ARTIFICIAL INTELLIGENCE(AI)

BSCYS-3rd Semester

Fall 2025

Submitted To:**Mam Fatiha Arshad**

Submitted By: **M.Umer**

Reg No: **(24-CyS-024)**

BS CYBER SECURITY PROGRAM

DEPARTMENT OF COMPUTER SCIENCE HITEC UNIVERSITY TAXILA

Task 1:

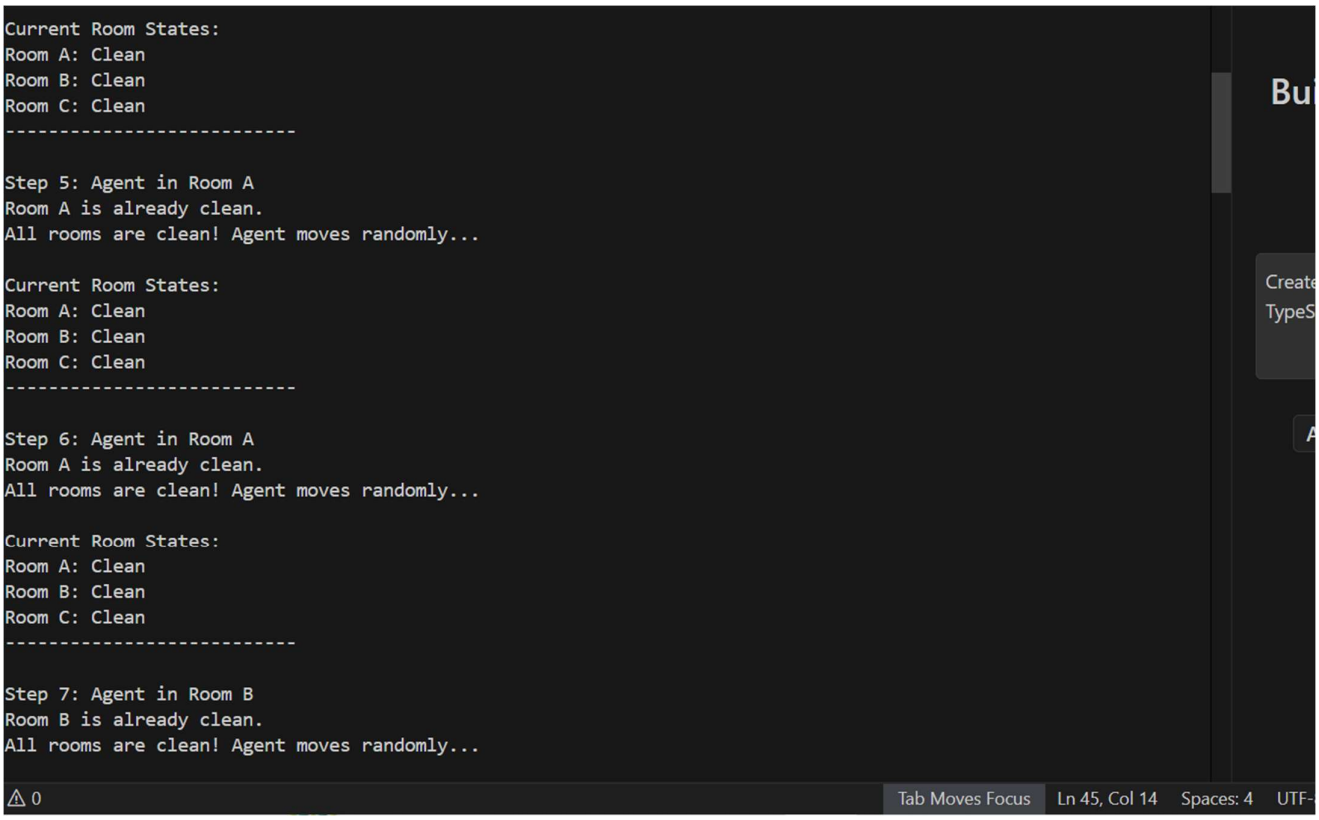
Code:

```python
import random

# Define environment with 3 rooms
rooms = {
    'A': 'Dirty',
    'B': 'Dirty',
    'C': 'Dirty'
}

# Counter to count cleaning actions
clean_count = 0

# Function to display environment status
def show_environment():
    print("\nCurrent Room States:")
    for room, state in rooms.items():
        print(f"Room {room}: {state}")
    print("--------------------------")

# Agent starts in room A
current_room = 'A'

# Agent process
for step in range(10):  # Run 10 steps
    print(f"\nStep {step + 1}: Agent in Room {current_room}")

    # If current room is dirty → clean it
    if rooms[current_room] == 'Dirty':
        print(f"Cleaning Room {current_room}...")
        rooms[current_room] = 'Clean'
        clean_count += 1
    else:
        print(f"Room {current_room} is already clean.")

    # Check if all rooms are clean
    if all(state == 'Clean' for state in rooms.values()):
        print("All rooms are clean! Agent moves randomly...")
        current_room = random.choice(list(rooms.keys()))  # Move randomly
    else:
        # Move to next room in sequence
        if current_room == 'A':
            current_room = 'B'
        elif current_room == 'B':
            current_room = 'C'
        else:
            current_room = 'A'

    show_environment()

print(f"\nTotal times cleaned: {clean_count}")
```

