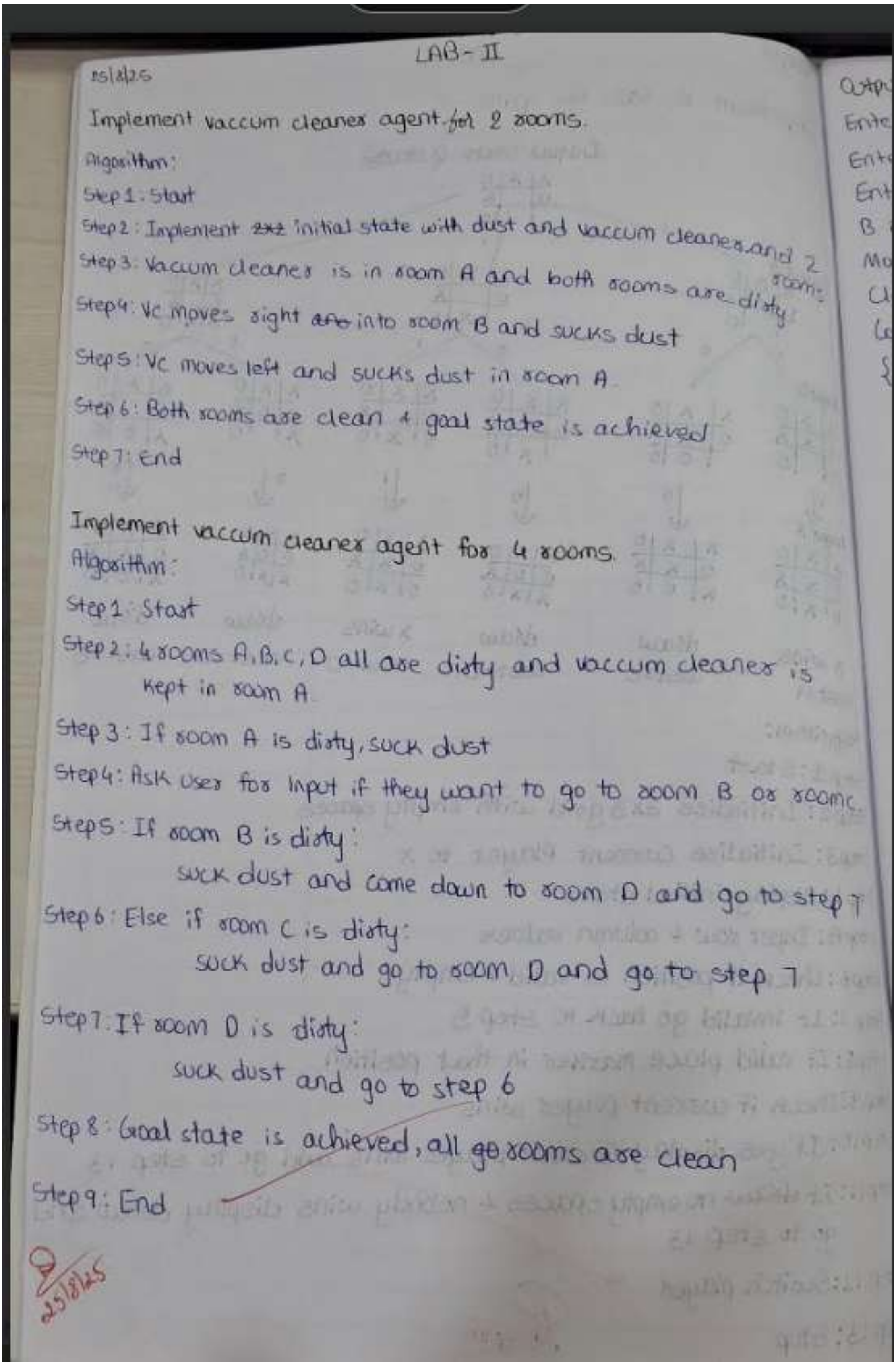
Implement vaccum cleaner agent with 2 room setup.



LAB - II

25/8/25

Implement vaccum cleaner agent for 2 rooms.

Algorithm:

```
Step 1: Start
Step 2: Implement ~~2x2~~ initial state with dust and vaccum cleaner and 2 rooms
Step 3: Vaccum cleaner is in room A and both rooms are dirty
Step 4: VC moves right ~~afo~~ into room B and sucks dust
Step 5: VC moves left and sucks dust in room A
Step 6: Both rooms are clean + goal state is achieved
Step 7: End
```

Implement vaccum cleaner agent for 4 rooms.

