ASSIGNMENT-7.5

M.AKHIL REDDY

2303A52315

BATCH-45

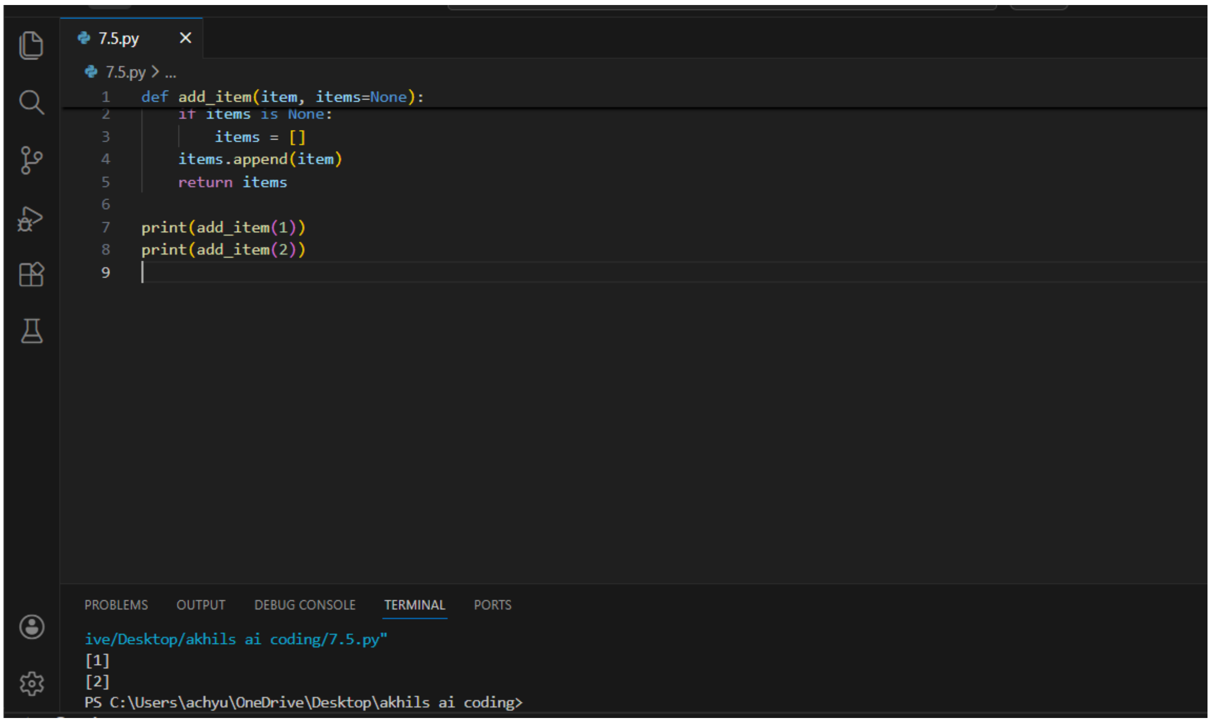
Task 1 (Mutable Default Argument – Function Bug)

Task: Analyze given code where a mutable default argument causes unexpected behavior. Use AI to fix it.

```
# Bug: Mutable default argument
def add_item(item, items=[]):
    items.append(item)
    return items
print(add_item(1))
print(add_item(2))
```

Expected Output: Corrected function avoids shared list bug.

- The list `items` is **shared across all function calls**
- That's why the second call keeps data from the first call



```python
def add_item(item, items=None):
    if items is None:
        items = []
    items.append(item)
    return items

print(add_item(1))
print(add_item(2))
```

```
ive/Desktop/akhils ai coding/7.5.py"
[1]
[2]
PS C:\Users\achyu\OneDrive\Desktop\akhils ai coding>
```

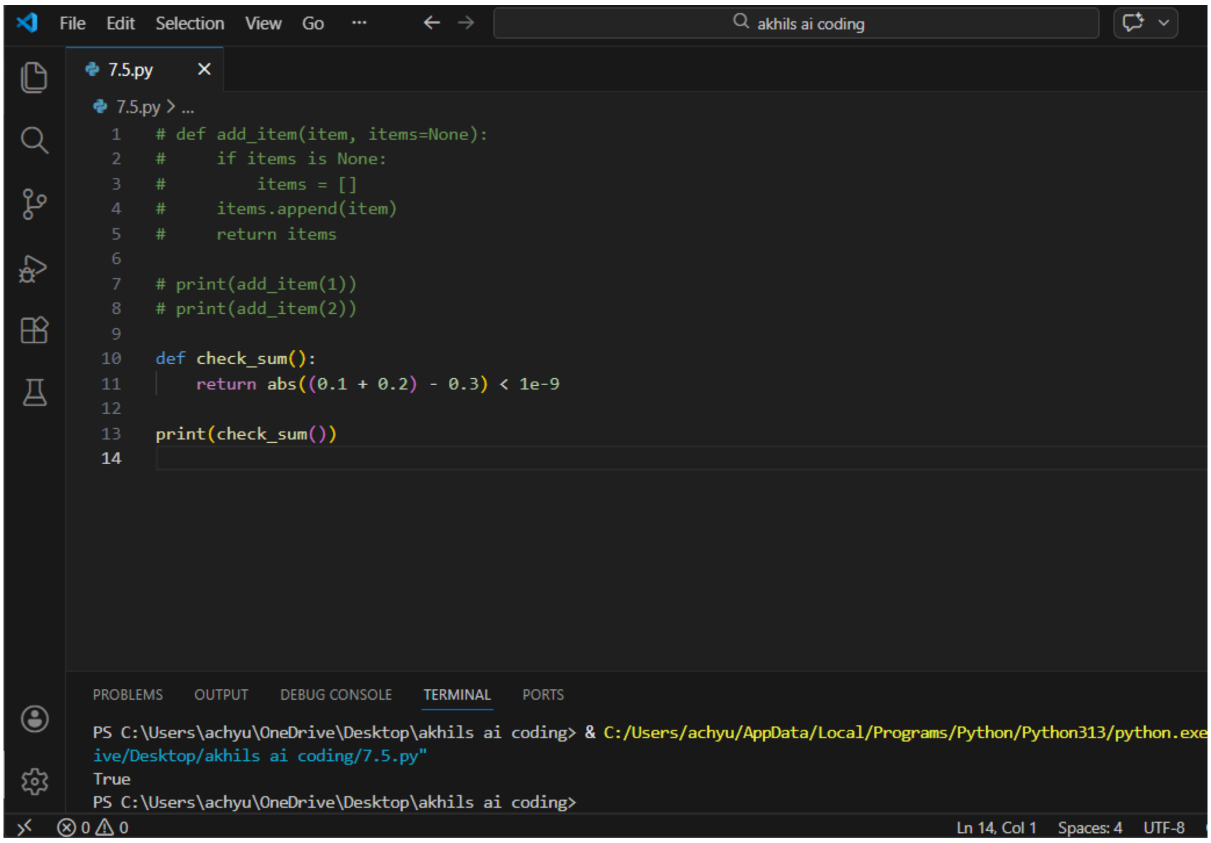
Task 2 (Floating-Point Precision Error)

Task2: Analyze given code where floating-point comparison fails.   Use AI to correct with tolerance.

```
# Bug: Floating point precision issue

def check_sum():

    return (0.1 + 0.2) == 0.3

print(check_sum())
```

Expected Output: Corrected function

Floating-point numbers are stored in **binary**, so some decimal values cannot be represented exactly.