Output:

```
Step 1: Agent in Room A
Cleaning Room A...

Current Room States:
Room A: Clean
Room B: Dirty
Room C: Dirty
--------------------------

Step 2: Agent in Room B
Cleaning Room B...

Current Room States:
Room A: Clean
Room B: Clean
Room C: Dirty
--------------------------

Step 3: Agent in Room C
Cleaning Room C...
All rooms are clean! Agent moves randomly...

Current Room States:
Room A: Clean
Room B: Clean
Room C: Clean
--------------------------

Step 4: Agent in Room B
Room B is already clean.
All rooms are clean! Agent moves randomly...
```

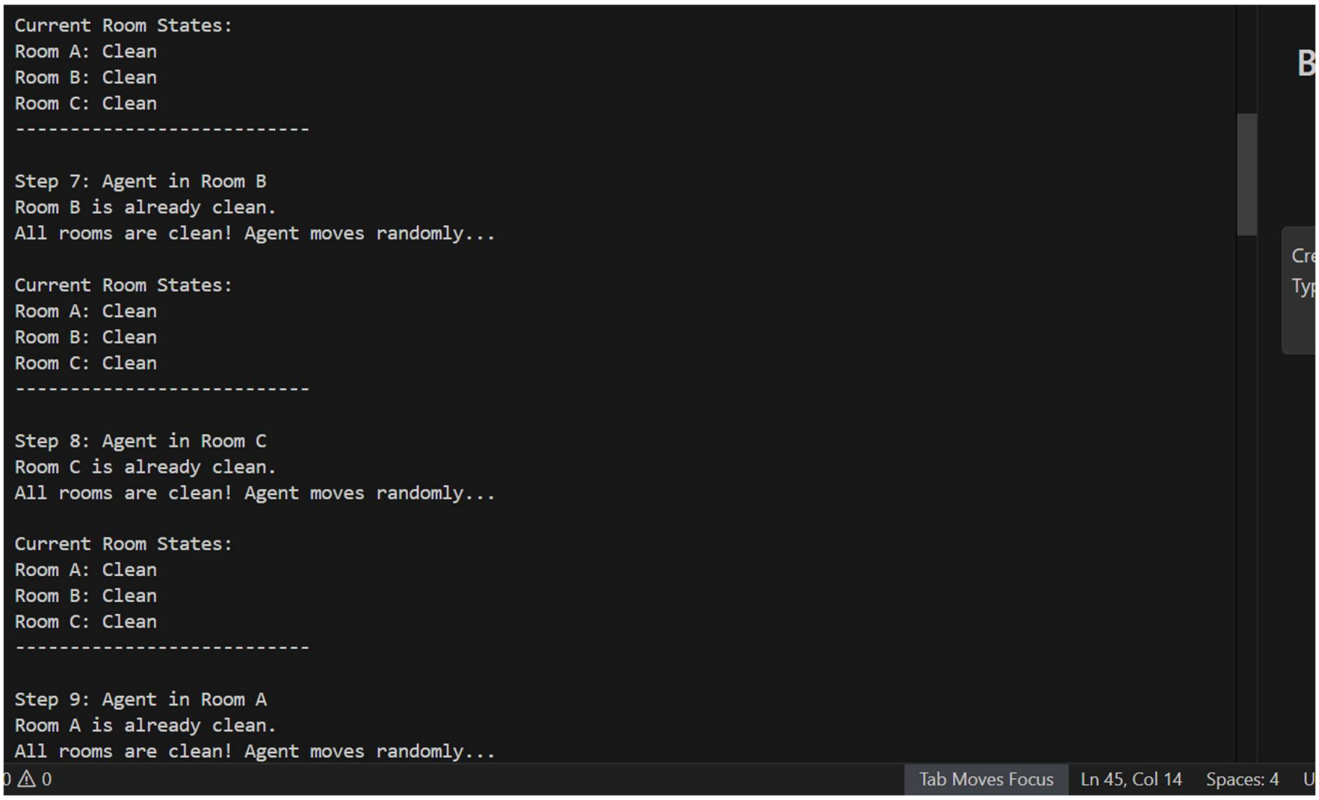

