Assignment 13

# Task 01

* **Task:** Refactor repeated loops into a cleaner, more Pythonic approach.

**Instructions:**

* Analyze the legacy code.
* Identify the part that uses loops to compute values.
* Refactor using **list comprehensions** or helper functions while keeping the output the same.  
  **Legacy Code:**

numbers = [1, 2, 3, 4, 5]

squares = []

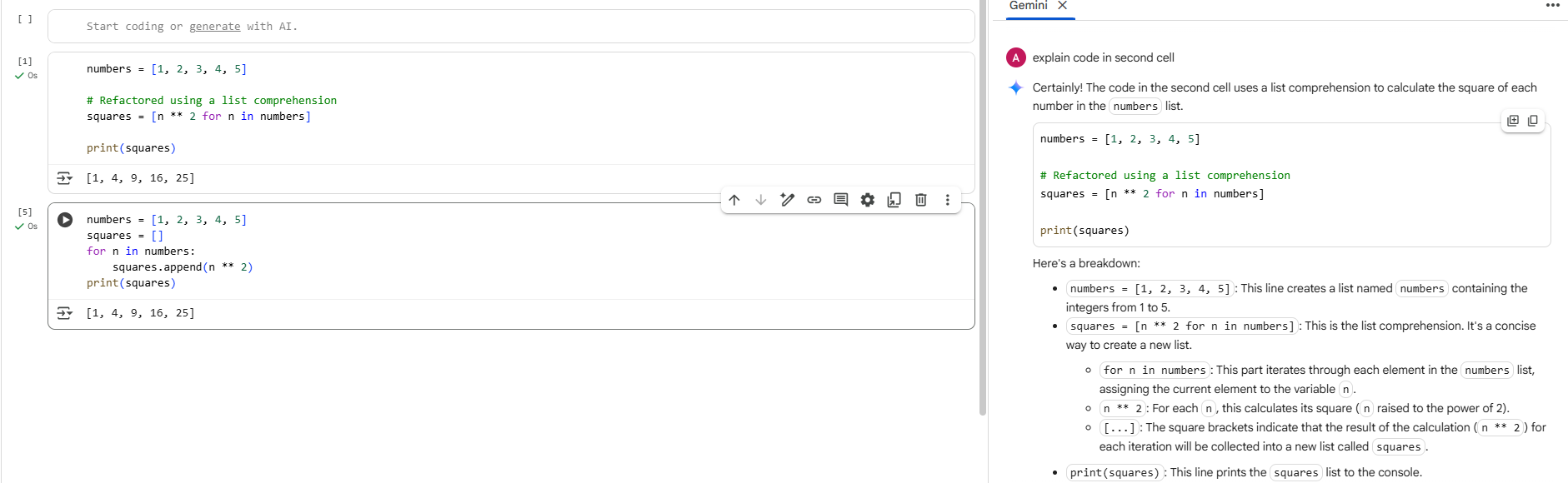
for n in numbers:

squares.append(n \*\* 2)

print(squares)

**Expected Output:**

[1, 4, 9, 16, 25]



# Task 02

**Task:** Simplify string concatenation.  
**Instructions:**

* Review the loop that builds a sentence using +=.
* Refactor using " ".join() to improve efficiency and readability.  
  **Legacy Code:**

words = ["AI", "helps", "in", "refactoring", "code"]

sentence = ""