File Edit Selection View Go ···
akhils ai coding
7.5.py
7.5.py > ...
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
Ln 14, Col 1 Spaces: 4 UTF-8

```python
# def add_item(item, items=None):
#     if items is None:
#         items = []
#     items.append(item)
#     return items

# print(add_item(1))
# print(add_item(2))

def check_sum():
    return abs((0.1 + 0.2) - 0.3) < 1e-9

print(check_sum())
```

```
PS C:\Users\achyu\OneDrive\Desktop\akhils ai coding> & C:/Users/achyu/AppData/Local/Programs/Python/Python313/python.exe
ive/Desktop/akhils ai coding/7.5.py"
True
PS C:\Users\achyu\OneDrive\Desktop\akhils ai coding>
```

Task 3 (Recursion Error – Missing Base Case)

Task: Analyze given code where recursion runs infinitely due to missing base case. Use AI to fix.

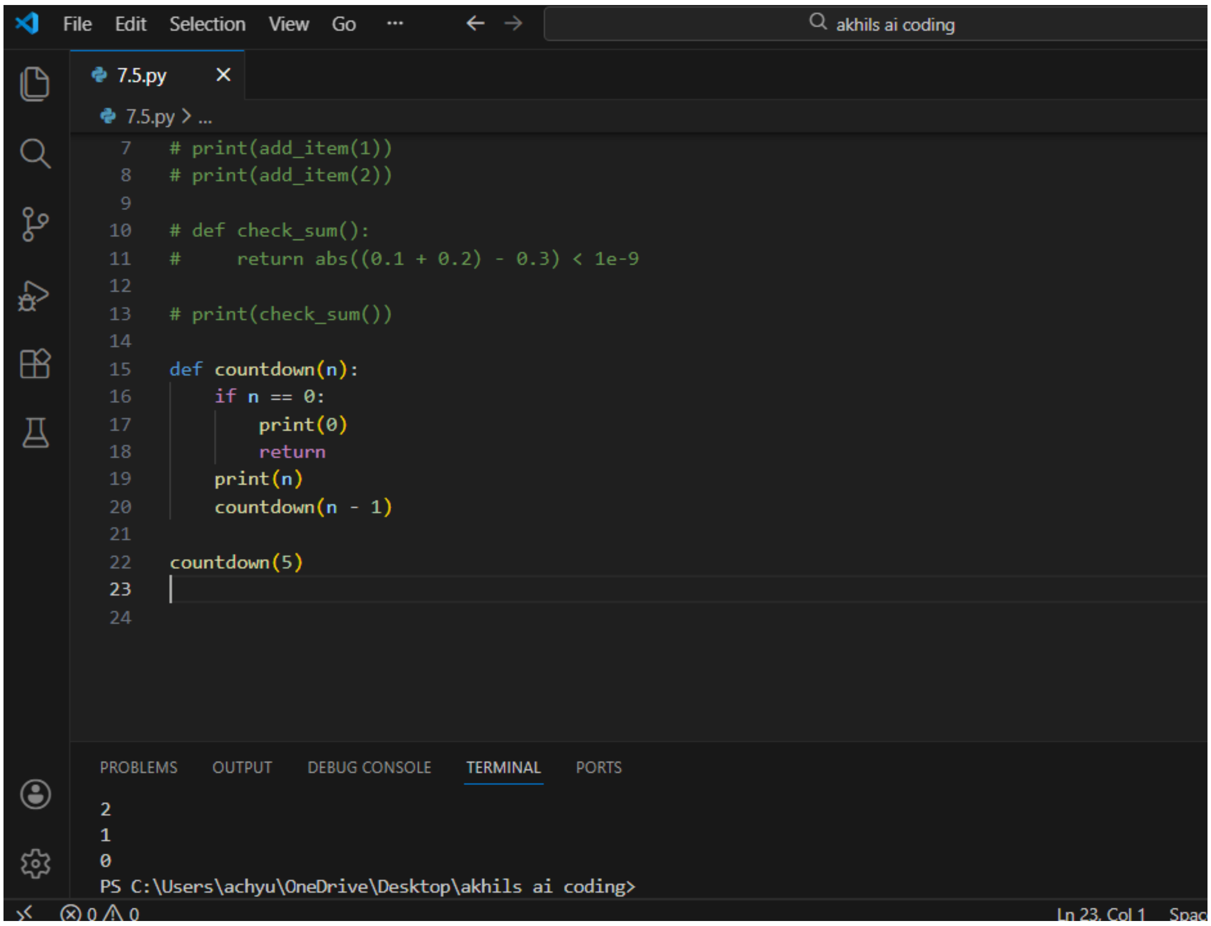
```
# Bug: No base case
def countdown(n):
    print(n)
    return countdown(n-1)
countdown(5)
```

Expected Output : Correct recursion with stopping condition.

There is **no base case**, so the function keeps calling itself forever.

This leads to:

- Infinite recursion
- `RecursionError: maximum recursion depth exceeded`



```python
# print(add_item(1))
# print(add_item(2))

# def check_sum():
#     return abs((0.1 + 0.2) - 0.3) < 1e-9

# print(check_sum())

def countdown(n):
    if n == 0:
        print(0)
        return
    print(n)
    countdown(n - 1)

countdown(5)

```

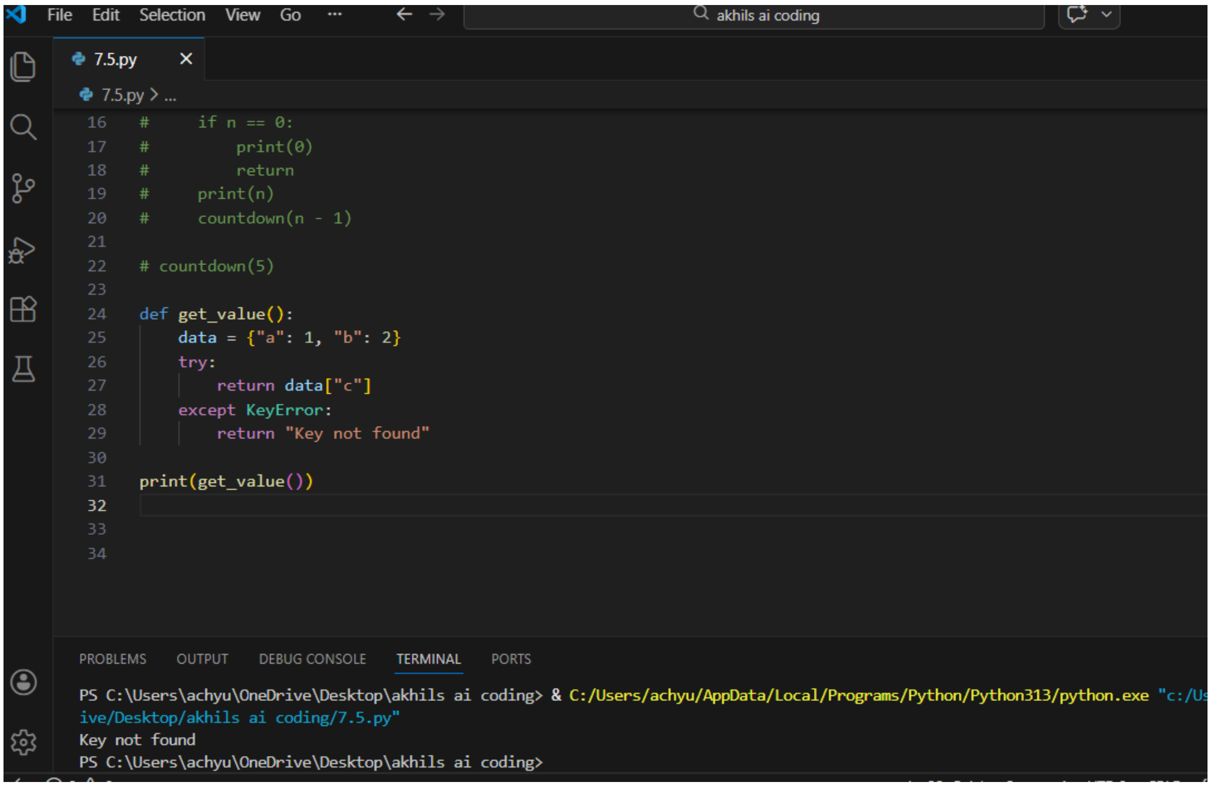
```
2
1
0
PS C:\Users\achyu\OneDrive\Desktop\akhils ai coding>
```

Task 4 (Dictionary Key Error)

Task: Analyze given code where a missing dictionary key causes error. Use AI to fix it.

```
# Bug: Accessing non-existing key
def get_value():
    data = {"a": 1, "b": 2}
    return data["c"]
print(get_value())
```

Expected Output: Corrected with .get() or error handling.