Output:


Current Room States:
Room A: Clean
Room B: Clean
Room C: Clean

Step 5: Agent in Room A
Room A is already clean.
All rooms are clean! Agent moves randomly...
Current Room States:
Room A: Clean
Room B: Clean
Room C: Clean

Step 6: Agent in Room A
Room A is already clean.
All rooms are clean! Agent moves randomly...
Current Room States:
Room A: Clean
Room B: Clean
Room C: Clean

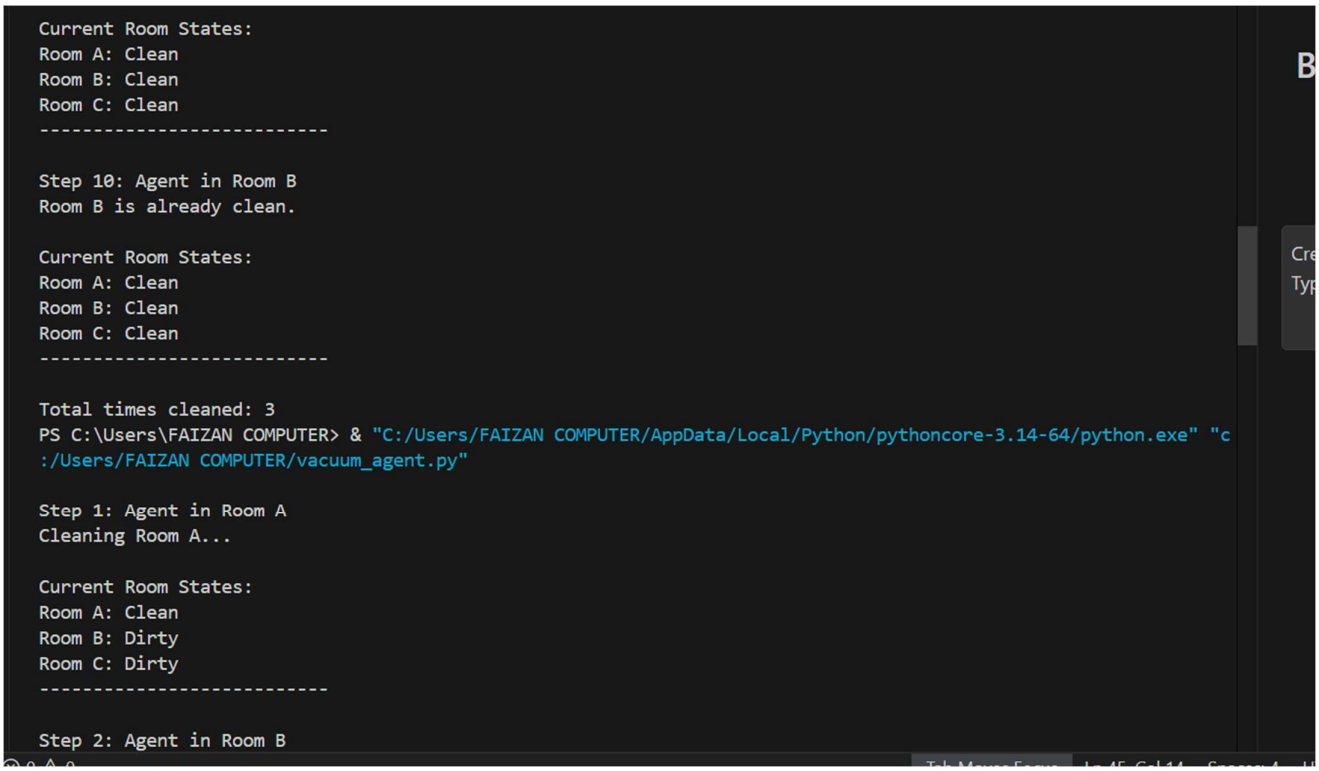
Step 7: Agent in Room B
Room B is already clean.
All rooms are clean! Agent moves randomly...
⚠ 0
Tab Moves Focus
Ln 45, Col 14
Spaces: 4


