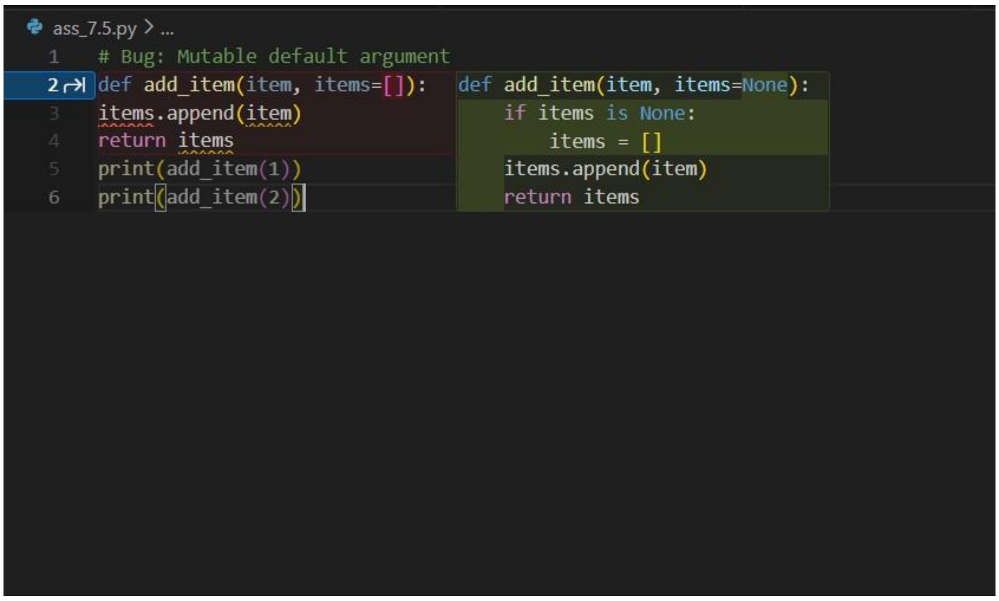
# ASSIGNMENT-7.5

NAME: Sreeja Thalla

HALLTICKET-2303A51220

BATCH-04

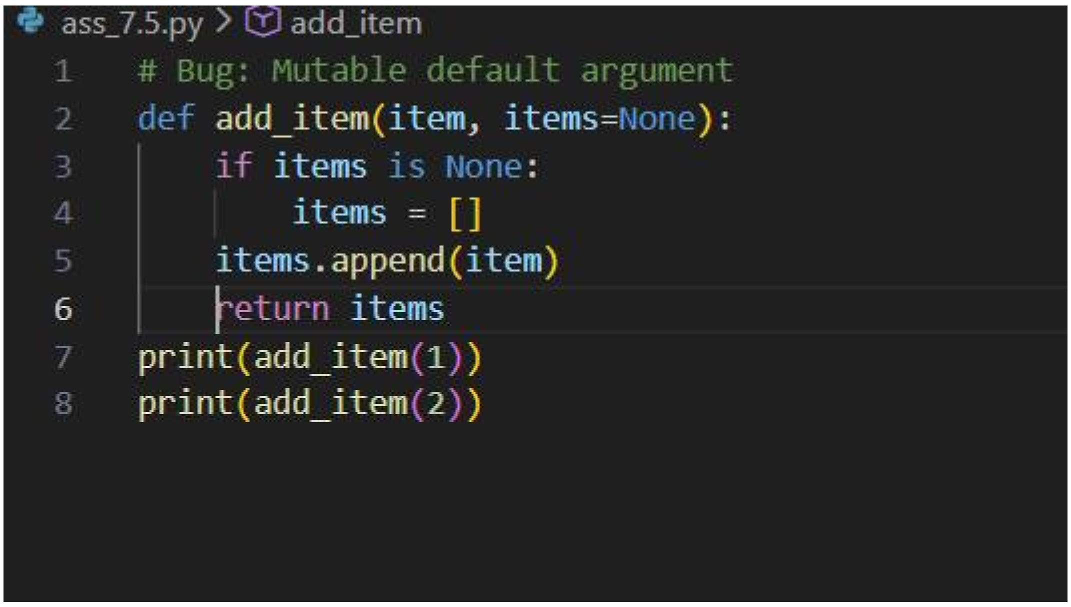
## 1) INPUT 1:



ass_7.5.py > ...