Algorithm:

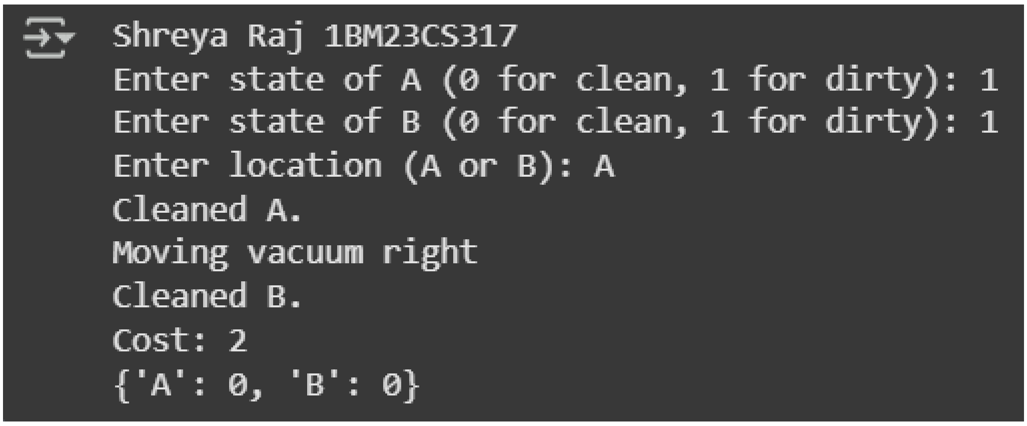
```
Step 1: Start
Step 2: 4 rooms A,B,C,D all are dirty and vaccum cleaner is
        kept in room A
Step 3: If room A is dirty, suck dust
Step 4: Ask user for input if they want to go to room B or roomC
Step 5: If room B is dirty:
            suck dust and come down to room D and go to step 7
Step 6: Else if room C is dirty:
            suck dust and go to room D and go to step 7
Step 7: If room D is dirty:
            suck dust and go to step 6
Step 8: Goal state is achieved, all ~~go~~ rooms are clean
Step 9: End
```

25/8/25

Code:

```
print(“Shreya Raj 1BM23CS317”)
def vacuum_cleaner():
    # Input the state of rooms A and B
    state_A = int(input("Enter state of A (0 for clean, 1 for dirty): "))
    state_B = int(input("Enter state of B (0 for clean, 1 for dirty): "))
    location = input("Enter location (A or B): ").upper()

    cost = 0
    rooms = {'A': state_A, 'B': state_B}

    # Function to clean a room if dirty
    def clean_room(room):
        nonlocal cost
        if rooms[room] == 1:
            print(f"Cleaned {room}.")
            rooms[room] = 0
            cost += 1
        else:
            print(f"{room} is clean.")

    # Start cleaning based on location
    if location == 'A':
        clean_room('A')
        print("Moving vacuum right")
        clean_room('B')
    elif location == 'B':
        clean_room('B')
        print("Moving vacuum left")
        clean_room('A')
    else:
        print("Invalid starting location!")

    print(f"Cost: {cost}")
    print(rooms)

if __name__ == "__main__":
    vacuum_cleaner()
```

Output:



```
Shreya Raj 1BM23CS317
Enter state of A (0 for clean, 1 for dirty): 1
Enter state of B (0 for clean, 1 for dirty): 1
Enter location (A or B): A
Cleaned A.
Moving vacuum right
Cleaned B.
Cost: 2
{'A': 0, 'B': 0}
```

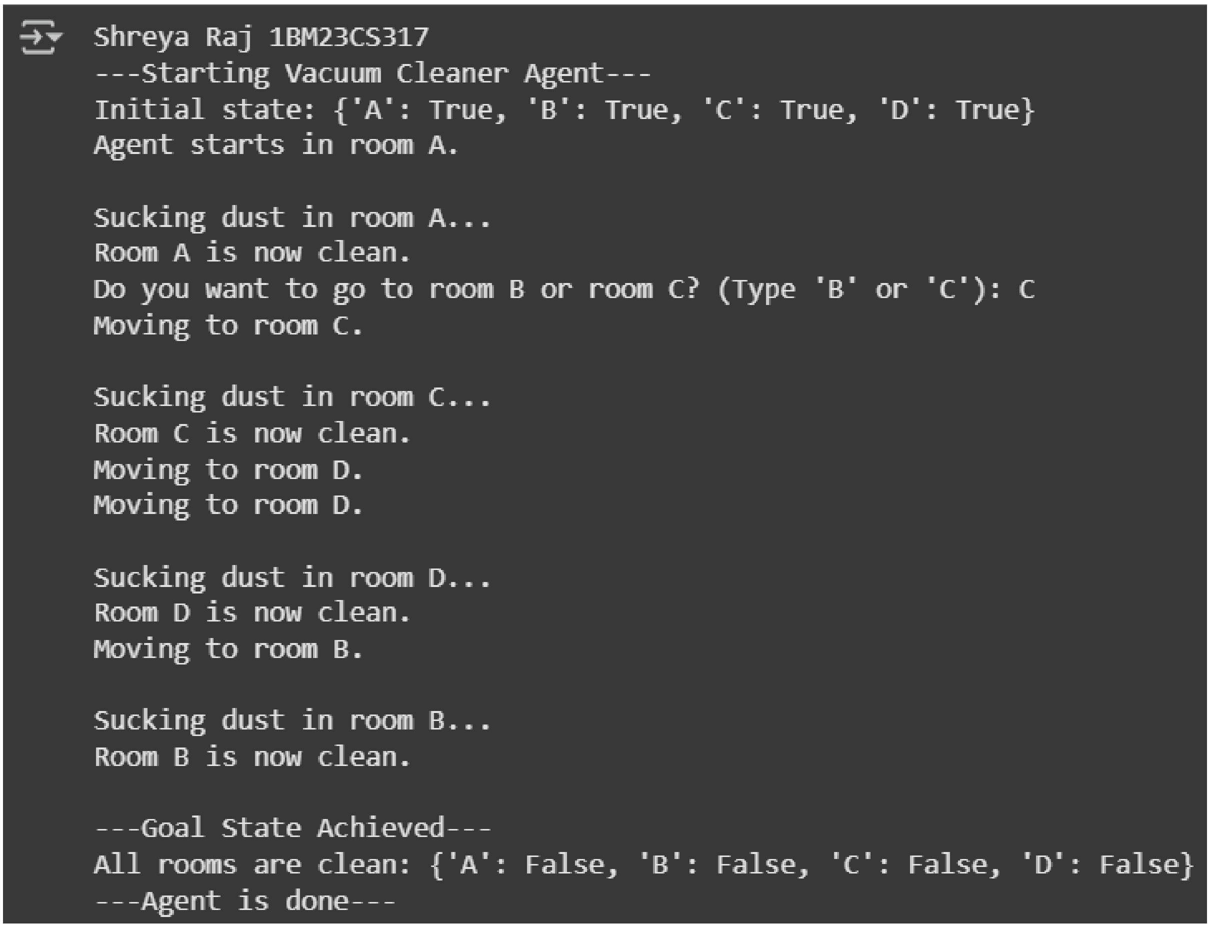
Implement vaccum cleaner agent with 4 room setup.