Current Room States:
Room A: Clean
Room B: Clean
Room C: Clean

Step 7: Agent in Room B
Room B is already clean.
All rooms are clean! Agent moves randomly...
Current Room States:
Room A: Clean
Room B: Clean
Room C: Clean

Step 8: Agent in Room C
Room C is already clean.
All rooms are clean! Agent moves randomly...
Current Room States:
Room A: Clean
Room B: Clean
Room C: Clean

Step 9: Agent in Room A
Room A is already clean.
All rooms are clean! Agent moves randomly...
⚠ 0
Tab Moves Focus
Ln 45, Col 14
Spaces: 4


Current Room States:
Room A: Clean
Room B: Clean
Room C: Clean

Step 10: Agent in Room B
Room B is already clean.
Current Room States:
Room A: Clean
Room B: Clean
Room C: Clean

Total times cleaned: 3
PS C:\Users\FAIZAN COMPUTER> & "C:/Users/FAIZAN COMPUTER/AppData/Local/Python/pythoncore-3.14-64/python.exe" "c
:/Users/FAIZAN COMPUTER/vacuum_agent.py"
Step 1: Agent in Room A
Cleaning Room A...
Current Room States:
Room A: Clean
Room B: Dirty
Room C: Dirty

Step 2: Agent in Room B

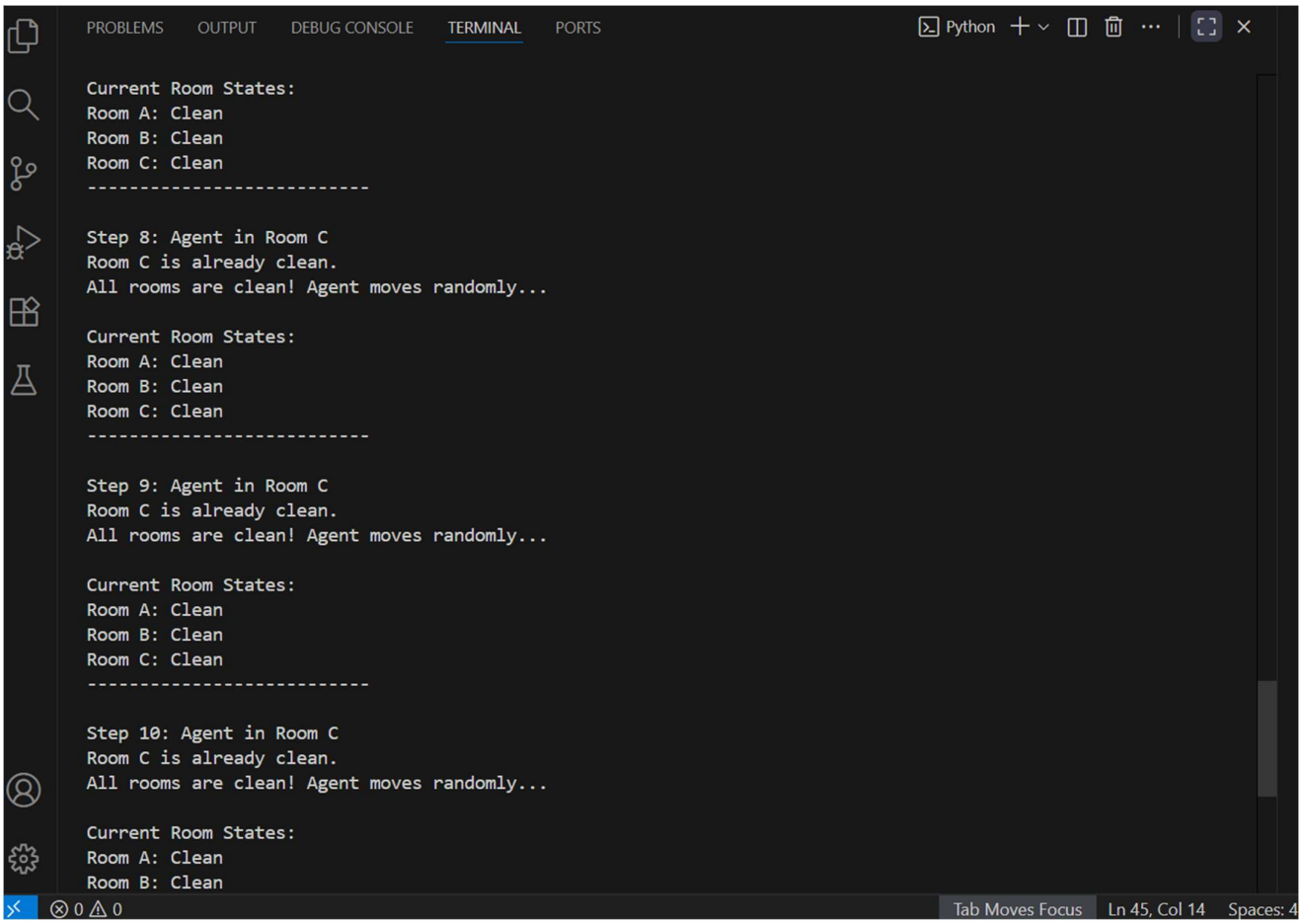
Output:

```
Step 2: Agent in Room B
Cleaning Room B...

Current Room States:
Room A: Clean
Room B: Clean
Room C: Dirty
