AI ASSISTED CODING

LAB-7.5

Jadala Varshini

2303A51758

Batch-11

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=[]):

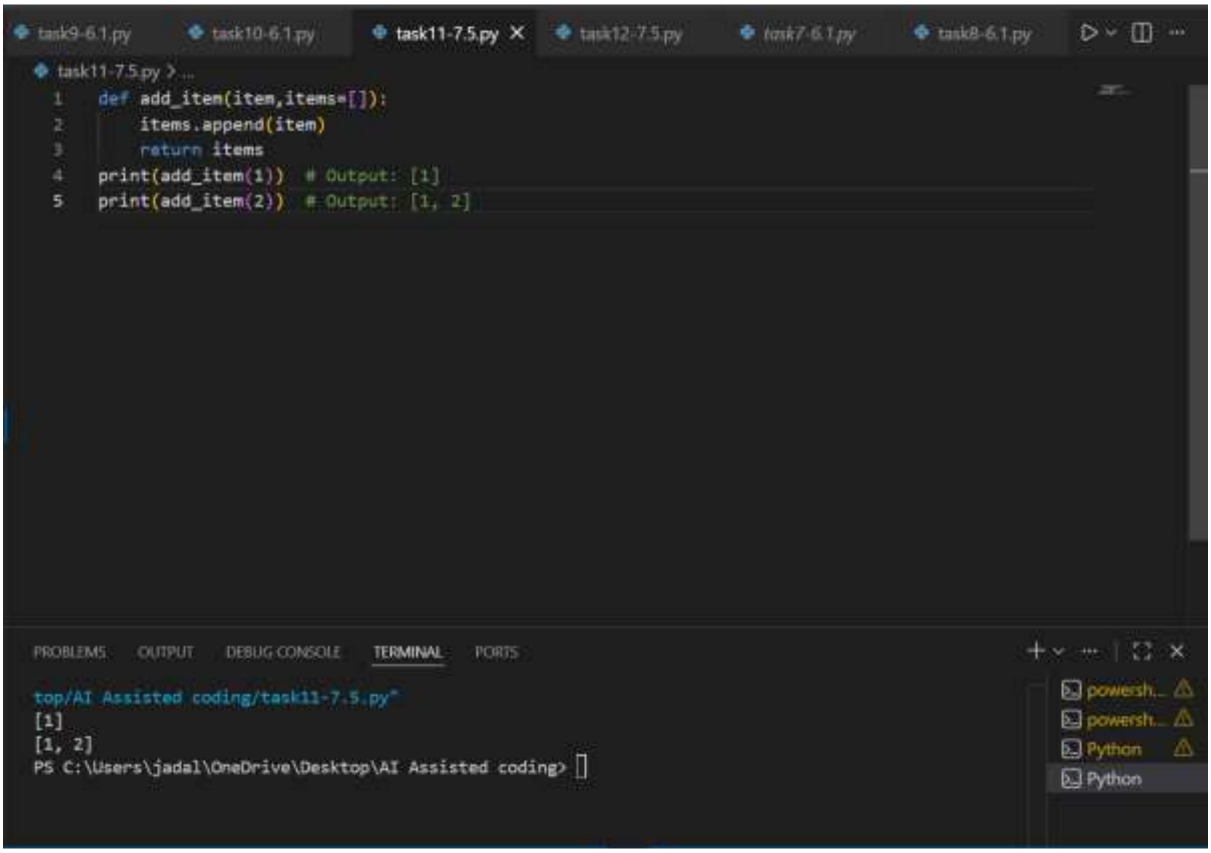
items.append(item) return

items print(add_item(1))

print(add_item(2))

Expected Output: Corrected function avoids shared list bug.

Given Code and Output:



```python
def add_item(item,items=[]):
    items.append(item)
    return items
print(add_item(1))  # Output: [1]
print(add_item(2))  # Output: [1, 2]
```

```
top/AI Assisted coding/task11-7.5.py"
[1]
[1, 2]
PS C:\Users\jadal\OneDrive\Desktop\AI Assisted coding> 
```

Task 2 (Floating-Point Precision Error)

Task: Analyze given code where floating-point comparison fails.

