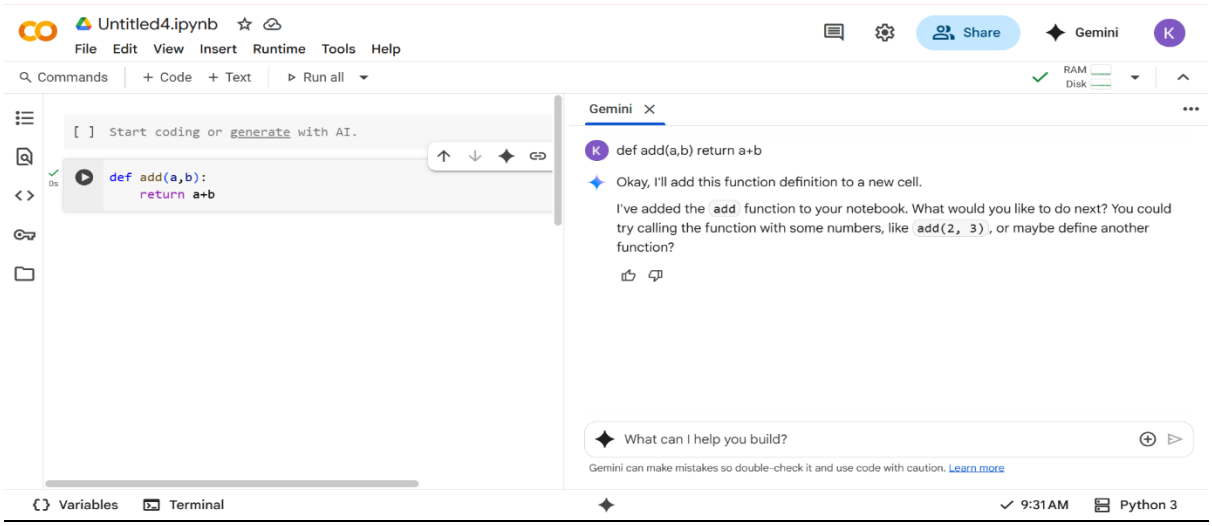
Assignment-7

2403A52118

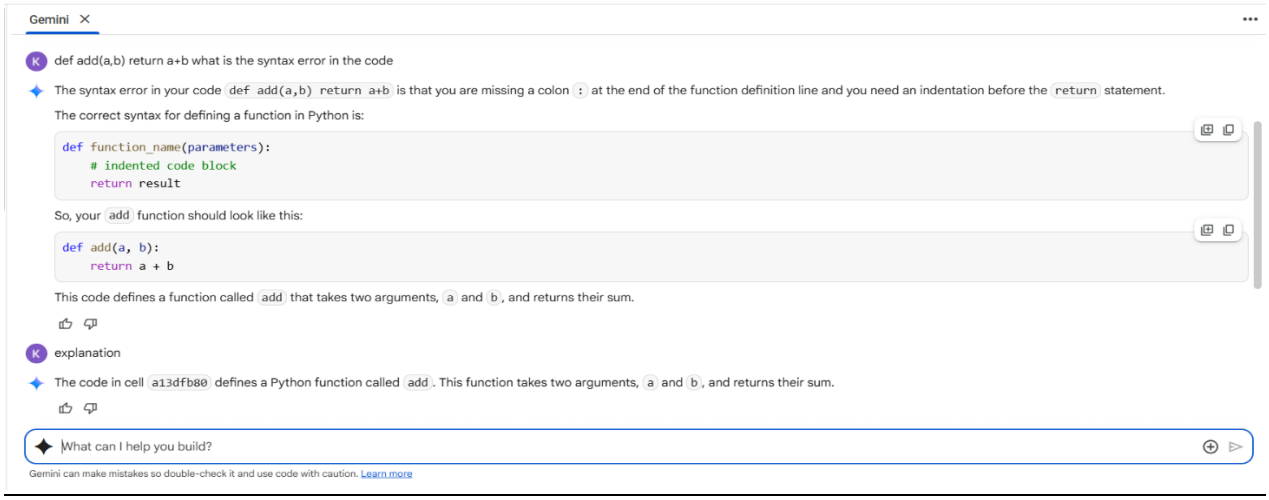
Task 1:

```python
def add(a, b)
    return a + b
```