File Edit Selection View Go ...
akhils ai coding
7.5.py
7.5.py > ...
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```python
#     if n == 0:
#         print(0)
#         return
#     print(n)
#     countdown(n - 1)

# countdown(5)

def get_value():
    data = {"a": 1, "b": 2}
    try:
        return data["c"]
    except KeyError:
        return "Key not found"

print(get_value())
```

```
PS C:\Users\achyu\OneDrive\Desktop\akhils ai coding> & C:/Users/achyu/AppData/Local/Programs/Python/Python313/python.exe "c:/Us
ive/Desktop/akhils ai coding/7.5.py"
Key not found
PS C:\Users\achyu\OneDrive\Desktop\akhils ai coding>
```

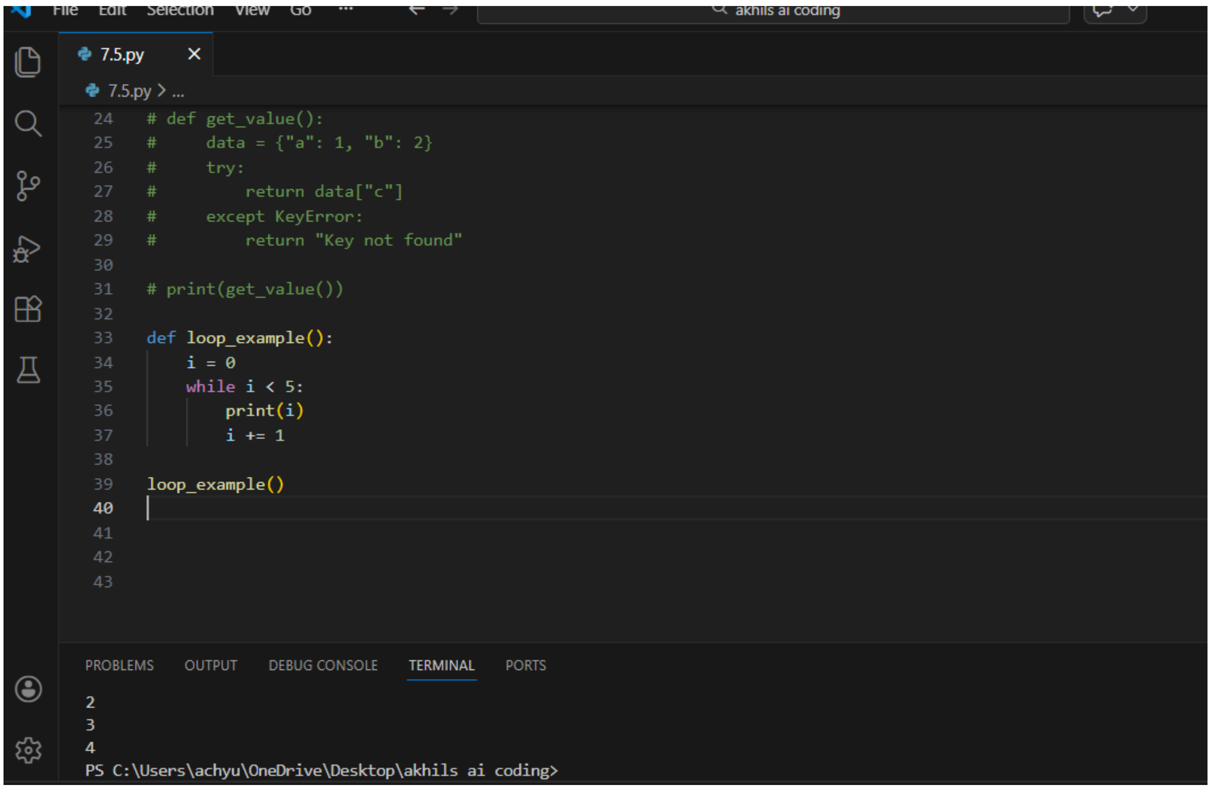
given very well

Task 5 (Infinite Loop – Wrong Condition)

Task: Analyze given code where loop never ends. Use AI to detect and fix it.

```
# Bug: Infinite loop
def loop_example():
    i = 0
    while i < 5:
        print(i)
```

Expected Output: Corrected loop increments i.


File Edit Selection View Go ...
akhils ai coding
7.5.py
7.5.py > ...
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```python
# def get_value():
#     data = {"a": 1, "b": 2}
#     try:
#         return data["c"]
#     except KeyError:
#         return "Key not found"

# print(get_value())

def loop_example():
    i = 0
    while i < 5:
        print(i)
        i += 1

loop_example()
```

```
2
3
4
PS C:\Users\achyu\OneDrive\Desktop\akhils ai coding>
```

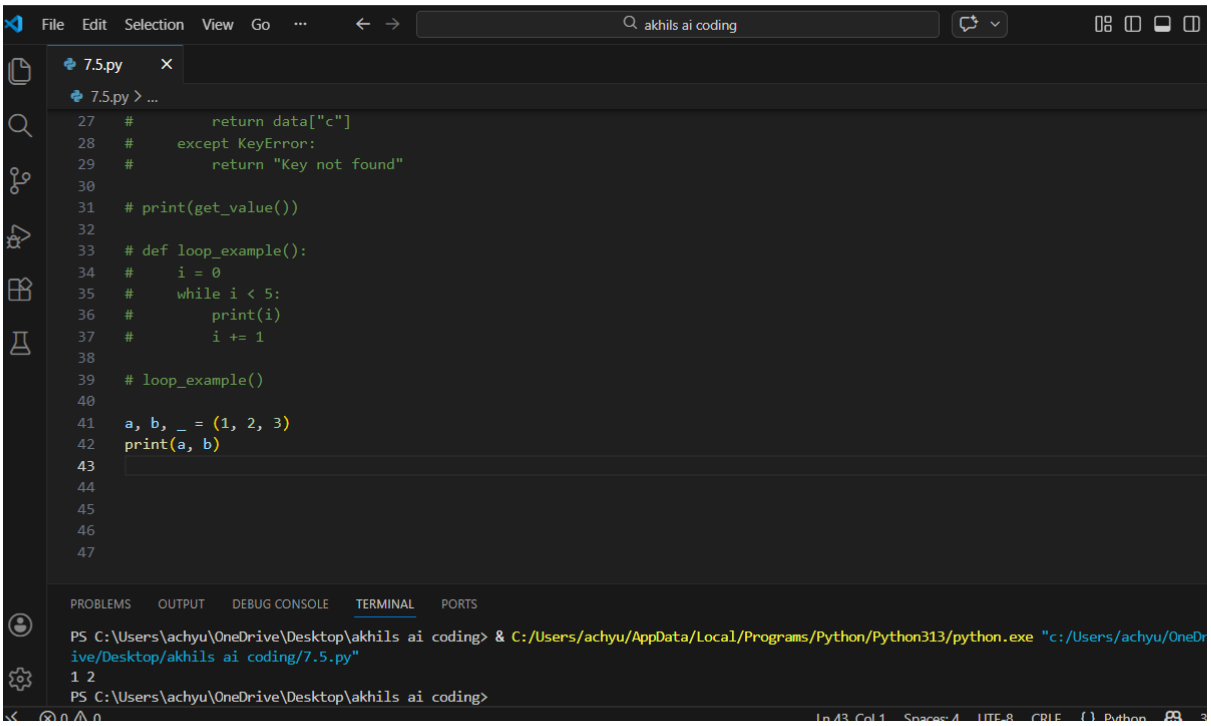
Task 6 (Unpacking Error – Wrong Variables)

Task: Analyze given code where tuple unpacking fails. Use AI to fix it.