Code:

```
print(“Shreya Raj 1BM23CS317”)
rooms = {
    'A': True,
    'B': True,
    'C': True,
    'D': True
}

# The agent's current location
current_room = 'A'

def vacuum_cleaner_agent():

    global current_room

    print("---Starting Vacuum Cleaner Agent---")
    print("Initial state:", rooms)
    print("Agent starts in room A.")

    # A set to track visited rooms to avoid loops
    visited = set()

    # While there's any dirty room left
    while any(rooms.values()):
        # Clean the current room if dirty
        if rooms[current_room]:
            print(f"\nSucking dust in room {current_room}...")
            rooms[current_room] = False
            print(f"Room {current_room} is now clean.")
        visited.add(current_room)

        # Decide where to go next based on current location and available dirty rooms
        next_room = None

        if current_room == 'A':

            options = [room for room in ['B', 'C'] if rooms[room] and room not in visited]
            if options:
                while True:
                    user_choice = input(f"Do you want to go to room {options[0]} or room {options[-1]}? (Type '{options[0]}' or '{options[-1]}'): ").upper()
                    if user_choice in options:
                        next_room = user_choice
                        break
                    else:
                        print("Invalid input. Please choose a valid dirty room.")
            else:
                # Default to B or C if no input needed
                for room in ['B', 'C']:
                    if rooms[room] and room not in visited:
                        next_room = room
                        break

        elif current_room == 'B':
            if rooms['D'] and 'D' not in visited:
                print("Moving to room D.")
                next_room = 'D'
            elif rooms['A'] and 'A' not in visited:
```

```
        next_room = 'A'

elif current_room == 'C':
    if rooms['D'] and 'D' not in visited:
        print("Moving to room D.")
        next_room = 'D'
    elif rooms['A'] and 'A' not in visited:
        next_room = 'A'

elif current_room == 'D':
    if rooms['C'] and 'C' not in visited:
        print("Moving to room C.")
        next_room = 'C'
    elif rooms['B'] and 'B' not in visited:
        next_room = 'B'

# Fallback: find any remaining dirty room not visited yet
if not next_room:
    for room in ['A', 'B', 'C', 'D']:
        if rooms[room] and room not in visited:
            next_room = room
            break

if next_room:
    print(f"Moving to room {next_room}.")
    current_room = next_room
else:
    # No dirty unvisited rooms left
    break
    print("\n---Goal State Achieved---")

    print("All rooms are clean:", rooms)

    print("---Agent is done---")


vacuum_cleaner_agent()
```

Outputs:



```
Shreya Raj 1BM23CS317
---Starting Vacuum Cleaner Agent---
Initial state: {'A': True, 'B': True, 'C': True, 'D': True}
Agent starts in room A.

Sucking dust in room A...
Room A is now clean.
Do you want to go to room B or room C? (Type 'B' or 'C'): C
Moving to room C.

Sucking dust in room C...
Room C is now clean.
Moving to room D.
Moving to room D.

Sucking dust in room D...
Room D is now clean.
Moving to room B.

Sucking dust in room B...
Room B is now clean.

---Goal State Achieved---
All rooms are clean: {'A': False, 'B': False, 'C': False, 'D': False}
---Agent is done---
```