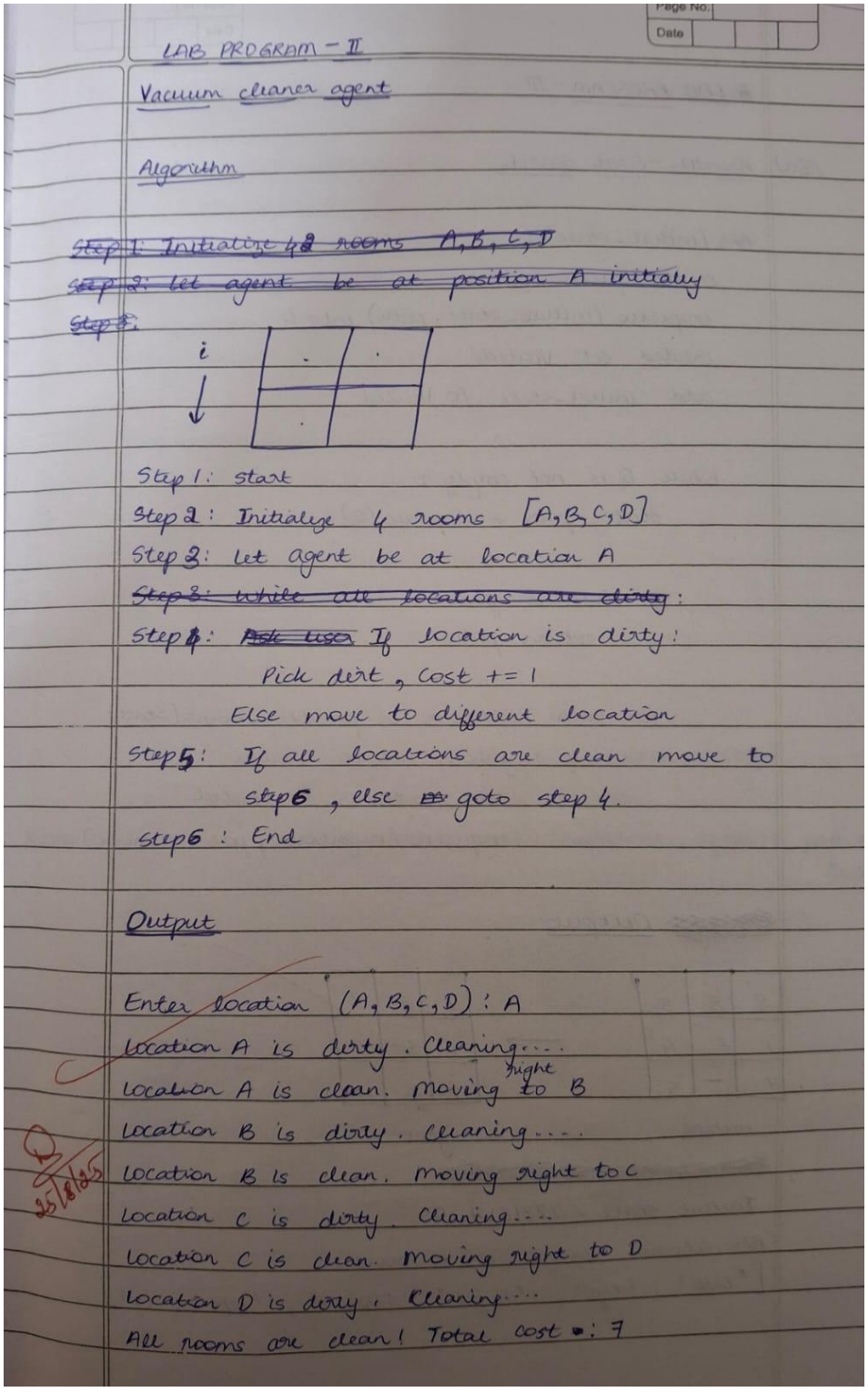
LAB PROGRAM 2:

Implement a vacuum cleaner agent

PSEUDOCODE:



LAB PROGRAM - II

Vacuum cleaner agent

Algorithm

~~Step 1: Initialize 4 rooms A, B, C, D~~

~~Step 2: let agent be at position A initialy~~

~~Step 3:~~

Step 1: start

Step 2: Initialize 4 rooms [A, B, C, D]

Step 3: let agent be at location A

~~Step 3: while all locations are dirty~~:

Step 4: ~~Ask user~~ If location is dirty:

Pick dirt, Cost += 1

Else move to different location

Step 5: If all locations are clean move to step 6, else ~~go~~ goto step 4.

Step 6: End

Output

Enter location (A, B, C, D): A

location A is dirty. Cleaning....

location A is clean. Moving right to B

location B is dirty. Cleaning....

location B is clean. Moving right to C

location C is dirty. Cleaning....

location C is clean. Moving right to D

location D is dirty. Cleaning....

All rooms are clean! Total cost: 7

25/8/25

CODE:

```
class VacuumEnvironment4Rooms:
    def __init__(self):
        self.rooms = {'A': True, 'B': True, 'C': True, 'D': True}
        self.agent_location = None

        self.room_order = ['A', 'B', 'C', 'D']

    def is_dirty(self, location):
        return self.rooms[location]

    def clean(self, location):
        self.rooms[location] = False

    def move_agent(self, location):
        self.agent_location = location

    def get_percept(self):
        return (self.agent_location,
self.is_dirty(self.agent_location))

    def all_clean(self):
        return all(not dirty for dirty in self.rooms.values())


class VacuumAgent4Rooms:
    def __init__(self, environment):
        self.env = environment
        self.room_order = environment.room_order

        self.direction = 1

    def act(self):
        location, dirty = self.env.get_percept()

        if dirty:
            print(f"Location {location} is dirty. Cleaning...")
            self.env.clean(location)
            return 'Suck'

        current_index = self.room_order.index(location)

        next_index = current_index + self.direction

        if next_index >= len(self.room_order):
            self.direction = -1
            next_index = current_index + self.direction
        elif next_index < 0:
```

```
            self.direction = 1
            next_index = current_index + self.direction

        next_location = self.room_order[next_index]
        print(f"Location {location} is clean. Moving {'right' if
self.direction == 1 else 'left'} to {next_location}...")
        self.env.move_agent(next_location)
        return 'Move'


def main():
    env = VacuumEnvironment4Rooms()

    start = input("Enter starting location (A, B, C, D):
").strip().upper()
    while start not in env.room_order:
        start = input("Invalid input. Enter starting location (A, B, C,
D): ").strip().upper()

    env.move_agent(start)

    agent = VacuumAgent4Rooms(env)

    steps = 0
    while not env.all_clean():
        agent.act()
        steps += 1

    print(f"All rooms are clean! Total steps taken: {steps}")


if __name__ == "__main__":
    main()
print("Sinchana Hemanth (1BM23CS330)")
```

OUTPUT:

```
Enter starting location (A, B, C, D): A
Location A is dirty. Cleaning...
Location A is clean. Moving right to B...
Location B is dirty. Cleaning...
Location B is clean. Moving right to C...
Location C is dirty. Cleaning...
Location C is clean. Moving right to D...
Location D is dirty. Cleaning...
All rooms are clean! Total steps taken: 7
Sinchana Hemanth (1BM23CS330)
```