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

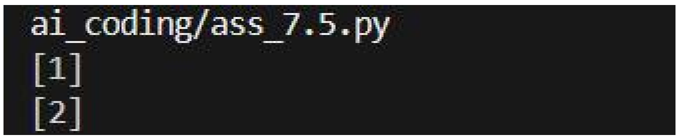
```python
def add_item(item, items=None):
    if items is None:
        items = []
    items.append(item)
    return items
```



ass_7.5.py > add_item

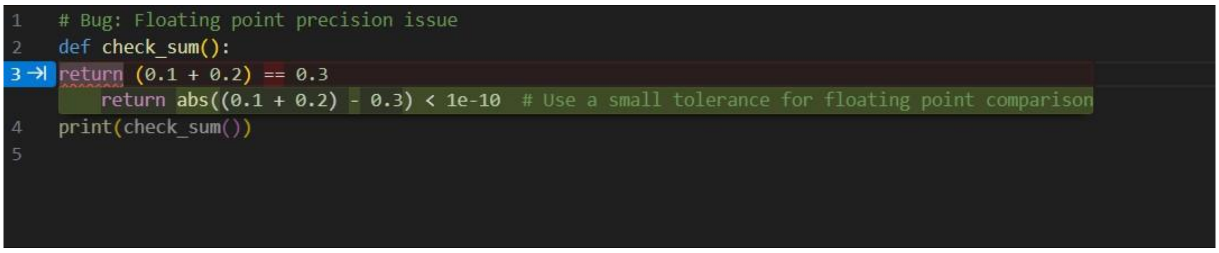
```python
# Bug: Mutable default argument
def add_item(item, items=None):
    if items is None:
        items = []
    items.append(item)
    return items
print(add_item(1))
print(add_item(2))
```

OUTPUT:

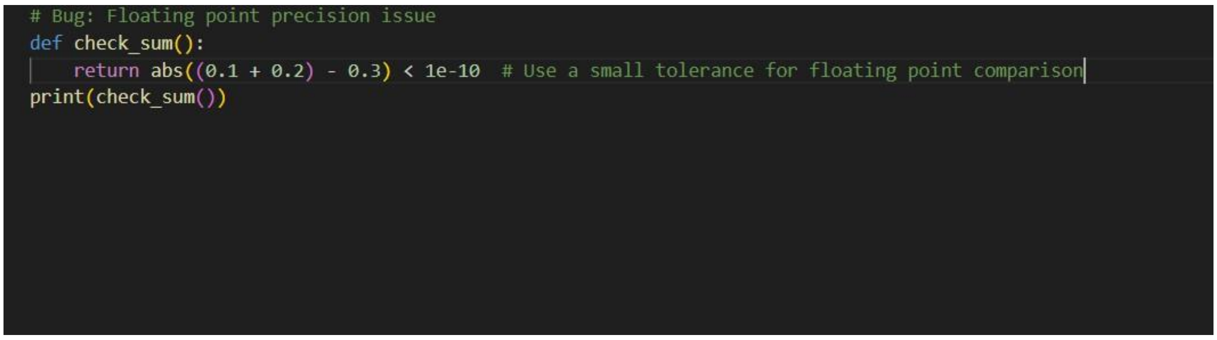


```
ai_coding/ass_7.5.py
[1]
[2]
```

## 2) INPUT:

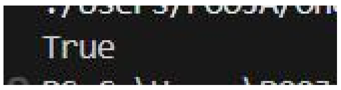


```
# Bug: Floating point precision issue
def check_sum():
return (0.1 + 0.2) == 0.3
    return abs((0.1 + 0.2) - 0.3) < 1e-10  # Use a small tolerance for floating point comparison
print(check_sum())
```