--------------------------

Step 3: Agent in Room C
Cleaning Room C...
All rooms are clean! Agent moves randomly...

Current Room States:
Room A: Clean
Room B: Clean
Room C: Clean
--------------------------

Step 4: Agent in Room A
Room A is already clean.
All rooms are clean! Agent moves randomly...

Current Room States:
Room A: Clean
Room B: Clean
Room C: Clean
--------------------------

Step 5: Agent in Room B
Room B is already clean.
All rooms are clean! Agent moves randomly...
```

```
Current Room States:
Room A: Clean
Room B: Clean
Room C: Clean
--------------------------

Step 6: Agent in Room C
Room C is already clean.
All rooms are clean! Agent moves randomly...

Current Room States:
Room A: Clean
Room B: Clean
Room C: Clean
--------------------------

Step 7: Agent in Room B
Room B is already clean.
All rooms are clean! Agent moves randomly...

Current Room States:
Room A: Clean
Room B: Clean
Room C: Clean
--------------------------

Step 8: Agent in Room C
Room C is already clean.
All rooms are clean! Agent moves randomly...

Current Room States:
```

0 ⚠ 0 Tab Moves Focus Ln 45, Col 14

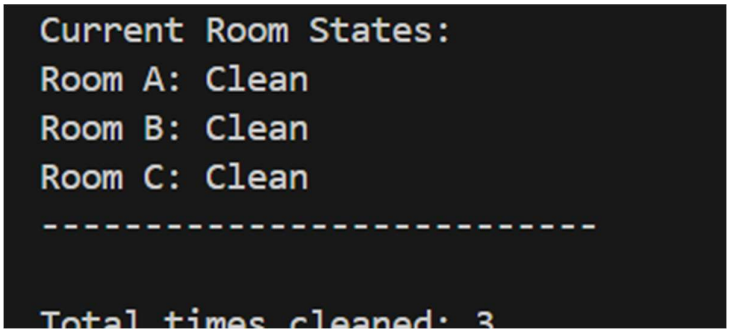
Output:


PROBLEMS
OUTPUT
DEBUG CONSOLE
TERMINAL
PORTS
Python
Current Room States:
Room A: Clean
Room B: Clean
Room C: Clean

Step 8: Agent in Room C
Room C is already clean.
All rooms are clean! Agent moves randomly...
Current Room States:
Room A: Clean
Room B: Clean
Room C: Clean

Step 9: Agent in Room C
Room C is already clean.
All rooms are clean! Agent moves randomly...
Current Room States:
Room A: Clean
Room B: Clean
Room C: Clean

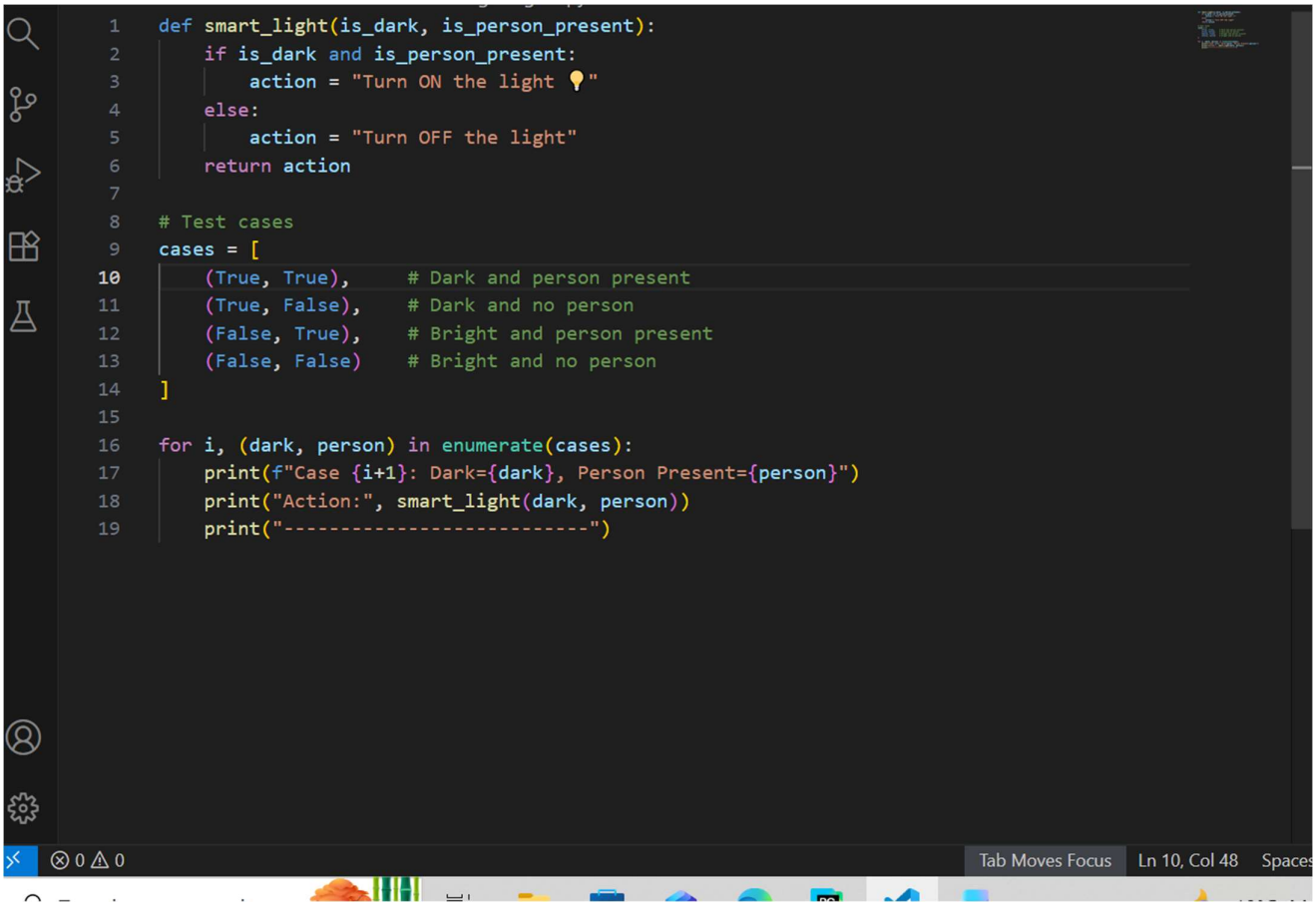
Step 10: Agent in Room C
Room C is already clean.
All rooms are clean! Agent moves randomly...
Current Room States:
Room A: Clean
Room B: Clean
0 0
Tab Moves Focus
Ln 45, Col 14
Spaces: 4