Use AI to correct with tolerance. #

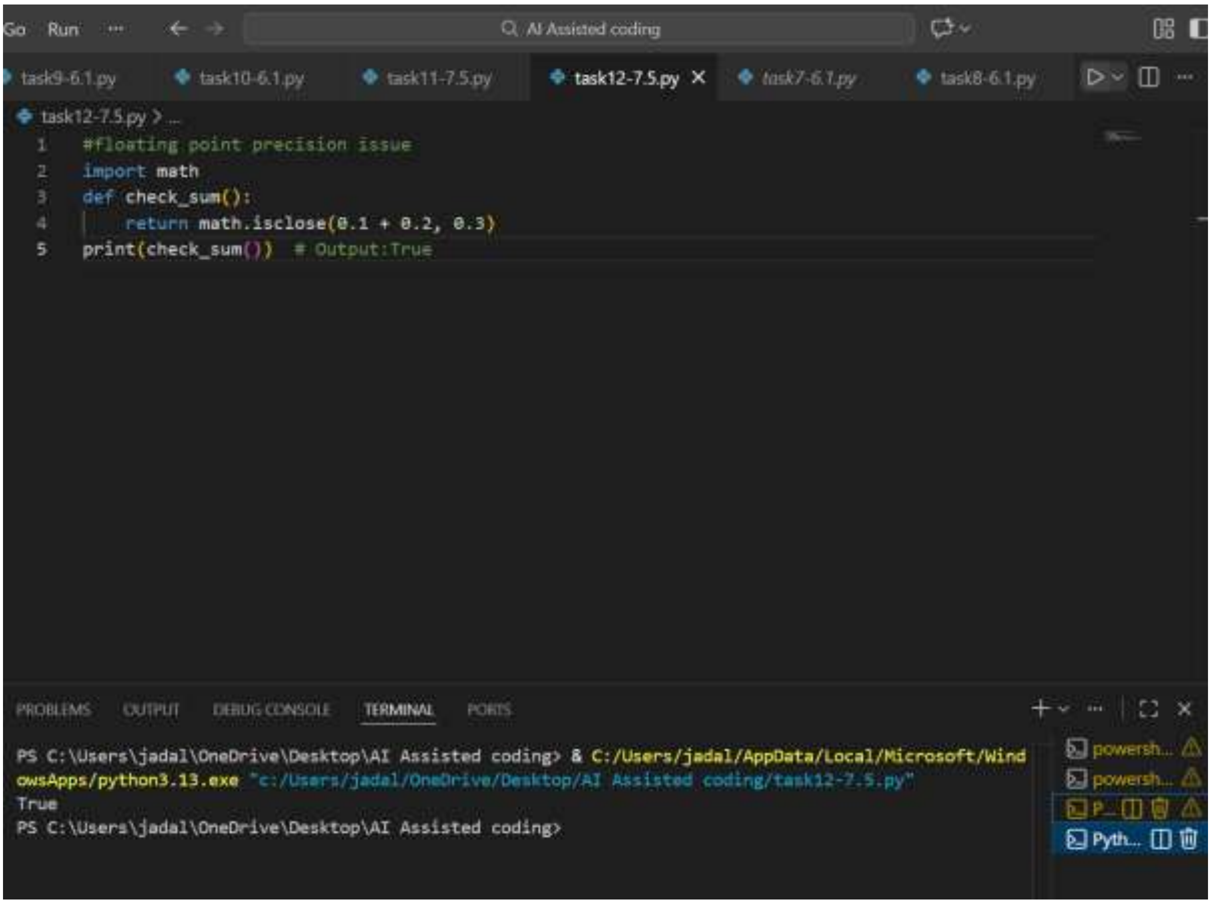
Bug: Floating point precision issue

def check_sum(): return (0.1 + 0.2)

== 0.3 print(check_sum())