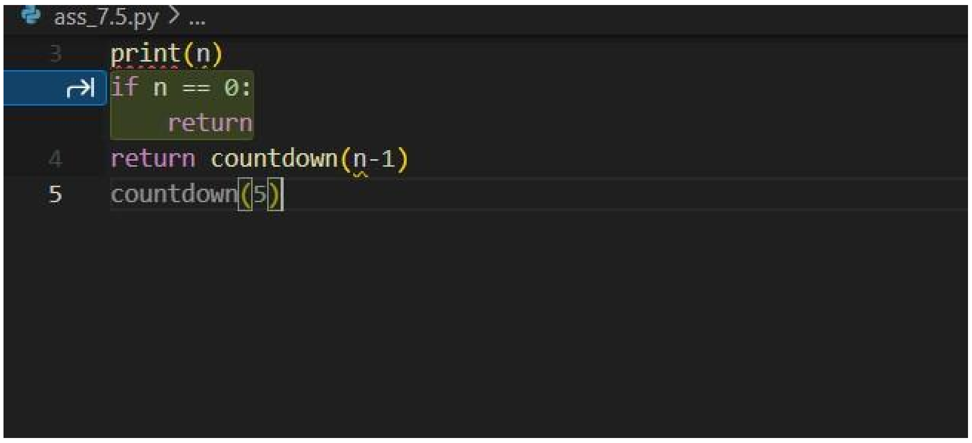
```
# Bug: Floating point precision issue
def check_sum():
    return abs((0.1 + 0.2) - 0.3) < 1e-10  # Use a small tolerance for floating point comparison
print(check_sum())
```

OUTPUT:

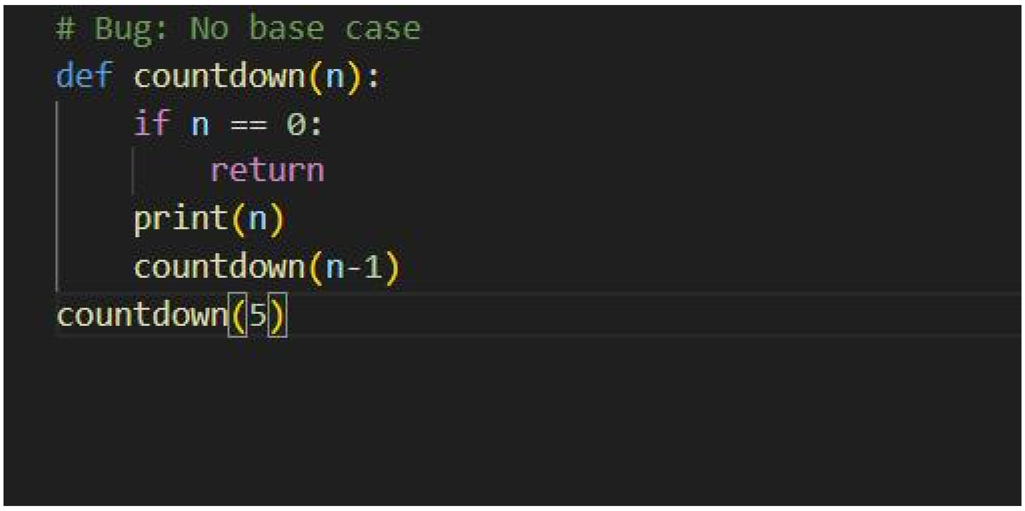


```
True
```

## 3) INPUT:

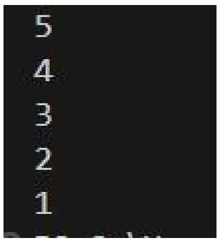

ass_7.5.py > ...

```
print(n)
if n == 0:
    return
return countdown(n-1)
countdown(5)
```



