Vacuum cleaner problem

```
def_innit_(self) -
  self.state {
vacuum_pos = input ("Enter A or B ")
room_A = input("Enter clean or dirty ")
room_B = input ("Enter clean 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"
       suck()
     else
        move_left()
  else if (room_A == "clean" && room_B == "clean")
    break
```

```
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["vacum_pos"]=="A":
       if self.state["room_A"]=="dirty":
          self.suck()
      else:
          self.move_right()
    elif self.state["vacum_pos"]=="B":
      if self.state["room_B"]=="dirty":
          self.suck()
      else:
          self.move_left()
    print("Both rooms are clean now")
    self.show_state()
vacum= VacumCleaner()
vacum.run()
```

Output:

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

1/10/24

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'}
```