# Bug: Wrong unpacking

a, b = (1, 2, 3)

Expected Output: Correct unpacking or using _ for extra values.


File Edit Selection View Go ···
akhils ai coding
7.5.py
7.5.py > ...
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
Ln 43, Col 1 Spaces: 4 UTF-8 CRLF {} Python

```
#         return data["c"]
#     except KeyError:
#         return "Key not found"

# print(get_value())

# def loop_example():
#     i = 0
#     while i < 5:
#         print(i)
#         i += 1

# loop_example()

a, b, _ = (1, 2, 3)
print(a, b)
```

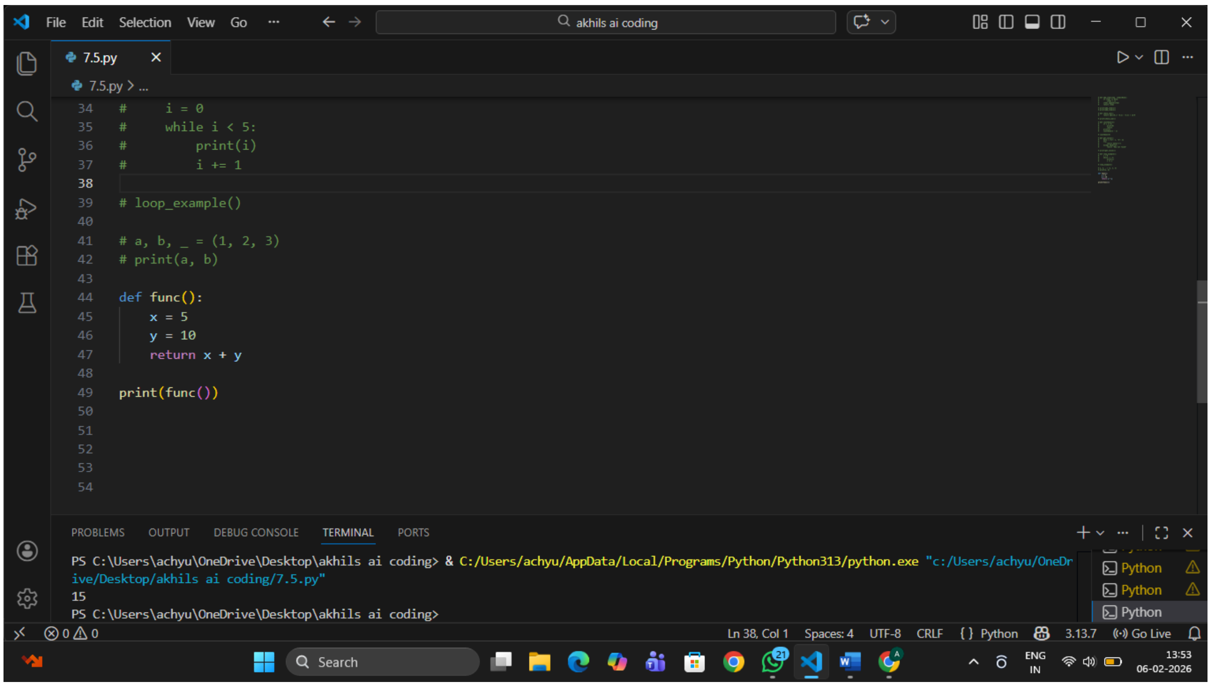
```
PS C:\Users\achyu\OneDrive\Desktop\akhils ai coding> & C:/Users/achyu/AppData/Local/Programs/Python/Python313/python.exe "c:/Users/achyu/OneDr
ive/Desktop/akhils ai coding/7.5.py"
1 2
PS C:\Users\achyu\OneDrive\Desktop\akhils ai coding>
```

Task 7 (Mixed Indentation – Tabs vs Spaces)

Task: Analyze given code where mixed indentation breaks execution. Use AI to fix it.

```
# Bug: Mixed indentation
def func():
    x = 5
        y = 10
    return x+y
```

Expected Output : Consistent indentation applied.



```
#     i = 0
#     while i < 5:
#         print(i)
#         i += 1

# loop_example()

# a, b, _ = (1, 2, 3)
# print(a, b)

def func():
    x = 5
    y = 10
    return x + y

print(func())
```

```
PS C:\Users\achyu\OneDrive\Desktop\akhils ai coding> & C:/Users/achyu/AppData/Local/Programs/Python/Python313/python.exe "c:/Users/achyu/OneDrive/Desktop/akhils ai coding/7.5.py"
15
PS C:\Users\achyu\OneDrive\Desktop\akhils ai coding>
```

Task 8 (Import Error – Wrong Module Usage)

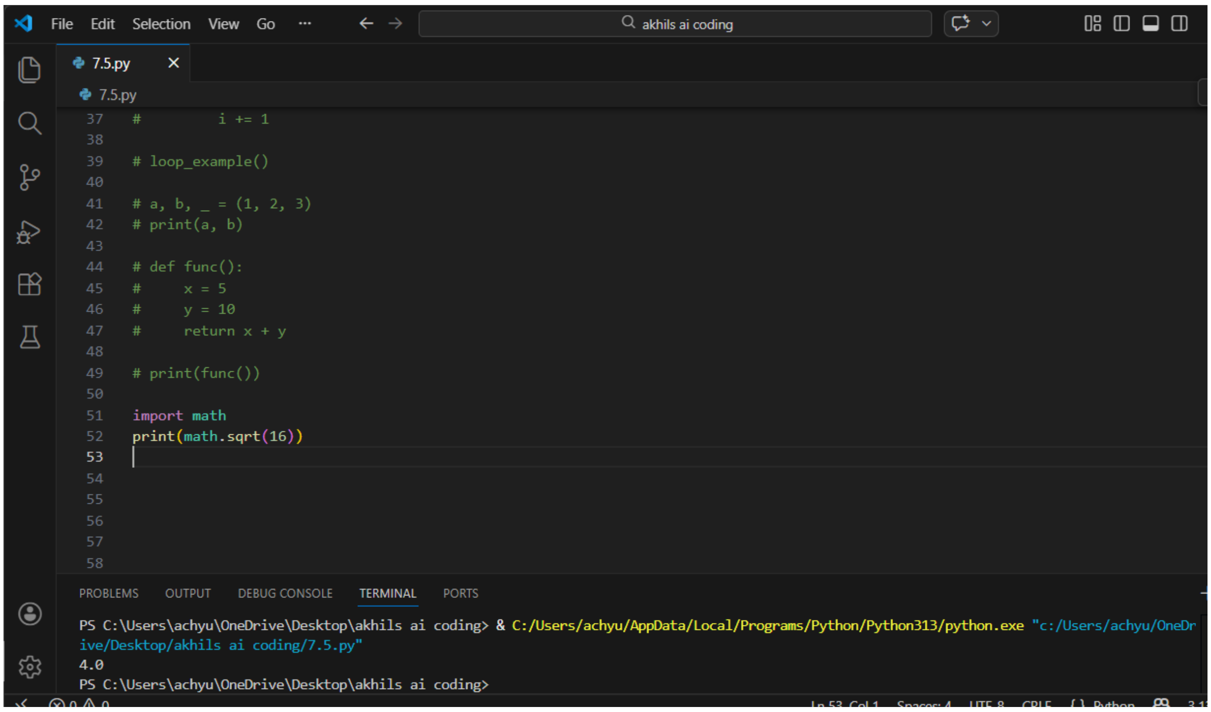
Task: Analyze given code with incorrect import. Use AI to fix.