Expected Output: Corrected function Given

Code and Output:



```python
#floating point precision issue
import math
def check_sum():
    return math.isclose(0.1 + 0.2, 0.3)
print(check_sum())  # Output:True
```

```
PS C:\Users\jadal\OneDrive\Desktop\AI Assisted coding> & C:/Users/jadal/AppData/Local/Microsoft/Wind
owsApps/python3.13.exe "c:/Users/jadal/OneDrive/Desktop/AI Assisted coding/task12-7.5.py"
True
PS C:\Users\jadal\OneDrive\Desktop\AI Assisted 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

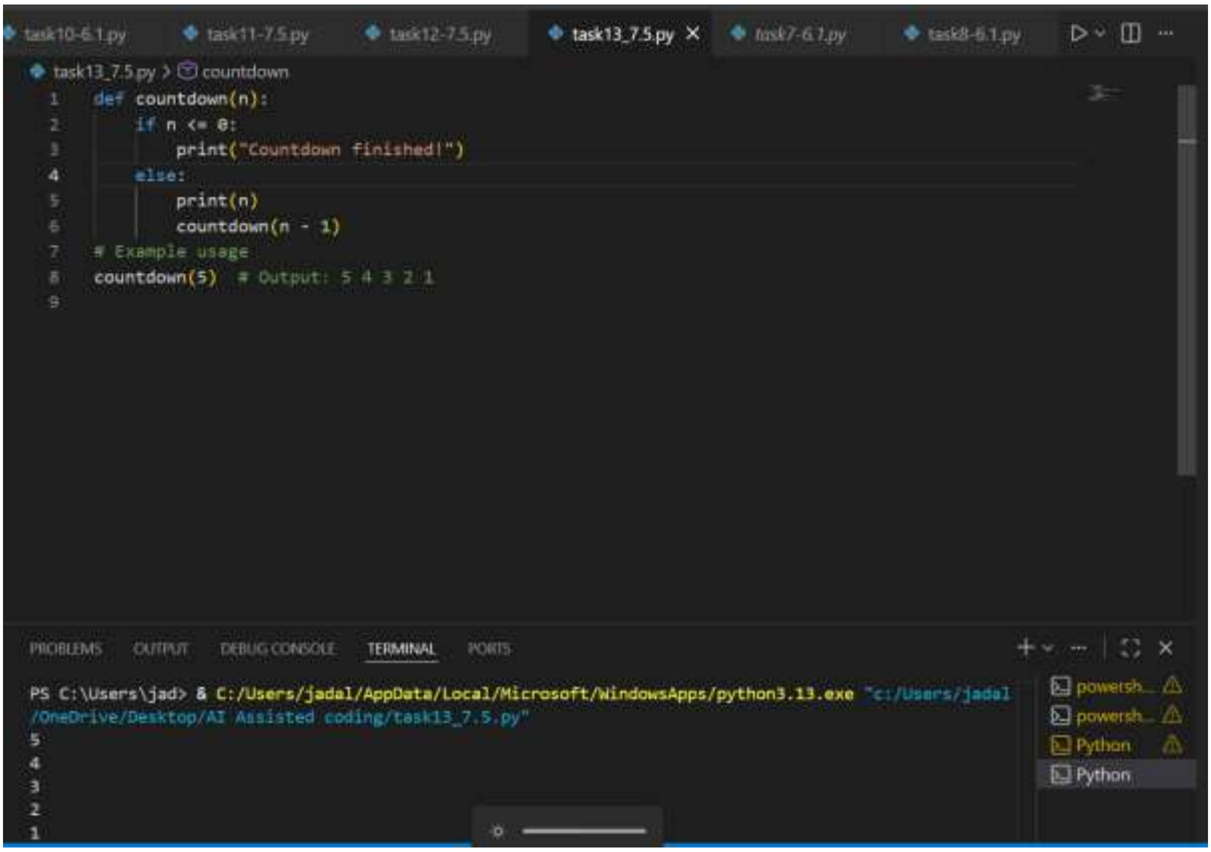
countdown(n):

print(n)

return countdown(n-1) countdown(5)

Expected Output : Correct recursion with stopping condition.

<u>Given Code and Output:</u>



```python
def countdown(n):
    if n <= 0:
        print("Countdown finished!")
    else:
        print(n)
        countdown(n - 1)
# Example usage
countdown(5)  # Output: 5 4 3 2 1
```