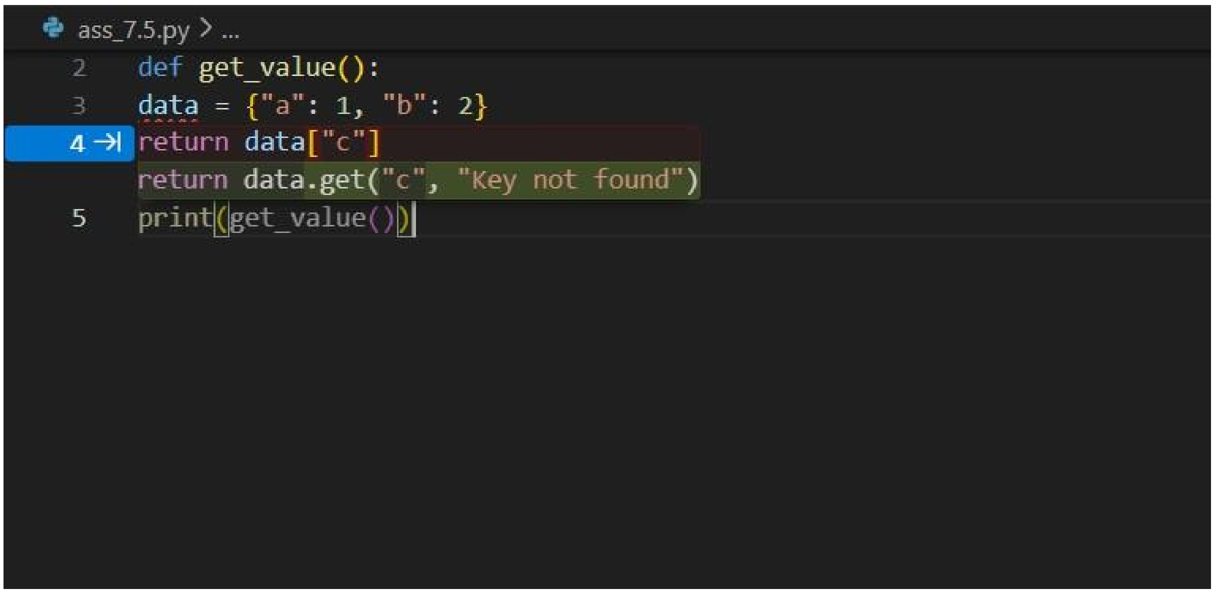
```
# Bug: No base case
def countdown(n):
    if n == 0:
        return
    print(n)
    countdown(n-1)
countdown(5)
```

OUTPUT:

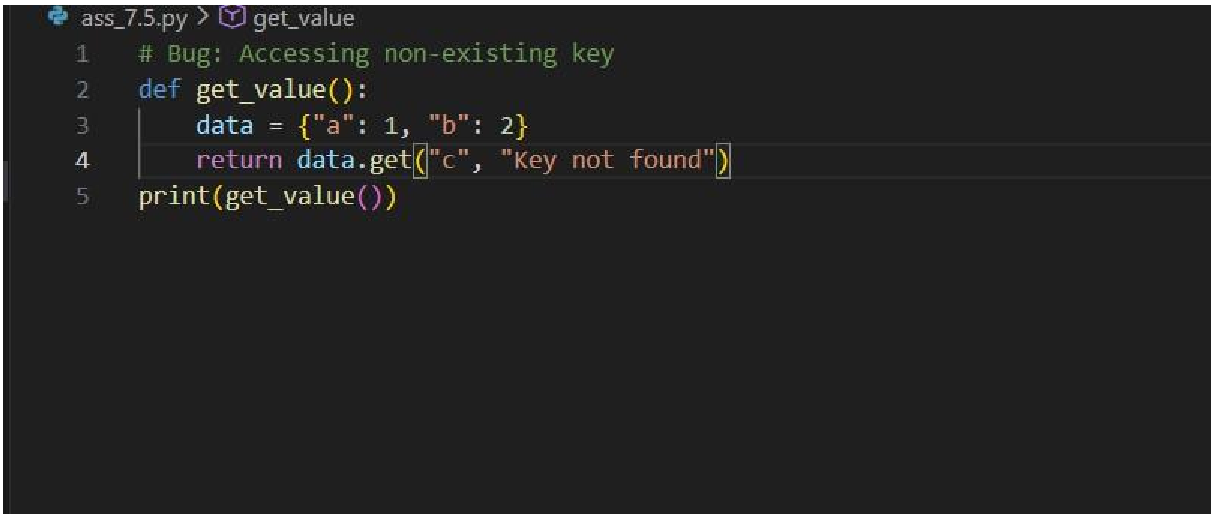


```
5
4
3
2
1
```

4) INPUT:

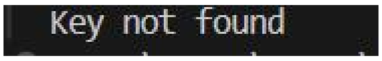


```
ass_7.5.py > ...
def get_value():
data = {"a": 1, "b": 2}
return data["c"]
return data.get("c", "Key not found")
print(get_value())
```