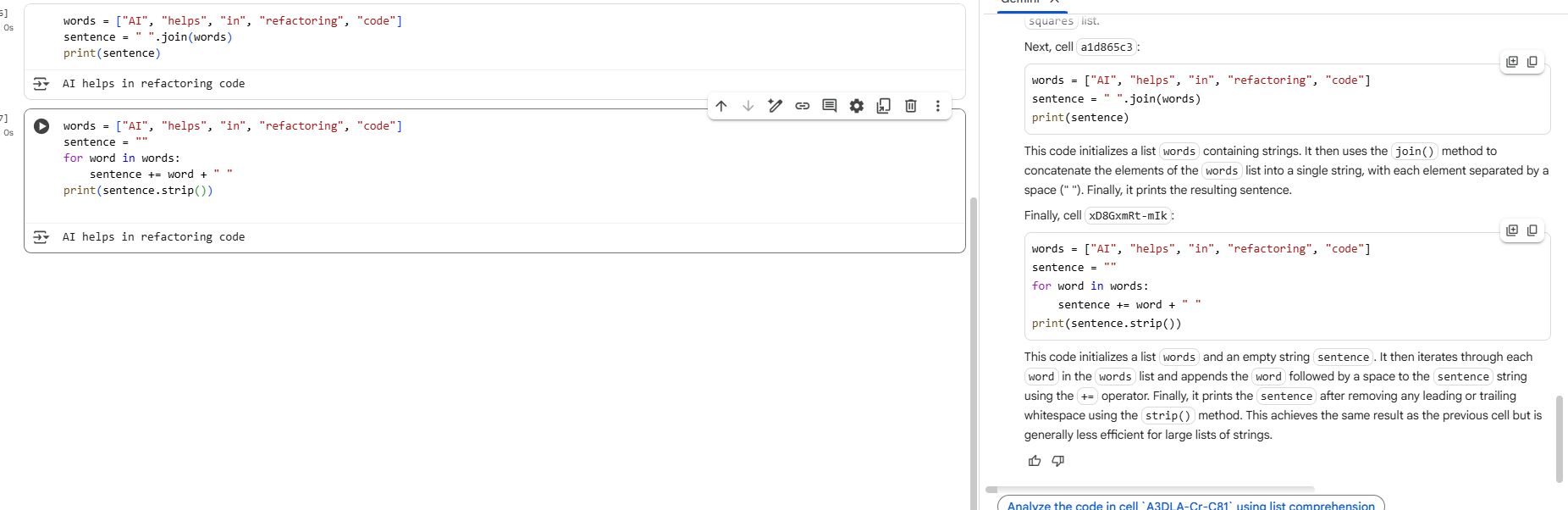
for word in words:

sentence += word + " "

print(sentence.strip())

**Expected Output:**

AI helps in refactoring code



# Task 03

**Task 3**

**Task:** Replace manual dictionary lookup with a safer method.  
**Instructions:**

* Check how the code accesses dictionary keys.
* Use .get() or another Pythonic approach to handle missing keys gracefully.  
  **Legacy Code:**

student\_scores = {"Alice": 85, "Bob": 90}

if "Charlie" in student\_scores:

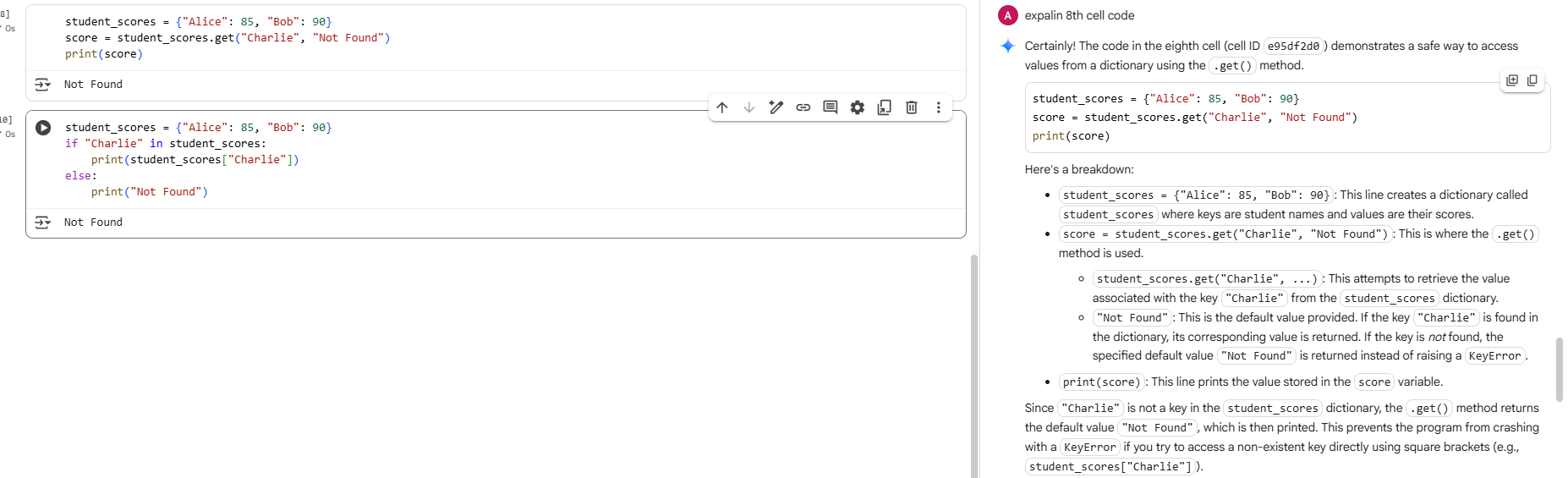
print(student\_scores["Charlie"])

else:

print("Not Found")