Current Room States:
Room A: Clean
Room B: Clean
Room C: Clean

Total times cleaned: 3

Task 2:

Code:



```python
def smart_light(is_dark, is_person_present):
    if is_dark and is_person_present:
        action = "Turn ON the light 💡"
    else:
        action = "Turn OFF the light"
    return action

# Test cases
cases = [
    (True, True),     # Dark and person present
    (True, False),    # Dark and no person
    (False, True),    # Bright and person present
    (False, False)    # Bright and no person
]

for i, (dark, person) in enumerate(cases):
    print(f"Case {i+1}: Dark={dark}, Person Present={person}")
    print("Action:", smart_light(dark, person))
    print("--------------------------")
```

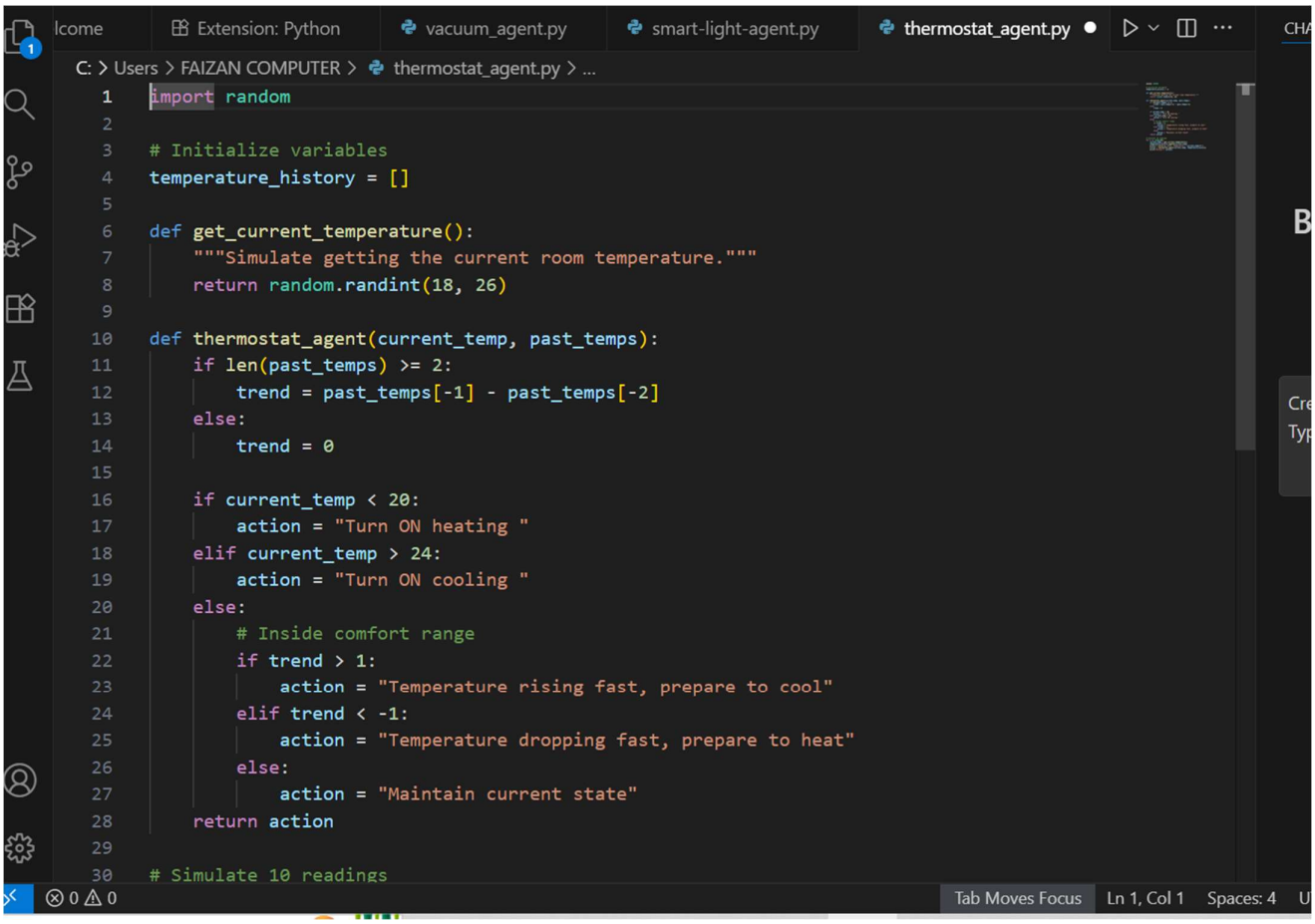

Output:

Output:

```
:/Users/FAIZAN COMPUTER/smart-light-agent.py"
Case 1: Dark=True, Person Present=True
Action: Turn ON the light 💡
--------------------------
Case 2: Dark=True, Person Present=False
Action: Turn OFF the light
--------------------------
Case 3: Dark=False, Person Present=True
Action: Turn OFF the light
--------------------------
Case 4: Dark=False, Person Present=False
Action: Turn OFF the light
--------------------------
PS C:\Users\FAIZAN COMPUTER> & "C:/Users/FAIZAN COMPUTER/AppData/Local/Python/pythoncore-3.14-64/python.exe
:/Users/FAIZAN COMPUTER/smart-light-agent.py"
Case 1: Dark=True, Person Present=True
Action: Turn ON the light 💡
--------------------------
Case 2: Dark=True, Person Present=False
Action: Turn OFF the light
--------------------------
Case 3: Dark=False, Person Present=True
Action: Turn OFF the light
--------------------------
Case 4: Dark=False, Person Present=False
Action: Turn OFF the light
--------------------------
PS C:\Users\FAIZAN COMPUTER>
```