Code and Output:



Explanation:

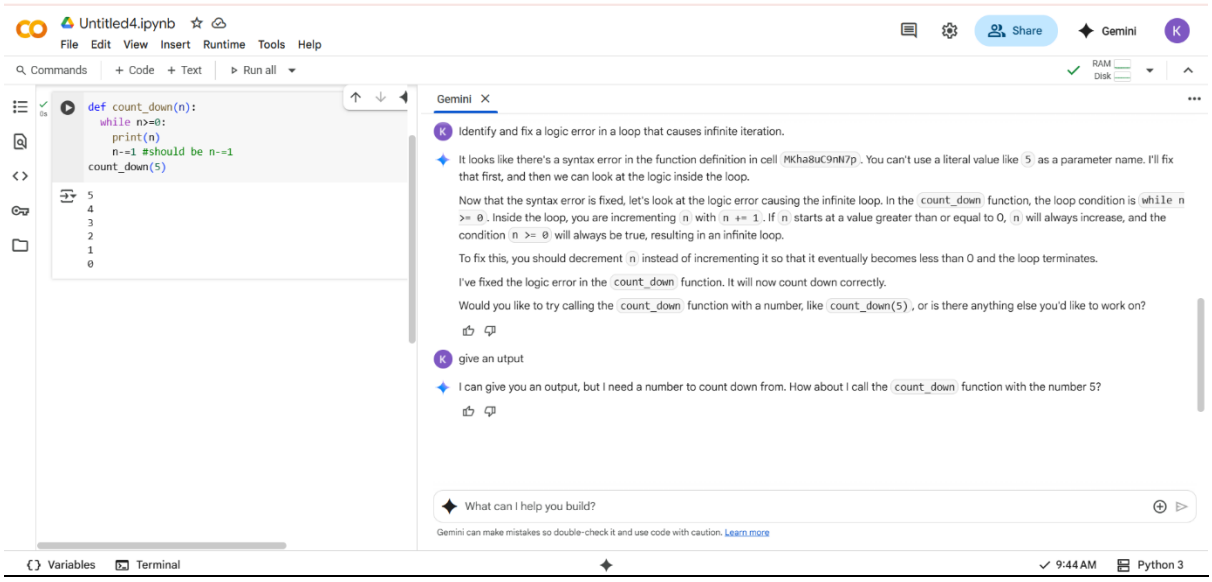


Task 2:

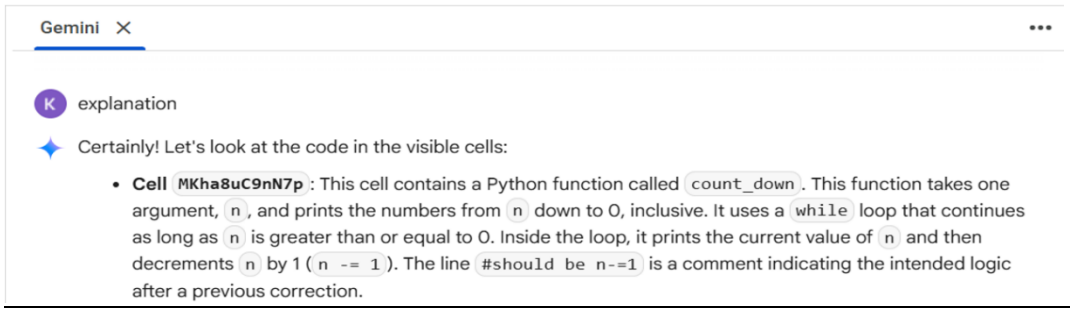
```python
def count_down(n):
    while n >= 0:
        print(n)
        n += 1  # Should be n -= 1
```

Identify and fix a logic error in a loop that causes infinite iteration.

