Lab 2 VACCUM WORLD CLEANER (1BM22CS258)

```
class VacuumCleaner:
    def __init__(self, goal_state):
        self.goal_state = goal_state
        self.rooms = {'A': 0, 'B': 0}

    def clean(self, room):
        if self.rooms[room] == 1:
            print(f"Cleaning Location {room}...")
            self.rooms[room] = 0
            print(f"Location {room} has been Cleaned.")
            self.update_goal_state(room, 0)
        else:
            print(f"Location {room} is already clean.")

    def move(self, current_room, next_room):
        print(f"Moving to Location {next_room}.")
        self.clean(next_room)

    def update_goal_state(self, room, status):
        for i in range(0, len(self.goal_state), 2):
            if self.goal_state[i] == room:
                self.goal_state[i + 1] = status
        print(f"Updated Goal State: {self.goal_state}")
    def print_final_goal_state(self):
        print(f"Final Goal State: {self.goal_state}")


def get_room_states():
    a_state = int(input("Enter state of Room A (0 for clean, 1 for dirty): "))
    b_state = int(input("Enter state of Room B (0 for clean, 1 for dirty): "))
    return a_state, b_state
0ś
def main():
    a_state, b_state = get_room_states()
    vacuum = VacuumCleaner(goal_state=['A', a_state, 'B', b_state])
    vacuum.rooms['A'], vacuum.rooms['B'] = a_state, b_state
    vacuum.clean('A')
    vacuum.move('A', 'B')
    vacuum.print_final_goal_state()


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

OUTPUT:-

```
Enter state of Room A (0 for clean, 1 for dirty): 0
Enter state of Room B (0 for clean, 1 for dirty): 0
Location A is already clean.Moving to Location B.
Location B is already clean.
Final Goal State: ['A', 0, 'B', 0]

Enter state of Room A (0 for clean, 1 for dirty): 1
Enter state of Room B (0 for clean, 1 for dirty): 0
Cleaning Location A...
Location A has been Cleaned.
Updated Goal State: ['A', 0, 'B', 0]
Moving to Location B.
Location B is already clean.
Final Goal State: ['A', 0, 'B', 0]

Enter state of Room A (0 for clean, 1 for dirty): 0
Enter state of Room B (0 for clean, 1 for dirty): 1
Location A is already clean.
Moving to Location B.
Cleaning Location B...
Location B has been Cleaned.
Updated Goal State: ['A', 0, 'B', 0]
Final Goal State: ['A', 0, 'B', 0]


Enter state of Room A (0 for clean, 1 for dirty): 1
Enter state of Room B (0 for clean, 1 for dirty): 1
Cleaning Location A...
Location A has been Cleaned.
Updated Goal State: ['A', 0, 'B', 1]
Moving to Location B.
Cleaning Location B...
Location B has been Cleaned.
Updated Goal State: ['A', 0, 'B', 0]
Final Goal State: ['A', 0, 'B', 0]
```