**Expected Output:**

Not Found



## Task 04

**Task:** Refactor repetitive if-else blocks.  
**Instructions:**

* Examine multiple if-elif statements for operations.
* Refactor using **dictionary mapping** to make the code scalable and clean.  
  **Legacy Code:**

operation = "multiply"

a, b = 5, 3

if operation == "add":

result = a + b

elif operation == "subtract":

result = a - b

elif operation == "multiply":

result = a \* b

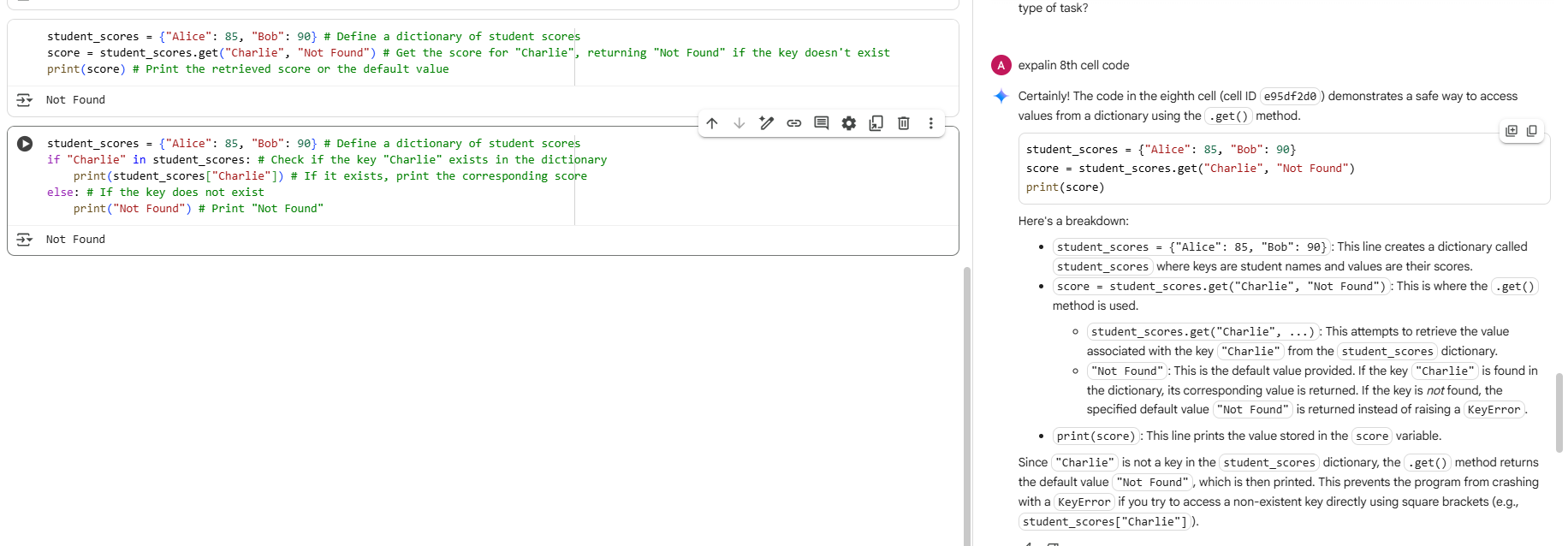
else:

result = None

print(result)

**Expected Output:**

15



## Task 05

**Task:** Optimize nested loops for searching.  
**Instructions:**

* Identify the nested loop used to find an element.
* Refactor using Python’s in keyword or other efficient search techniques.  
  **Legacy Code:**

items = [10, 20, 30, 40, 50]

found = False

for i in items:

if i == 30:

found = True

break

print("Found" if found else "Not Found")

**Expected Output:**

Found