```
PS C:\Users\jad> & C:/Users/jadal/AppData/Local/Microsoft/WindowsApps/python3.13.exe "c:/Users/jadal
/OneDrive/Desktop/AI Assisted coding/task13_7.5.py"
5
4
3
2
1
```

<u>Task 4 (Dictionary Key Error)</u>

Task: Analyze given code where a missing dictionary key causes

error. Use AI to fix it.

# Bug: Accessing non-existing key

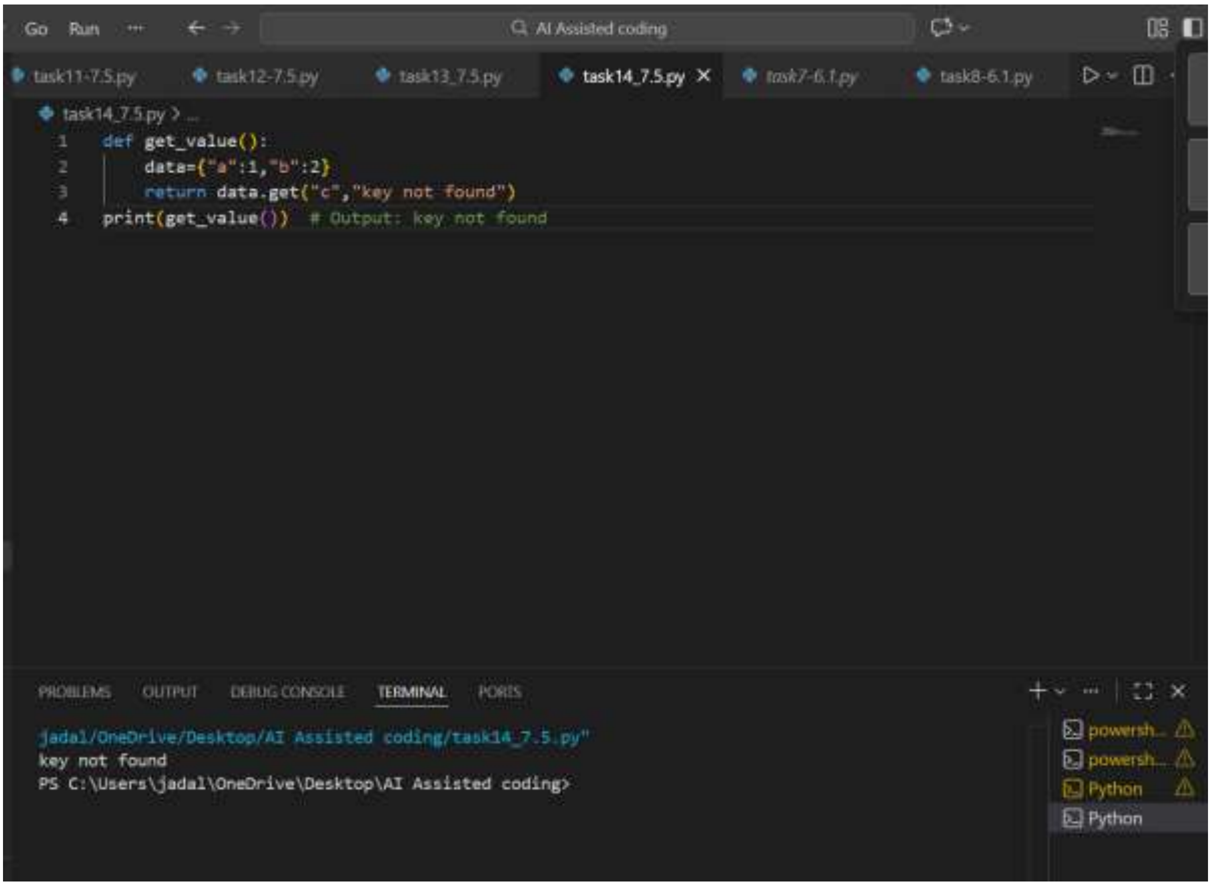
def get_value(): data = {"a": 1,

"b": 2} return data["c"]

print(get_value())

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

<u>Given Code and output:</u>



```python
def get_value():
    data={"a":1,"b":2}
    return data.get("c","key not found")
print(get_value())  # Output: key not found
```