Task 3:

Code:


lcome
Extension: Python
vacuum_agent.py
smart-light-agent.py
thermostat_agent.py
C: > Users > FAIZAN COMPUTER > thermostat_agent.py > ...
import random
# Initialize variables
temperature_history = []
def get_current_temperature():
"""Simulate getting the current room temperature."""
return random.randint(18, 26)
def thermostat_agent(current_temp, past_temps):
if len(past_temps) >= 2:
trend = past_temps[-1] - past_temps[-2]
else:
trend = 0
if current_temp < 20:
action = "Turn ON heating "
elif current_temp > 24:
action = "Turn ON cooling "
else:
# Inside comfort range
if trend > 1:
action = "Temperature rising fast, prepare to cool"
elif trend < -1:
action = "Temperature dropping fast, prepare to heat"
else:
action = "Maintain current state"
return action
Simulate 10 readings
0 0
Tab Moves Focus
Ln 1, Col 1
Spaces: 4

Output:

```python
# Simulate 10 readings
for i in range(10):
    current_temp = get_current_temperature()
    temperature_history.append(current_temp)
    print(f"\nReading {i+1}: Temperature = {current_temp}°C")
    action = thermostat_agent(current_temp, temperature_history)
    print("Action:", action)
```


Conclusion:

This in the modification experiment i understand how simple agents can exhibit structured and random behaviors while keeping track of their performance, helping to understand the basics of agent design. In the seconde task, rule-based reflex design enables basic automation effectively, showing how simple condition-action rules can create intelligent and energy-efficient smart systems.in the third task the model-based approach allows smarter and more stable control by using past information, making the thermostat more predictive, efficient, and adaptive to rapid changes in temperature.