Vacuum cleaner problem

```
def __init__(self)
  self.state {
vacuum_pos = input ("Enter A or B")
room_A : input("Enter clean or dirty")
room_B : input ("Enter clear or dirty")

def move_left() {
  if (self.state[vacuum_pos] == 'B')
    self.state[vacuum_pos] == 'A'

def move_right() {
  if (self.state[vacuum_pos] == 'A')
    self.state[vacuum_pos] == 'B'

def suck() {
  if (self.state[vacuum_pos] == 'A')
    if(self.state[room_A] == dirty)
      self.state[room_A] = "clean"

  else if (self.state[vacuum_pos] == 'B')
      if (self.state[room_B] == "dirty")
        self.state[room_B] = "clean"


def main()
  if (self.state[vacuum_pos] == 'A')
    if (self.state[room_A] == "dirty")
      suck()
    else
      move_right()
  else if (self.state[vacuum_pos] == 'B')
      if (self.state[room_B] == "dirty")
        else suck()
          move_left()
  else if (room_A == "clean" && room_B == "clean")
      break
```

Proved

```
class VacuumCleaner:
    def __init__(self):
        self.state = {
            "vacuum_pos": input("Enter the initial position of
            the vacuum cleaner (A or B): ").upper(),
            "room_A": input("Is Room A dirty or clean? ").lower(),
            "room_B": input("Is Room B dirty or clean? ").lower()
        }
    def show_state(self):
        print(f"Vacuum Position: {self.state['vacuum_pos']},
        Room A: {self.state['room_A']},
        Room B: {self.state['room_B']}")
    def is_clean(self):
        return self.state["room_A"] == "clean" and
        self.state["room_B"] == "clean"
    def move_right(self):
        if self.state["vacuum_pos"] == "A":
            self.state["vacuum_pos"] = "B"
            print("Moving to Room B")
    def move_left(self):
        if self.state["vacuum_pos"] == "B":
            self.state["vacuum_pos"] = "A"
            print("Moving to Room A")
    def suck(self):
        if self.state["vacuum_pos"] == "A":
            if self.state["room_A"] == "dirty":
                self.state["room_A"] = "clean"
                print("Cleaning room B")
```

```
def run(self):
  while not self.is_clean():
    self.show_state()
    if self.state["vacuum_pos"]=="A":
      if self.state["room_A"]=="dirty":
        self.suck()
      else:
        self.move_right()
    elif self.state["vacuum_pos"]=="B":
      if self.state["room_B"]=="dirty":
        self.suck()
      else:
        self.move_left()
  print("Both rooms are clean now")
  self.show_state()
vacuum: VacuumCleaner()
vacuum.run()
```

Output:

```
Enter initial position of the vacuum cleaner (AorB): A
Is Room A dirty or clean? clean
Is Room B dirty or clean? dirty
VacuumPosition: A, RoomA: clean, RoomB: dirty
Moving to Room B
VacuumPosition: B RoomA: clean, RoomB: dirty
Cleaning Room B
Both rooms are clean now!
Vacuum position: B, RoomA: clean. RoomB: clean
```

7/10/24

Output for four rooms:

step 1
Vacuum is in room A, Room state: clean
Action: move down
moved down to Room C
Room states: {'A': 'clean', 'B': 'clean', 'C': 'Dirty',
'D': Dirty}

step 2
Vacuum is in room C, Room state: Dirty
Action: suck
sucking dirt in room C
Room states: {'A': 'clean', 'B': 'clean', 'C': 'clean',
'D': 'Dirty'}

step 3:
Vacuum is in room C, Room state: clean
Action: move right
moved right to room D
Room states: {'A': 'clean', 'B': 'clean', 'C': 'clean',
'D': 'Dirty'}

step 4:
Vacuum is in room D, Room state: Dirty
Action: suck
sucking dirt in room D
Room states: {'A': 'clean', 'B': 'clean', 'C': 'clean',
'D': clean}

Program

```
self.rooms = {
    'A': 'clean',
    'B': 'clean',
    'C': 'Dirty',
    'D': 'Dirty',
}

self.neighbors = {
    'A': {'right': 'B', 'down': 'C'},
    'B': {'left': 'A', 'down': 'D'},
    'C': {'up': 'A', 'right': 'D'},
    'D': {'up': 'B', 'left': 'C'}
}
```

11/12/24

```
Enter the initial position of the vacuum cleaner (A or B): A
Is Room A dirty or clean? dirty
Is Room B dirty or clean? dirty
Vacuum Position: A, Room A: dirty, Room B: dirty
Cleaning Room A
Vacuum Position: A, Room A: clean, Room B: dirty
Moving to Room B
Vacuum Position: B, Room A: clean, Room B: dirty
Cleaning Room B
Both rooms are clean now!
Vacuum Position: B, Room A: clean, Room B: clean
```