```
jadal/OneDrive/Desktop/AI Assisted coding/task14_7.5.py"
key not found
PS C:\Users\jadal\OneDrive\Desktop\AI Assisted coding>
```

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():

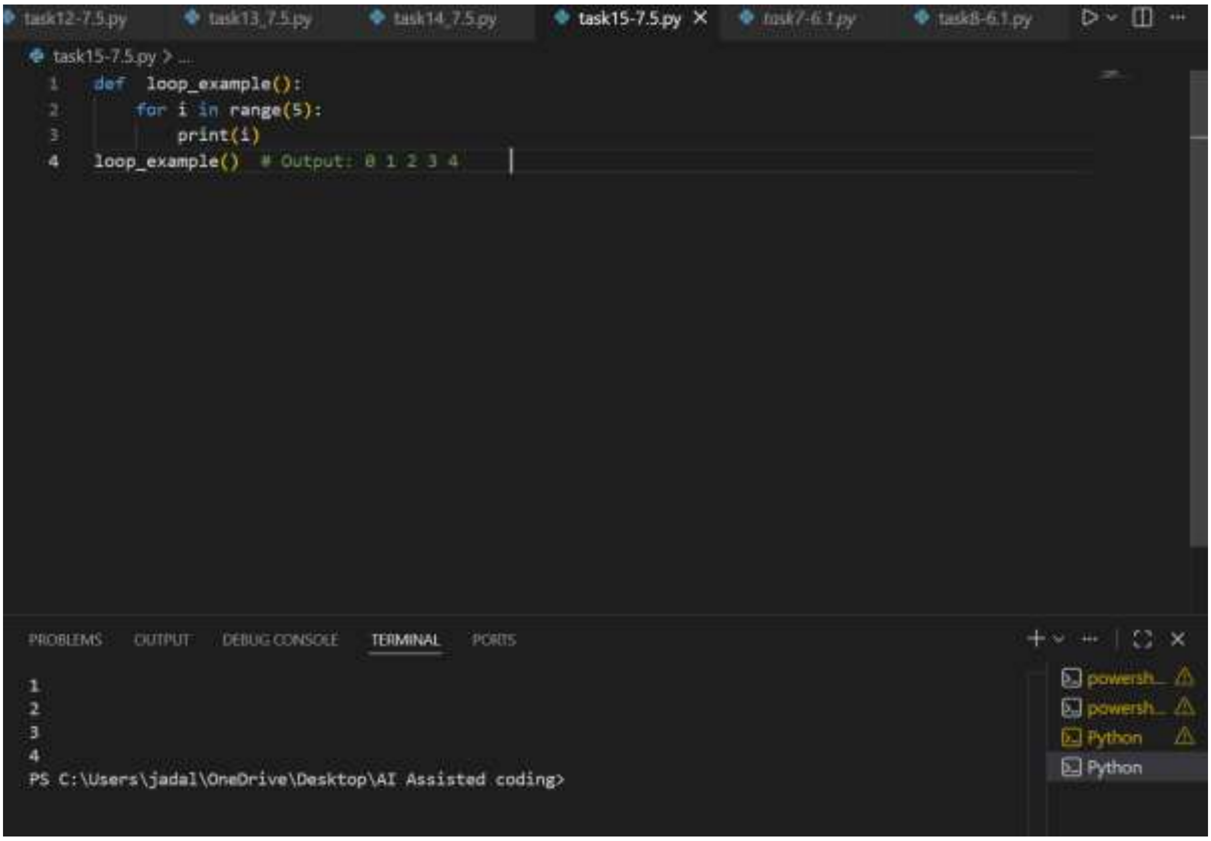
i = 0 while

i < 5:

print(i)

Expected Output: Corrected loop increments i.

Given Code and Output:



```python
def  loop_example():
    for i in range(5):
        print(i)
loop_example()  # Output: 0 1 2 3 4
```