Code and Output:



Explanation:

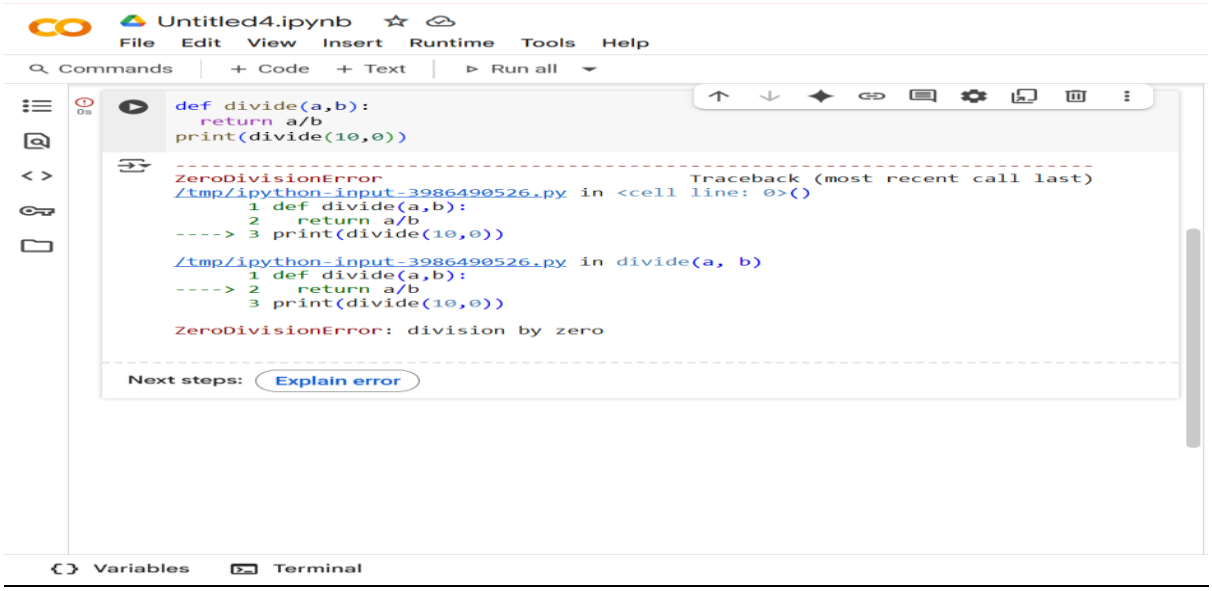


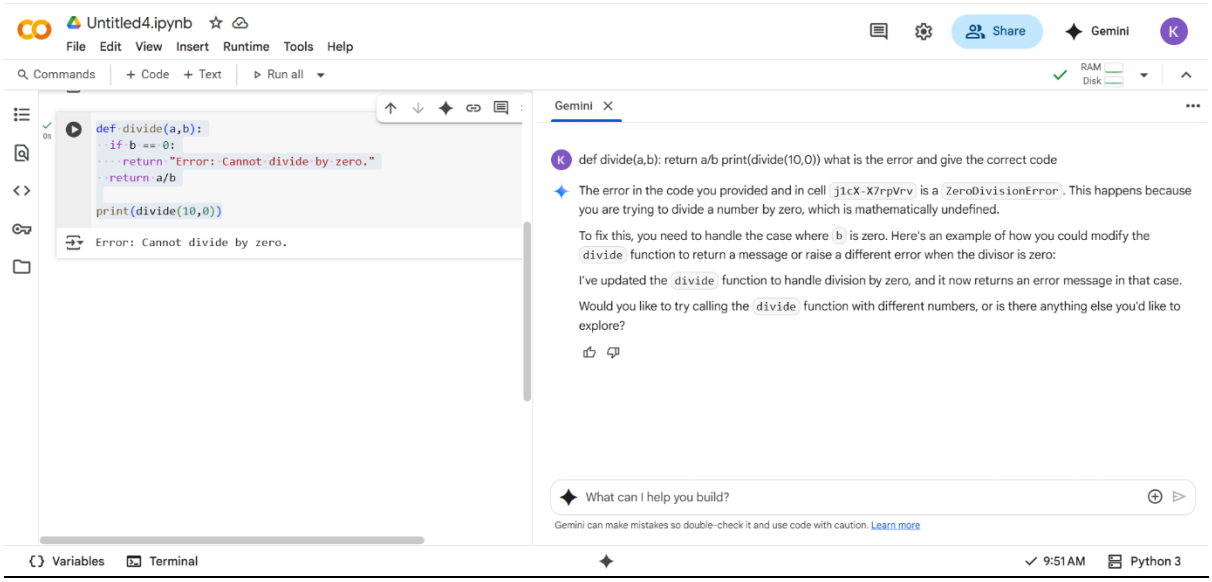
Task 3:

```
# Debug the following code
def divide(a, b):
    return a / b

print(divide(10, 0))
```