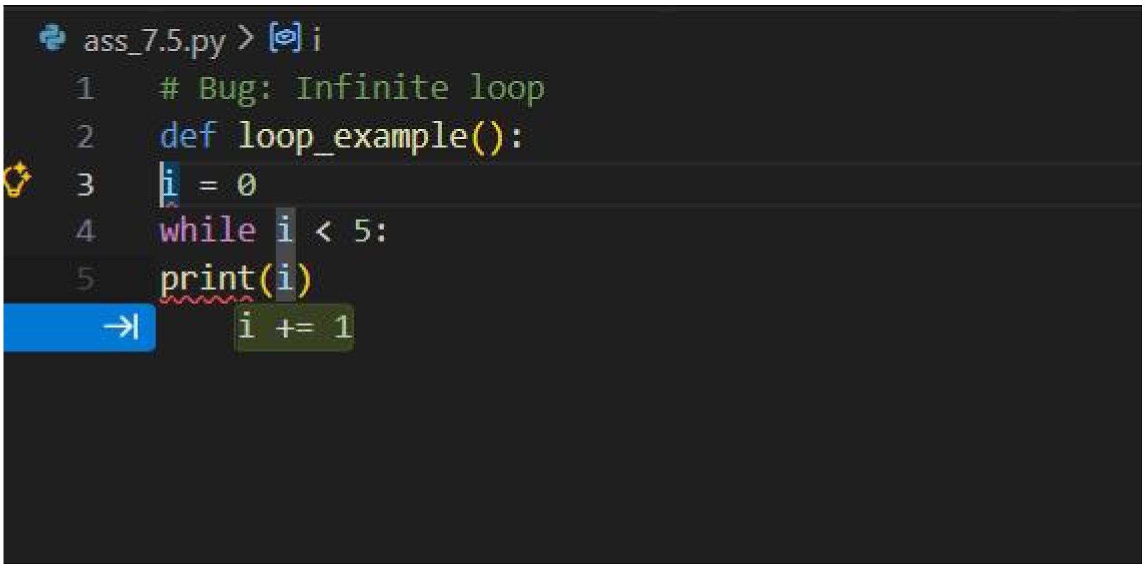
```
ass_7.5.py > get_value
# Bug: Accessing non-existing key
def get_value():
    data = {"a": 1, "b": 2}
    return data.get("c", "Key not found")
print(get_value())
```

OUTPUT:



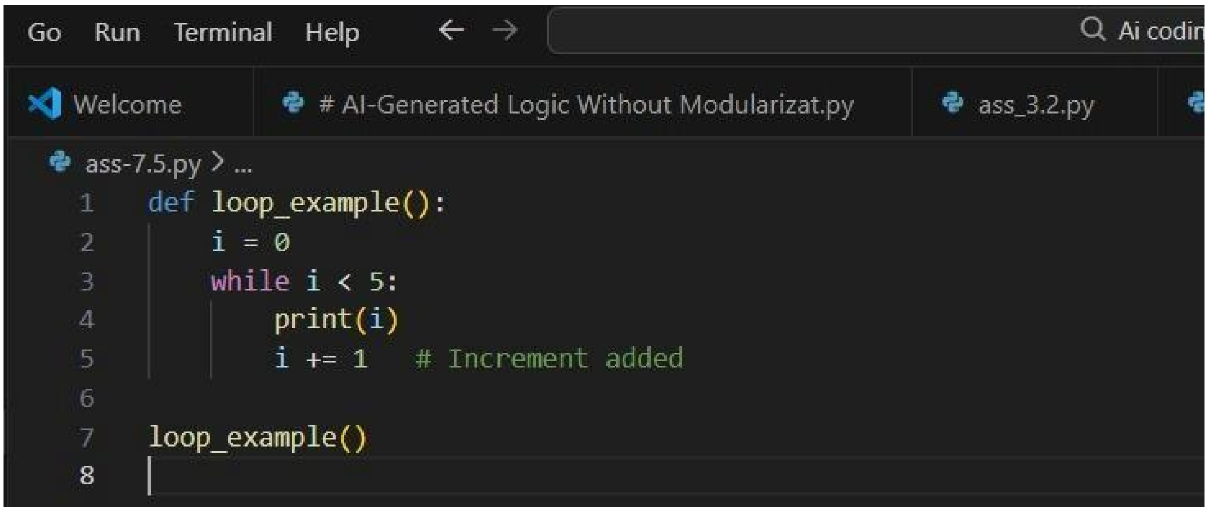
```
Key not found
```

5) INPUT:



ass_7.5.py > i

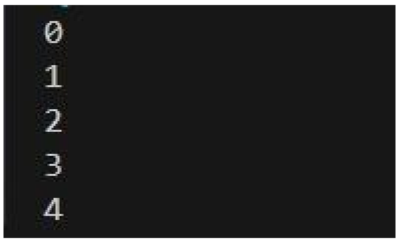
```
# Bug: Infinite loop
def loop_example():
i = 0
while i < 5:
print(i)
    i += 1
```



ass-7.5.py > ...