```
1
2
3
4
PS C:\Users\jadal\OneDrive\Desktop\AI Assisted coding>
```

Task 6 (Unpacking Error – Wrong Variables)

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

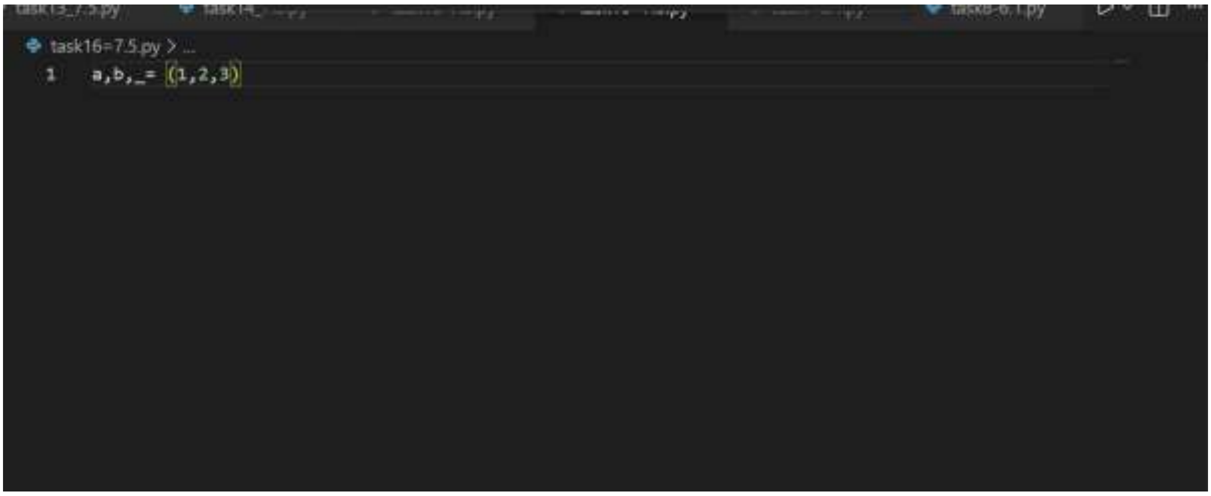
fix it.

# Bug: Wrong unpacking

a, b = (1, 2, 3)

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

Given Code:



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

Task 7 (Mixed Indentation – Tabs vs Spaces)

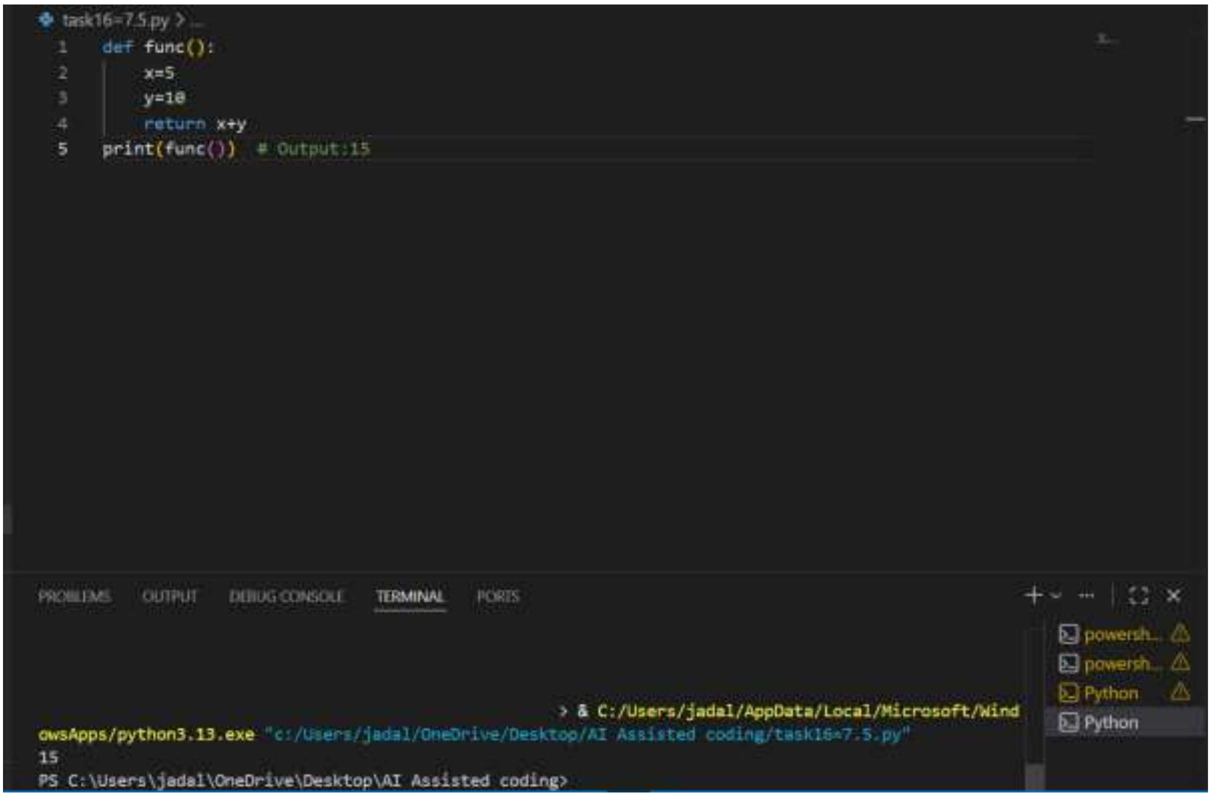
Task: Analyze given code where mixed indentation breaks

execution. Use AI to fix it. # Bug: Mixed indentation def

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

Expected Output : Consistent indentation applied.

Given Code and Output:



task16-7.5.py > ...