Code and Output:





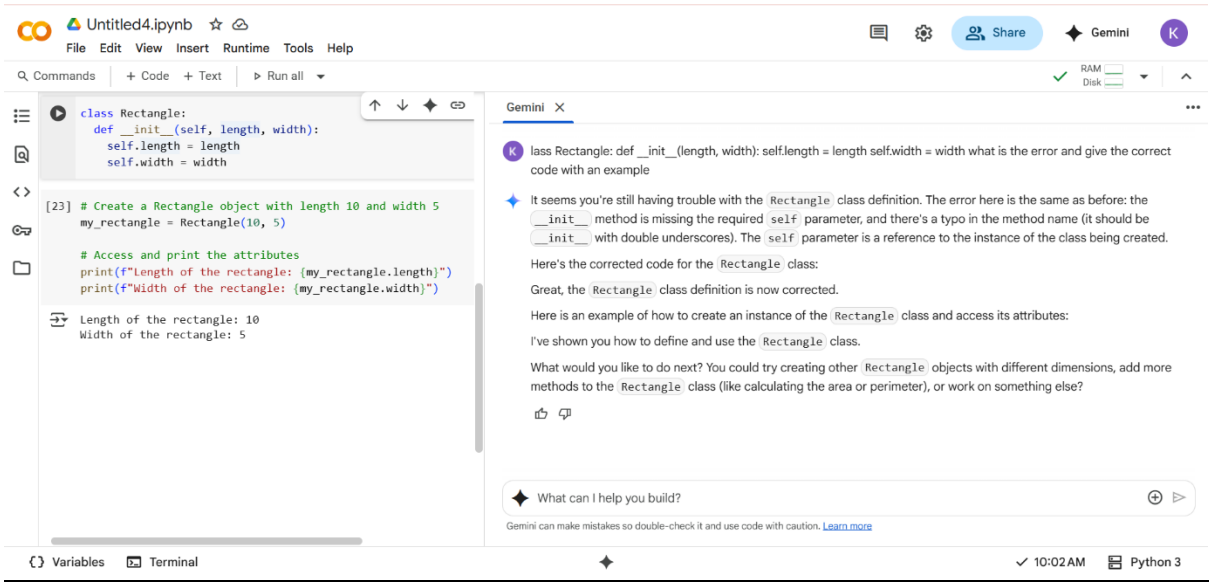
Explanation:

- **Cell** `j1cX-X7rpVrv` : This cell contains a Python function called `divide`. This function takes two arguments, `a` and `b`. It includes an `if` statement to check if `b` is equal to 0. If it is, the function returns the string "Error: Cannot divide by zero." to prevent a `ZeroDivisionError`. If `b` is not zero, the function returns the result of dividing `a` by `b` (`a / b`). The `print(divide(10,0))` line calls this function with `a=10` and `b=0` and prints the returned value, which will be the error message.