# Bug: Wrong import

import maths

print(maths.sqrt(16))

Expected Output: Corrected to import math



```
#         i += 1

# loop_example()

# a, b, _ = (1, 2, 3)
# print(a, b)

# def func():
#     x = 5
#     y = 10
#     return x + y

# print(func())

import math
print(math.sqrt(16))
```

```
PS C:\Users\achyu\OneDrive\Desktop\akhils ai coding> & C:/Users/achyu/AppData/Local/Programs/Python/Python313/python.exe "c:/Users/achyu/OneDrive/Desktop/akhils ai coding/7.5.py"
4.0
PS C:\Users\achyu\OneDrive\Desktop\akhils ai coding>
```

Task 9 (Unreachable Code – Return Inside Loop)

Task: Analyze given code where a return inside a loop prevents full iteration. Use AI to fix it.

# Bug: Early return inside loop

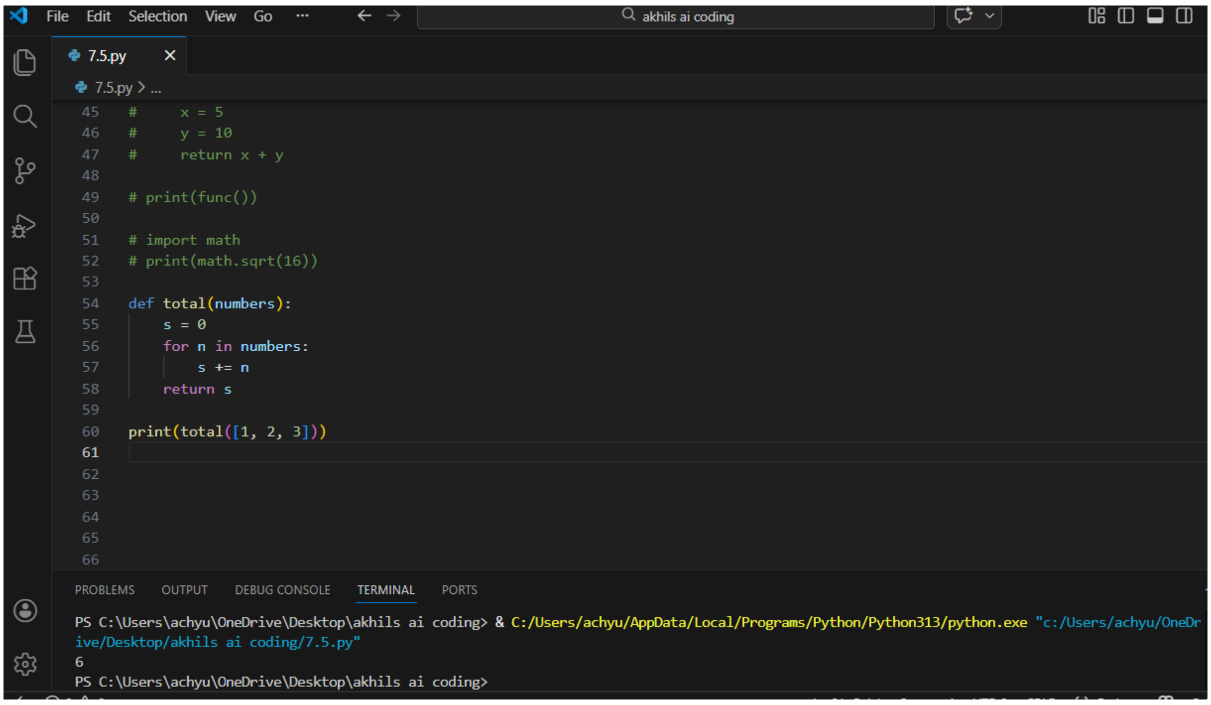
def total(numbers):

    for n in numbers:

        return n

print(total([1,2,3]))

Expected Output: Corrected code accumulates sum and returns after loop.



```python
#     x = 5
#     y = 10
#     return x + y

# print(func())

# import math
# print(math.sqrt(16))

def total(numbers):
    s = 0
    for n in numbers:
        s += n
    return s

print(total([1, 2, 3]))
```

```
PS C:\Users\achyu\OneDrive\Desktop\akhils ai coding> & C:/Users/achyu/AppData/Local/Programs/Python/Python313/python.exe "c:/Users/achyu/OneDrive/Desktop/akhils ai coding/7.5.py"
6
PS C:\Users\achyu\OneDrive\Desktop\akhils ai coding>
```

Task 10 (Name Error – Undefined Variable)

Task: Analyze given code where a variable is used before being defined. Let AI detect and fix the error.

# Bug: Using undefined variable

def calculate_area():

    return length * width

print(calculate_area())

Requirements:

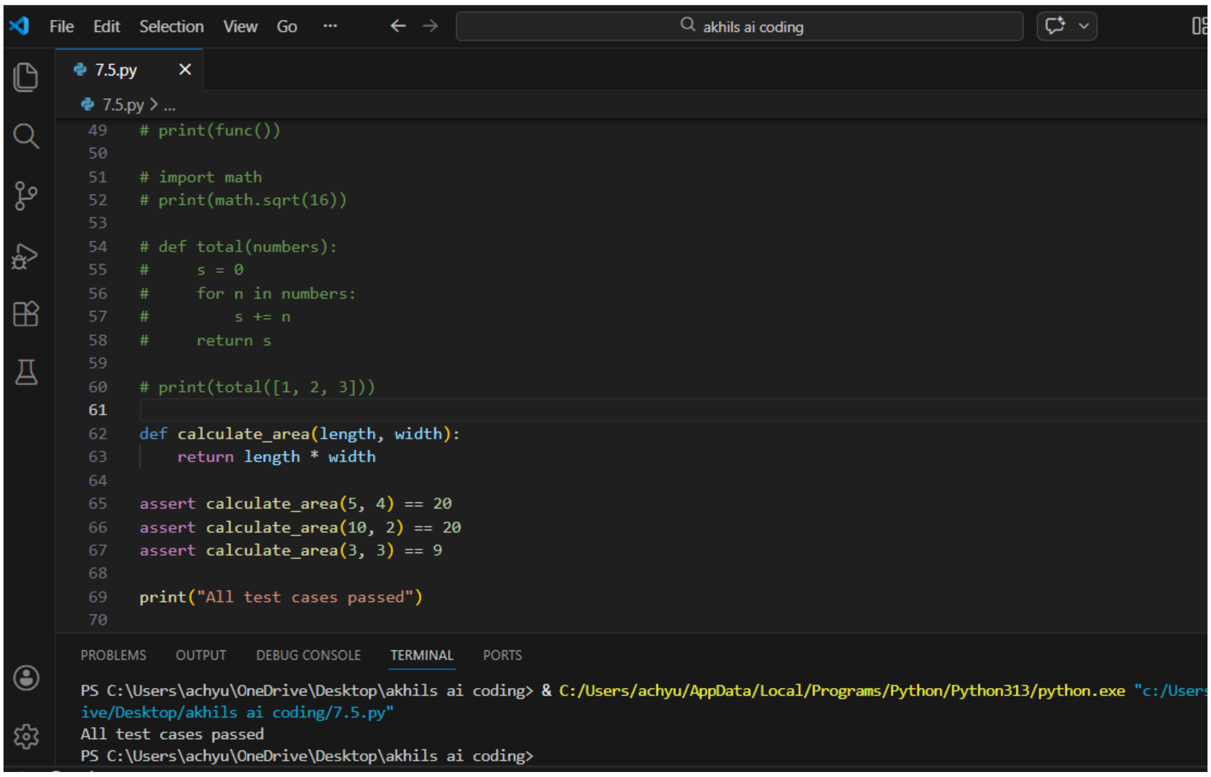
- Run the code to observe the error.

