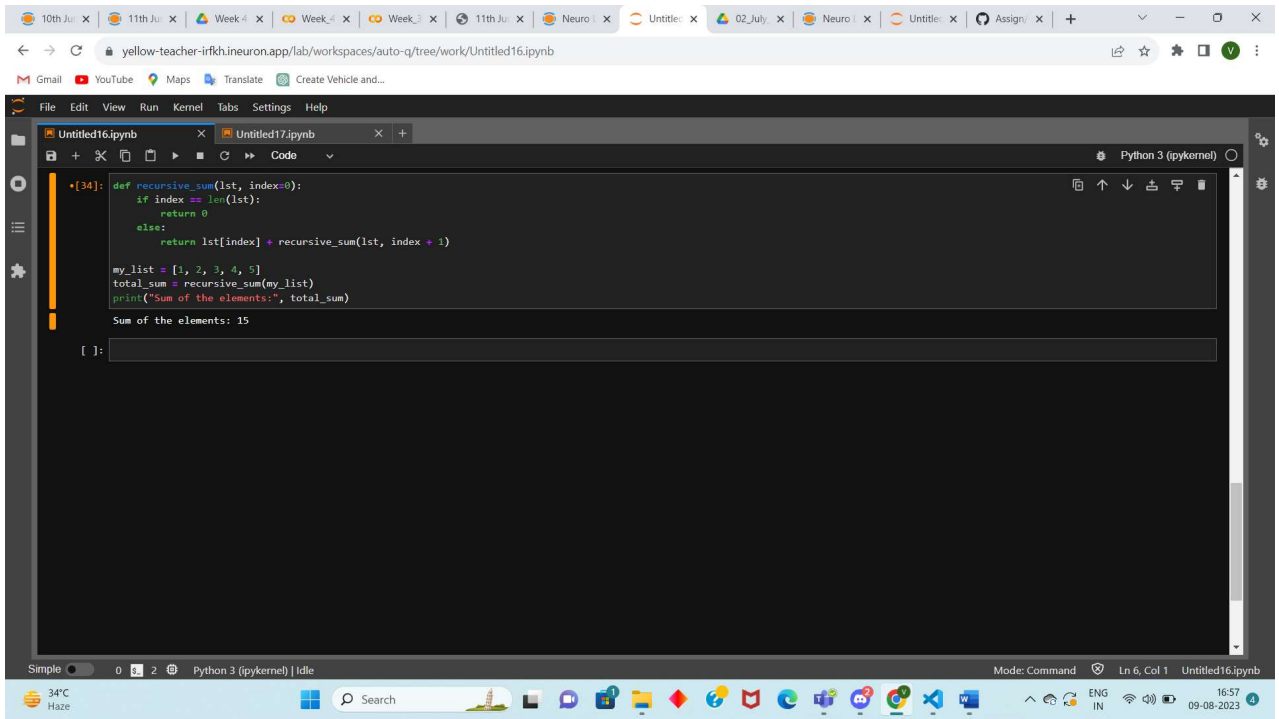
11. Write a recursive function to calculate the factorial of a given positive integer.



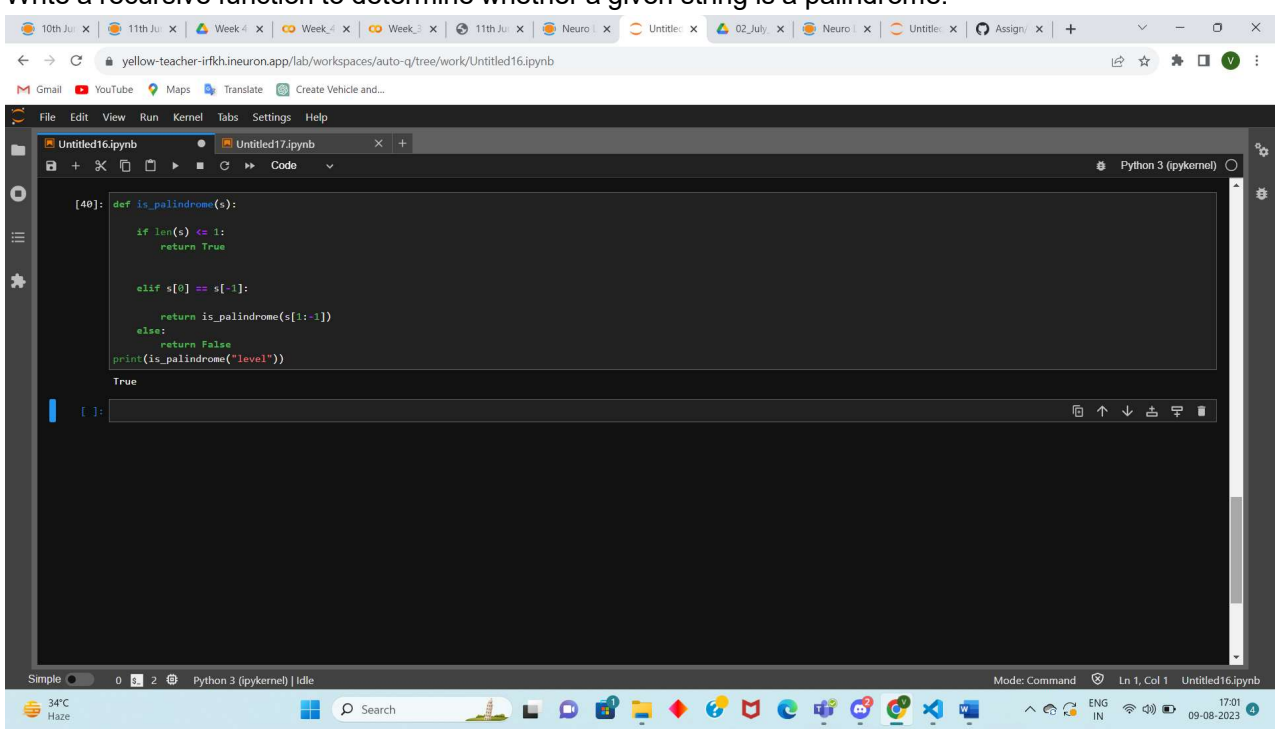
12. Implement a recursive function to compute the nth Fibonacci number.



13. Create a recursive function to find the sum of all the elements in a given list.



14. Write a recursive function to determine whether a given string is a palindrome.



15. Implement a recursive function to find the greatest common divisor (GCD) of two positive integers.