```
def func():
    x=5
    y=10
    return x+y
print(func())  # Output:15
```

```
> & C:/Users/jadal/AppData/Local/Microsoft/WindowsApps/python3.13.exe "c:/Users/jadal/OneDrive/Desktop/AI Assisted coding/task16-7.5.py"
15
PS C:\Users\jadal\OneDrive\Desktop\AI Assisted coding>
```

Task 8 (Import Error – Wrong Module Usage)

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

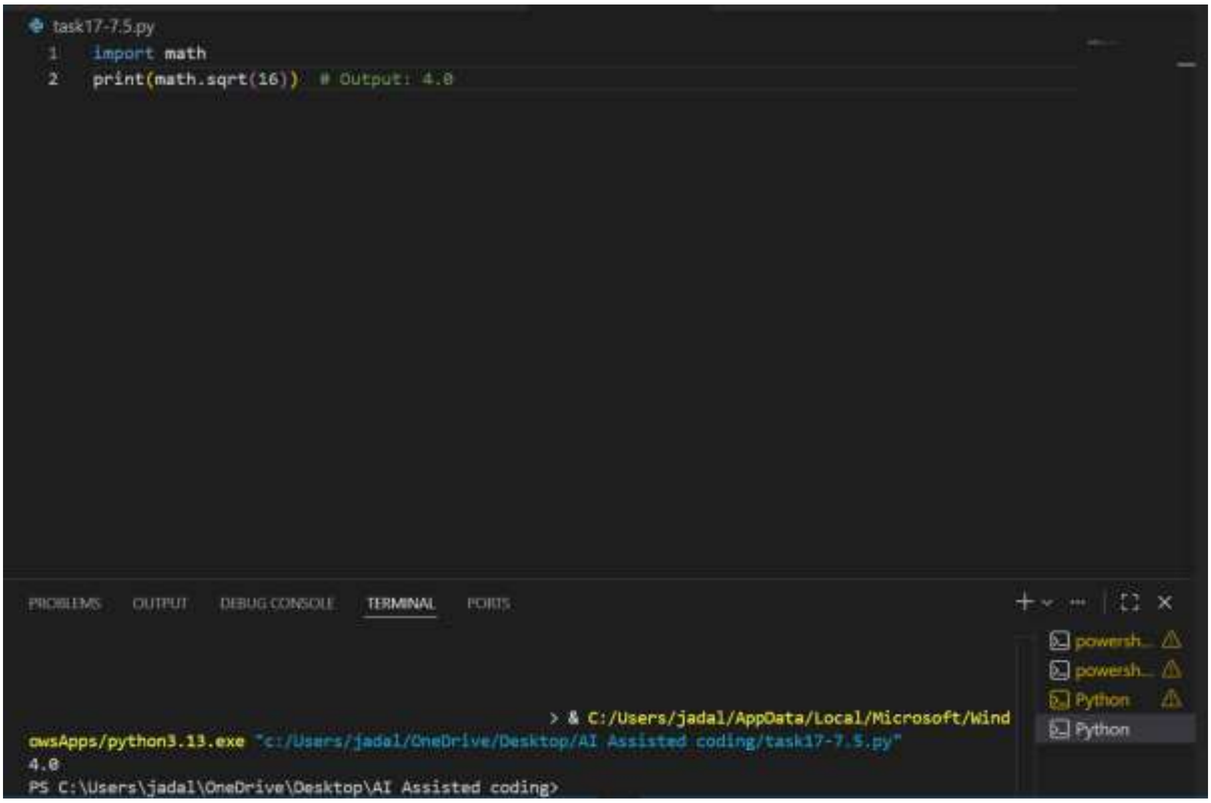
# Bug: Wrong import

import maths

print(maths.sqrt(16))

Expected Output: Corrected to import math Given

Code and Output:



```python
import math
print(math.sqrt(16))  # Output: 4.0
```

```
                                                  > & C:/Users/jadal/AppData/Local/Microsoft/Wind
owsApps/python3.13.exe "c:/Users/jadal/OneDrive/Desktop/AI Assisted coding/task17-7.5.py"
4.0
PS C:\Users\jadal\OneDrive\Desktop\AI Assisted coding>
```