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

loop_example()
```

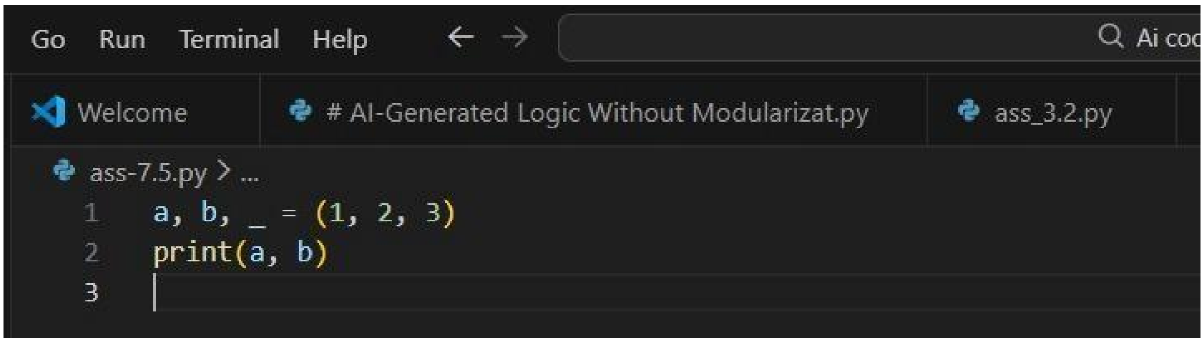
OUTPUT:



```
0
1
2
3
4
```

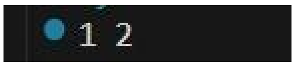
6) INPUT:

# Bug: Wrong unpacking
a, b = (1, 2, 3)
Expected Output: Correct unpacking or using _ for extra values.



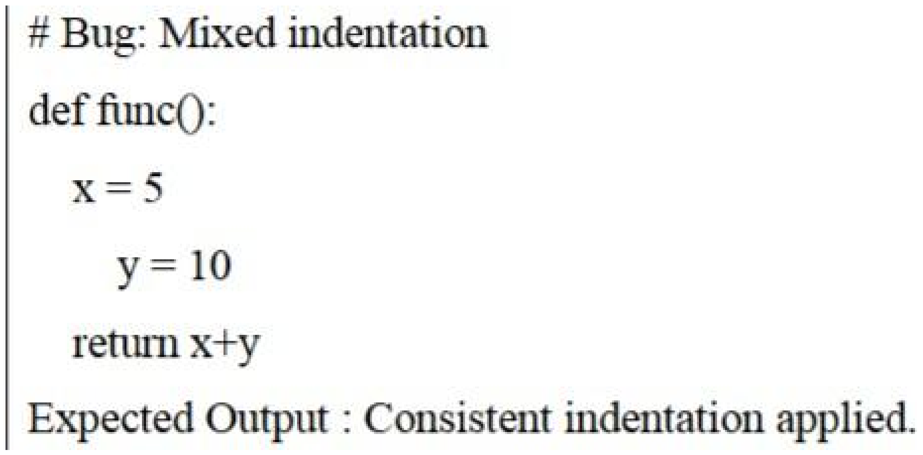
```python
a, b, _ = (1, 2, 3)
print(a, b)

```

OUTPUT:



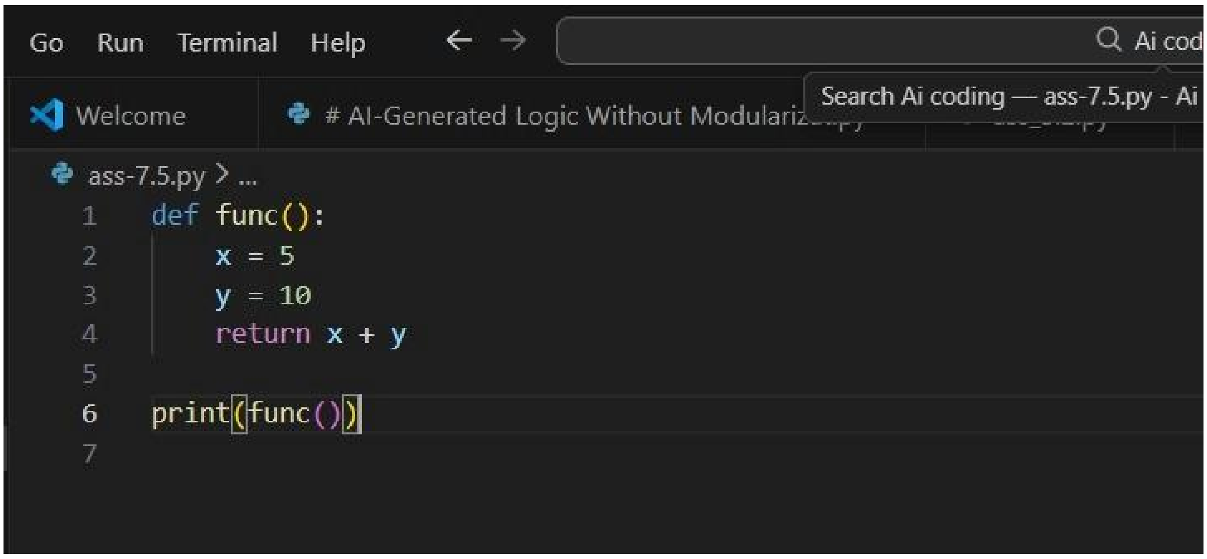
```
1 2
```

7) INPUT:



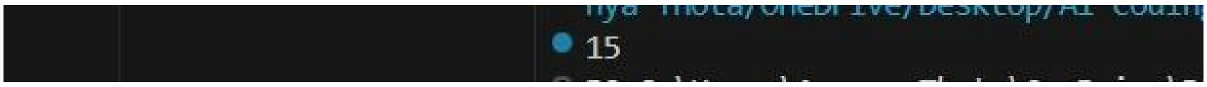
```
# Bug: Mixed indentation
def func():
   x = 5
      y = 10
   return x+y
```

Expected Output : Consistent indentation applied.


Go Run Terminal Help
Ai cod
Search Ai coding — ass-7.5.py - Ai
Welcome
# AI-Generated Logic Without Modulariz
ass-7.5.py > ...

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

print(func())
```