Task 4:

```python
class Rectangle:
    def __init__(length, width):
        self.length = length
        self.width = width
```

Code and output:



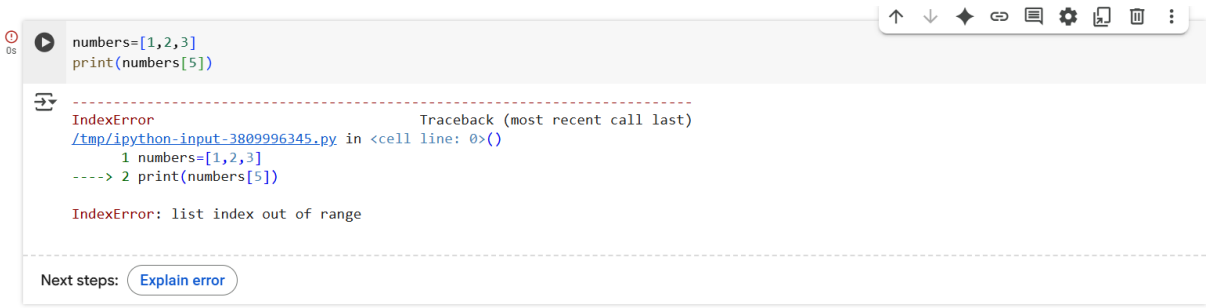
Explanation:

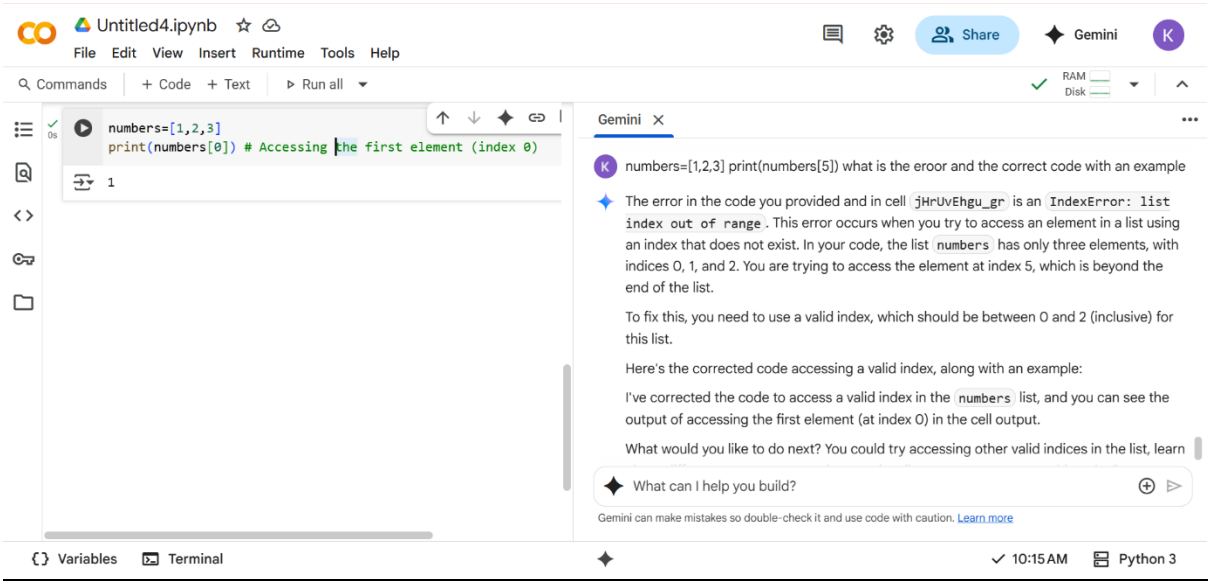
- **Cell** `q1ddCP8ZqyrS` : This cell defines a Python class called `Rectangle`. Classes are blueprints for creating objects. The `__init__` method is a special method called a constructor; it's automatically called when you create a new `Rectangle` object. The `self` parameter refers to the instance of the class being created. This `__init__` method takes `length` and `width` as arguments and stores them as attributes of the object using `self.length = length` and `self.width = width`.
- **Cell** `a498d427` : This cell demonstrates how to use the `Rectangle` class defined in the previous cell.
  - `my_rectangle = Rectangle(10, 5)` creates a new `Rectangle` object with a length of 10 and a width of 5. This calls the `__init__` method of the `Rectangle` class.
  - The `print` statements access the `length` and `width` attributes of the `my_rectangle` object using dot notation (`my_rectangle.length` and `my_rectangle.width`) and print their values.

Task 5:

```python
numbers = [1, 2, 3]
print(numbers[5])
```

Code and Output:





Explanation: