NAME:A.AKSHITH REDDY

HALL TICKET-2303A51177

BATCH-03

1)INPUT 1:

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

```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:

```
1    # Bug: Floating point precision issue
2    def check_sum():
3 →|  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
4    print(check_sum())
5
```

```
# 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:

```
:/users/rooja/un
True
```

3)INPUT:

```
 ass_7.5.py > ...
  3     print(n)
   ⇥ if n == 0:
            return
  4     return countdown(n-1)
  5     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 > ...
  2     def get_value():
  3     data = {"a": 1, "b": 2}
  4 →|  return data["c"]
        return data.get("c", "Key not found")
  5     print(get_value())
```

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

OUTPUT:

```
Key not found
```

5)INPUT:

```
ass_7.5.py > ⊙ i
1     # Bug: Infinite loop
2     def loop_example():
3     i = 0
4     while i < 5:
5     print(i)
→|        i += 1
```
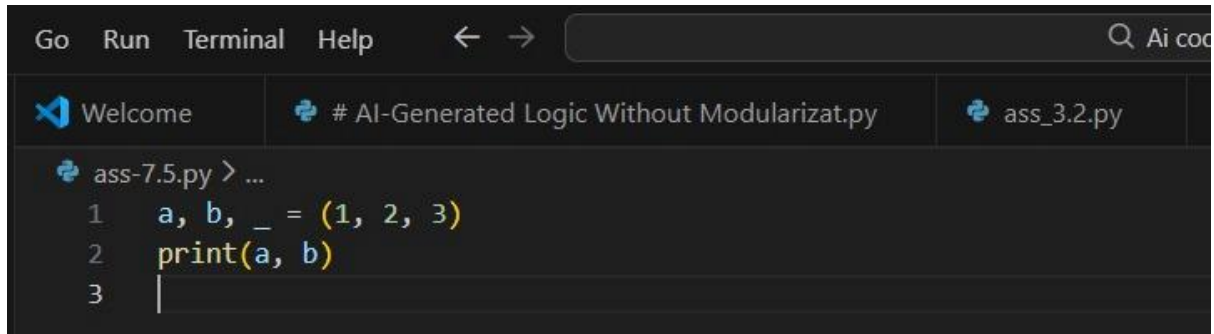
```
Go   Run   Terminal   Help        ← →                              Q Ai codin

Welcome          # AI-Generated Logic Without Modularizat.py       ass_3.2.py

ass-7.5.py > ...
1     def loop_example():
2         i = 0
3         while i < 5:
4             print(i)
5             i += 1    # Increment added
6
7     loop_example()
8     |
```
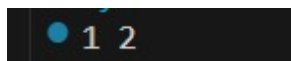
OUTPUT:

```
0
1
2
3
4
```

6)INPUT:

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

Welcome   # AI-Generated Logic Without Modularizat.py   ass_3.2.py

```python
ass-7.5.py > ...
1    a, b, _ = (1, 2, 3)
2    print(a, b)
3
```

OUTPUT:

```
1 2
```

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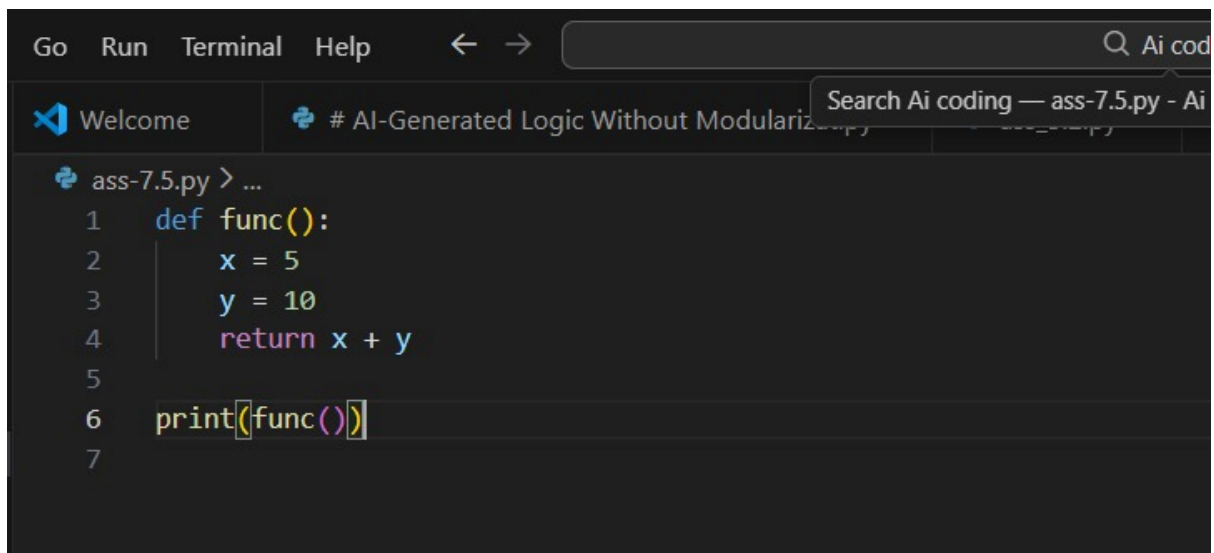
```python
# Bug: Mixed indentation
def func():
    x = 5
        y = 10
    return x+y
```

Expected Output : Consistent indentation applied.

```
Go   Run   Terminal   Help        ←  →                                          Q  Ai cod

◆) Welcome              ♦ # AI-Generated Logic Without Modulariz...py    Search Ai coding — ass-7.5.py - Ai

♦ ass-7.5.py > ...
   1    def func():
   2        x = 5
   3        y = 10
   4        return x + y
   5
   6    print(func())
   7
```

OUTPUT:

```
                                    nya  mota/OneDrive/Desktop/AI codin
                              ● 15
```
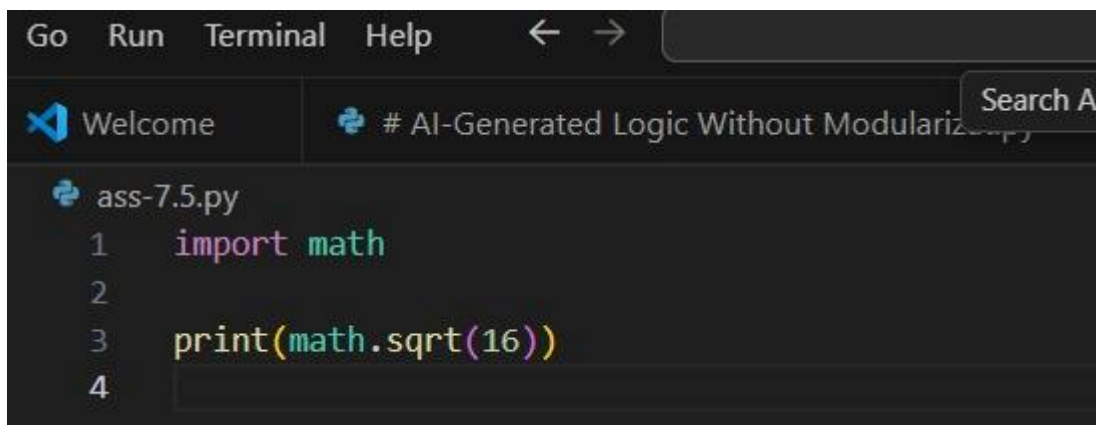
8)INPUT:

Bug: Wrong import

import maths

print(maths.sqrt(16))

Expected Output: Corrected to import math

```
Go   Run   Terminal   Help        ←  →

◆) Welcome              ♦ # AI-Generated Logic Without Modulariz...py    Search Ai

♦ ass-7.5.py
   1    import math
   2
   3    print(math.sqrt(16))
   4
```

OUTPUT:

```
4.0
```