- Ask AI to identify the missing variable definition.
- Fix the bug by defining length and width as parameters.
- Add 3 assert test cases for correctness.

Expected Output :

- Corrected code with parameters.
- AI explanation of the bug.

Successful execution of assertions.



File Edit Selection View Go ...
akhils ai coding
7.5.py
7.5.py > ...
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```python
# print(func())

# import math
# print(math.sqrt(16))

# def total(numbers):
#     s = 0
#     for n in numbers:
#         s += n
#     return s

# print(total([1, 2, 3]))

def calculate_area(length, width):
    return length * width

assert calculate_area(5, 4) == 20
assert calculate_area(10, 2) == 20
assert calculate_area(3, 3) == 9

print("All test cases passed")
```

```
PS C:\Users\achyu\OneDrive\Desktop\akhils ai coding> & C:/Users/achyu/AppData/Local/Programs/Python/Python313/python.exe "c:/Users
ive/Desktop/akhils ai coding/7.5.py"
All test cases passed
PS C:\Users\achyu\OneDrive\Desktop\akhils ai coding>
```

Task 11 (Type Error – Mixing Data Types Incorrectly)

Task: Analyze given code where integers and strings are added incorrectly. Let AI detect and fix the error.

```
# Bug: Adding integer and string

def add_values():

  return 5 + "10"

print(add_values())
```

Requirements:

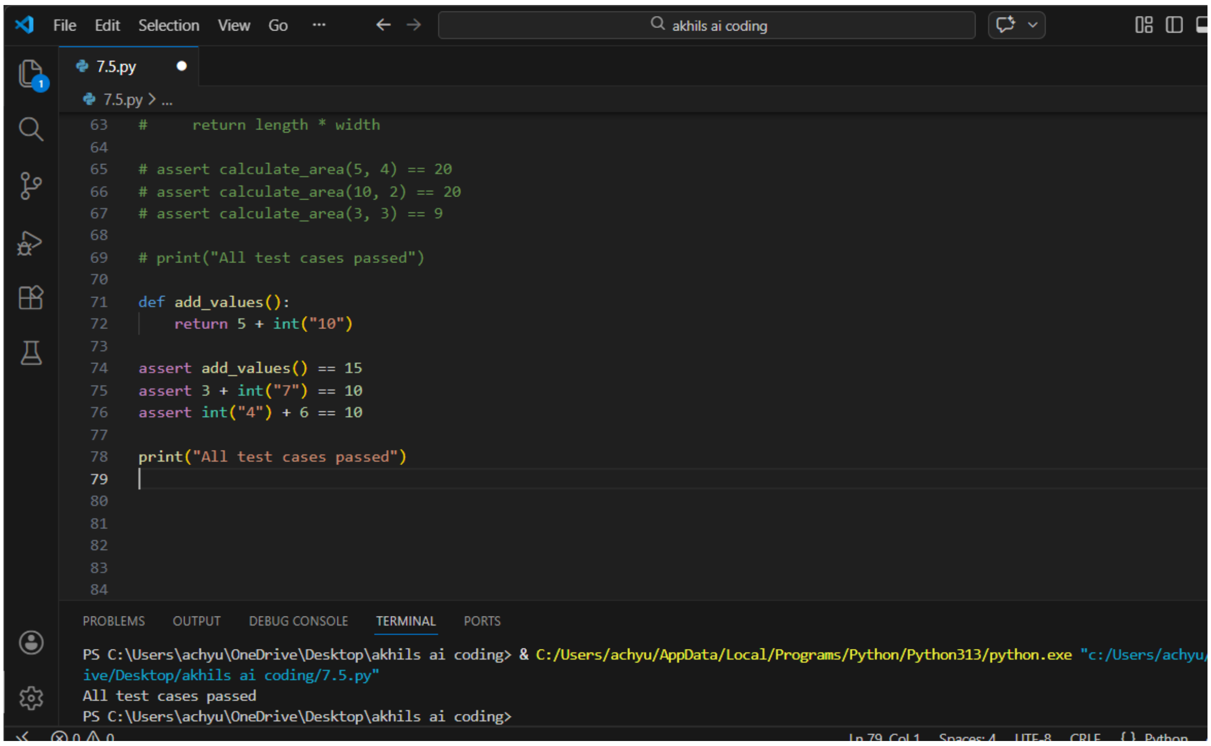
- Run the code to observe the error.
- AI should explain why int + str is invalid.

- Fix the code by type conversion (e.g., int("10") or str(5)).
- Verify with 3 assert cases.

Expected Output #6:

- Corrected code with type handling.
- AI explanation of the fix.

Successful test validation.



```
#     return length * width

# assert calculate_area(5, 4) == 20
# assert calculate_area(10, 2) == 20
# assert calculate_area(3, 3) == 9

# print("All test cases passed")

def add_values():
    return 5 + int("10")

assert add_values() == 15
assert 3 + int("7") == 10
assert int("4") + 6 == 10

print("All test cases passed")
```

```
PS C:\Users\achyu\OneDrive\Desktop\akhils ai coding> & C:/Users/achyu/AppData/Local/Programs/Python/Python313/python.exe "c:/Users/achyu/
ive/Desktop/akhils ai coding/7.5.py"
All test cases passed
PS C:\Users\achyu\OneDrive\Desktop\akhils ai coding>
```

Task 12 (Type Error – String + List Concatenation)

Task: Analyze code where a string is incorrectly added to a list.

```
# Bug: Adding string and list
def combine():
    return "Numbers: " + [1, 2, 3]
print(combine())
```

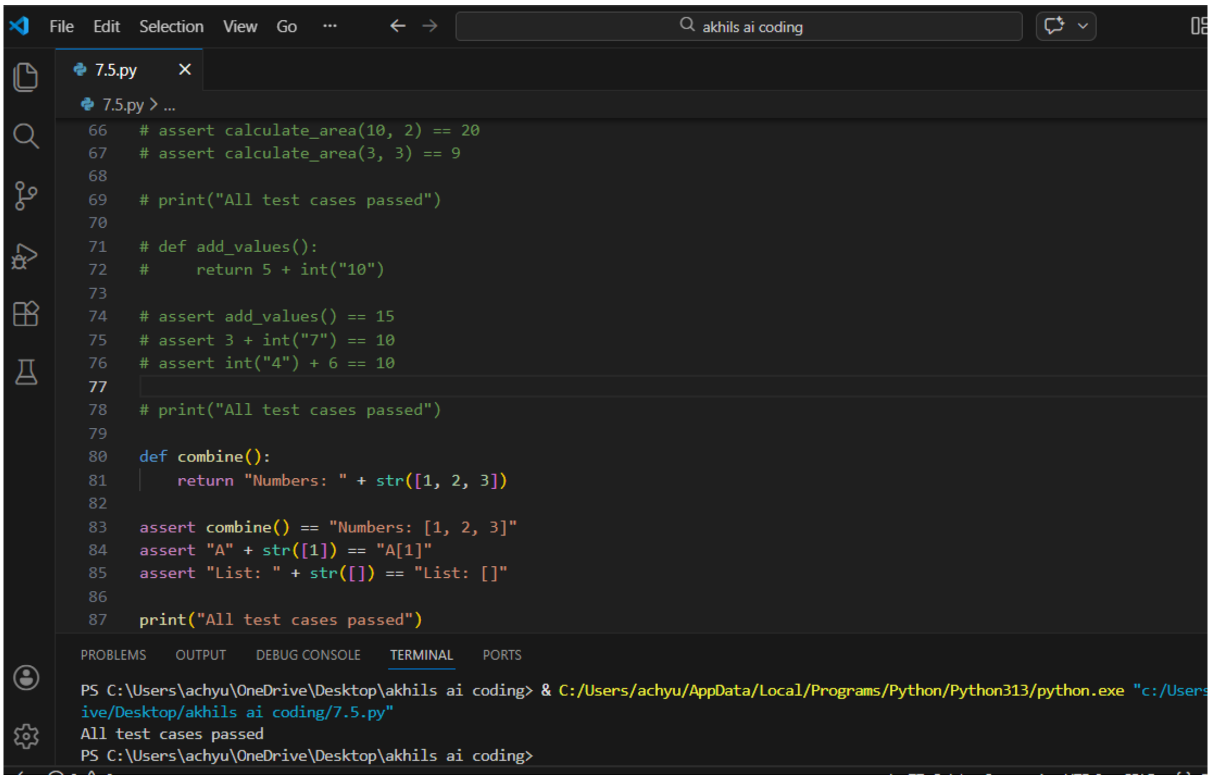
Requirements:

- Run the code to observe the error.

- Explain why str + list is invalid.
- Fix using conversion (str([1,2,3]) or " ".join()).
- Verify with 3 assert cases.

Expected Output:

- Corrected code
- Explanation
- Successful test validation



File Edit Selection View Go ···
akhils ai coding
7.5.py
7.5.py > ...
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
# assert calculate_area(10, 2) == 20
# assert calculate_area(3, 3) == 9

# print("All test cases passed")

# def add_values():
#     return 5 + int("10")

# assert add_values() == 15
# assert 3 + int("7") == 10
# assert int("4") + 6 == 10

# print("All test cases passed")

def combine():
    return "Numbers: " + str([1, 2, 3])

assert combine() == "Numbers: [1, 2, 3]"
assert "A" + str([1]) == "A[1]"
assert "List: " + str([]) == "List: []"

print("All test cases passed")
```

```
PS C:\Users\achyu\OneDrive\Desktop\akhils ai coding> & C:/Users/achyu/AppData/Local/Programs/Python/Python313/python.exe "c:/Users
ive/Desktop/akhils ai coding/7.5.py"
All test cases passed
PS C:\Users\achyu\OneDrive\Desktop\akhils ai coding>
```

Task 13 (Type Error – Multiplying String by Float)

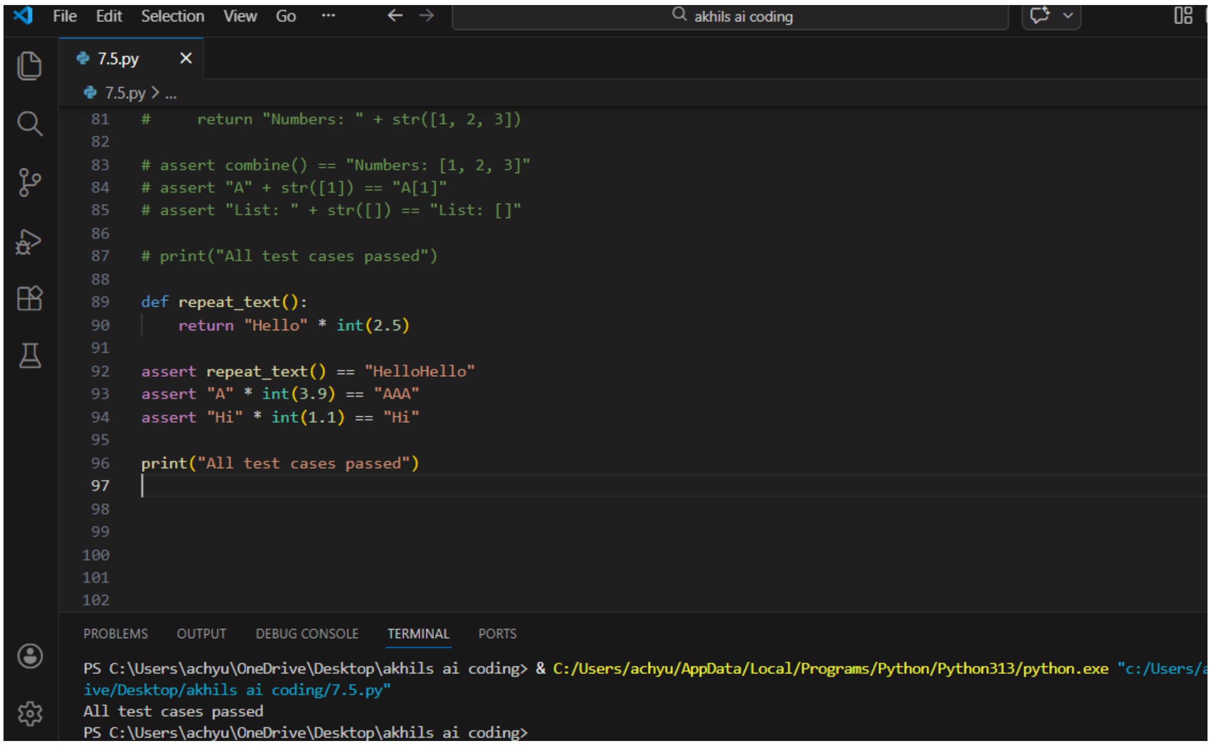
Task: Detect and fix code where a string is multiplied by a float.

```
# Bug: Multiplying string by float
def repeat_text():
    return "Hello" * 2.5
print(repeat_text())
```

Requirements:

- Observe the error.
- Explain why float multiplication is invalid for strings.

- Fix by converting float to int.
- Add 3 assert test cases.



```
#     return "Numbers: " + str([1, 2, 3])

# assert combine() == "Numbers: [1, 2, 3]"
# assert "A" + str([1]) == "A[1]"
# assert "List: " + str([]) == "List: []"

# print("All test cases passed")

def repeat_text():
    return "Hello" * int(2.5)

assert repeat_text() == "HelloHello"
assert "A" * int(3.9) == "AAA"
assert "Hi" * int(1.1) == "Hi"

print("All test cases passed")
```

```
PS C:\Users\achyu\OneDrive\Desktop\akhils ai coding> & C:/Users/achyu/AppData/Local/Programs/Python/Python313/python.exe "c:/Users/a
ive/Desktop/akhils ai coding/7.5.py"
All test cases passed
PS C:\Users\achyu\OneDrive\Desktop\akhils ai coding>
```

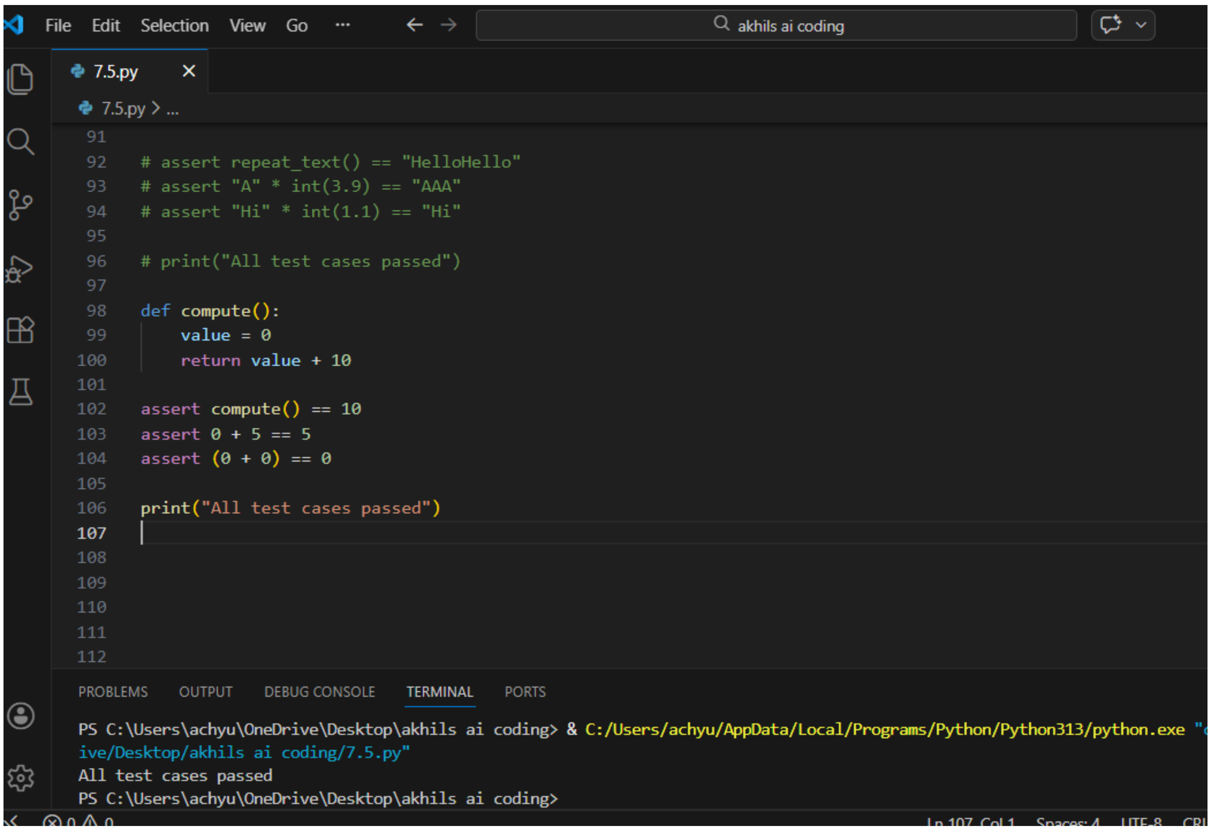
Task 14 (Type Error – Adding None to Integer)

Task: Analyze code where None is added to an integer.

```
# Bug: Adding None and integer
def compute():
    value = None
    return value + 10

print(compute())
```

Requirements:

- Run and identify the error.
- Explain why NoneType cannot be added.
- Fix by assigning a default value.
- Validate using asserts.


File Edit Selection View Go ...
akhils ai coding
7.5.py
7.5.py > ...
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
Ln 107, Col 1 Spaces: 4 UTF-8 CRL

```
# assert repeat_text() == "HelloHello"
# assert "A" * int(3.9) == "AAA"
# assert "Hi" * int(1.1) == "Hi"

# print("All test cases passed")

def compute():
    value = 0
    return value + 10

assert compute() == 10
assert 0 + 5 == 5
assert (0 + 0) == 0

print("All test cases passed")
```

```
PS C:\Users\achyu\OneDrive\Desktop\akhils ai coding> & C:/Users/achyu/AppData/Local/Programs/Python/Python313/python.exe "
ive/Desktop/akhils ai coding/7.5.py"
All test cases passed
PS C:\Users\achyu\OneDrive\Desktop\akhils ai coding>
```

____

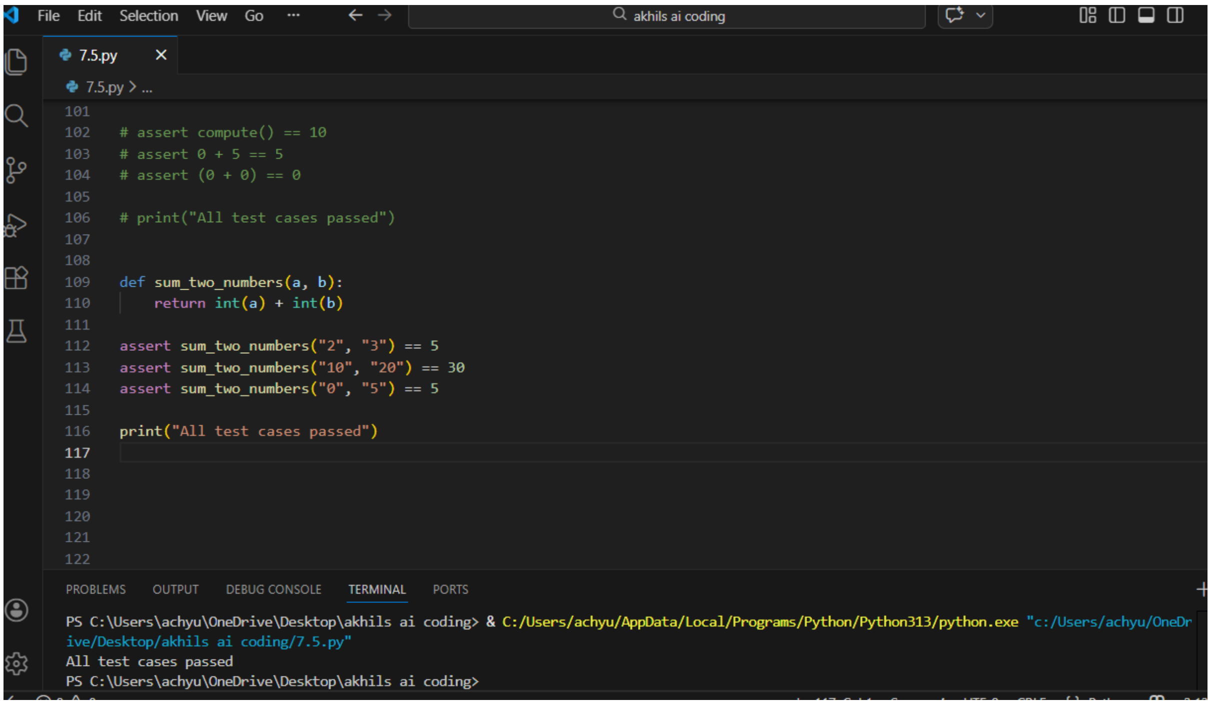
Task 15 (Type Error – Input Treated as String Instead of Number)

Task: Fix code where user input is not converted properly.

```
# Bug: Input remains string
def sum_two_numbers():
    a = input("Enter first number: ")
    b = input("Enter second number: ")
    return a + b


print(sum_two_numbers())
```

Requirements:

- Explain why input is always string.
- Fix using int() conversion.
- Verify with assert test cases.


File Edit Selection View Go ···
akhils ai coding
7.5.py
7.5.py > ...
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```python

# assert compute() == 10
# assert 0 + 5 == 5
# assert (0 + 0) == 0

# print("All test cases passed")


def sum_two_numbers(a, b):
    return int(a) + int(b)

assert sum_two_numbers("2", "3") == 5
assert sum_two_numbers("10", "20") == 30
assert sum_two_numbers("0", "5") == 5

print("All test cases passed")

```

```
PS C:\Users\achyu\OneDrive\Desktop\akhils ai coding> & C:/Users/achyu/AppData/Local/Programs/Python/Python313/python.exe "c:/Users/achyu/OneDr
ive/Desktop/akhils ai coding/7.5.py"
All test cases passed
PS C:\Users\achyu\OneDrive\Desktop\akhils ai coding>
```