OUTPUT:



```
15
```

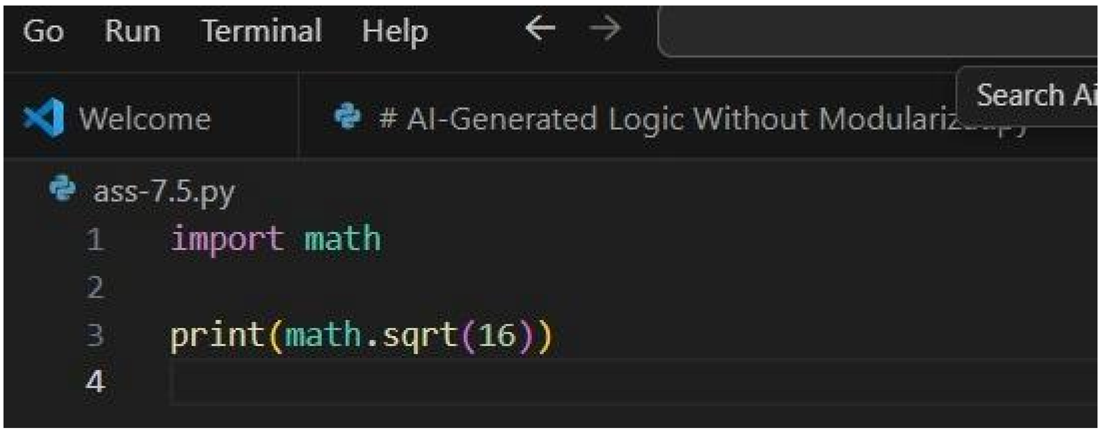
8) INPUT:

# Bug: Wrong import

import maths

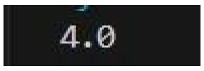
print(maths.sqrt(16))

Expected Output: Corrected to import math



```python
import math

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

OUTPUT:



```
4.0
```