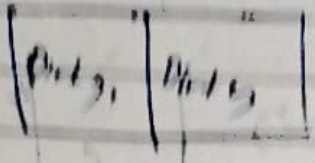
Vacuum Cleaner Problem

Agent + 2 rooms

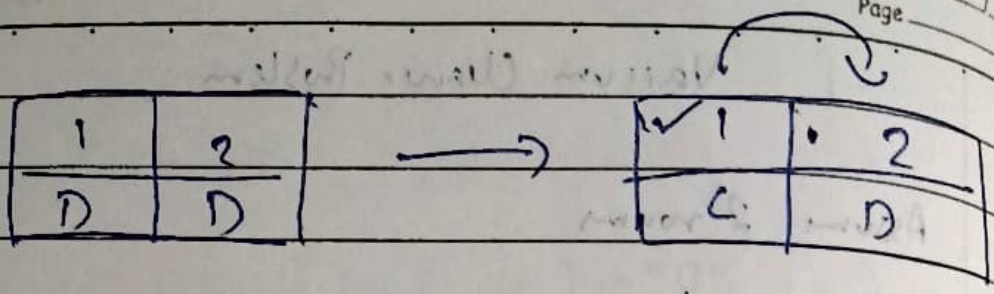


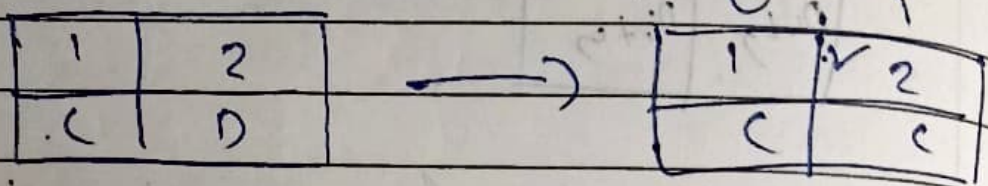
Initialize two states for a room being location & status. location is the room number (either 1 or 2) let status be either clean or dirty.

```
clean_room (room)
{ room_status = ["Dirty", "clean"]
  room.1 = "Dirty"
  room 2 = "Dirty"
}
```

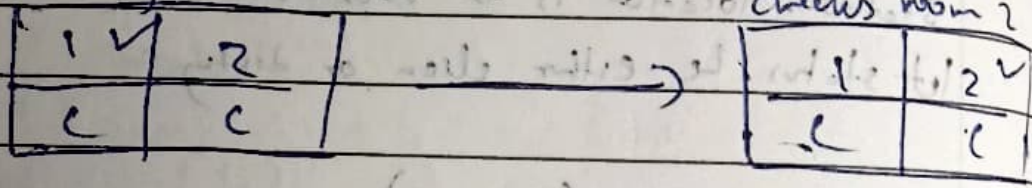
```
Vacuum cleaner (room)
{
  initial room = room1
    if (room.1 = "Dirty")
        clean the room. Set state to clean
  [Since room is clean it goes left]
    if (room.2 = "Dirty")
        Clean the room. set status to clean
  [Since room is clean, it goes right]
    if (room1 = "clean"):
        it goes left as state remains same
    if (room2 = "clean")
        it turns clean off and state remains same.
```

Date __/__/__
Page

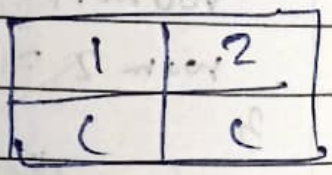




clock goes to room 2.



checks room 2



(1, dirty)
(1, clean) → (2, dirty)
(1, clean) (2, clean) → (1, clean) [stops]

Date __/__/__
Page _____

```
class Vaccum_Cleaner:
    def __init__(self, environment):
        self.environment = environment
        self.cleaned_cells = 0
        self.position = (0,0):

def clean(self):
    while True:
        x,y = self.position

        if self.environment[x][y] == 'D'
            self.environment[x][y] == 'D':
            self.cleaned_cell += 1
            print("cleaned position {self.position}")

        next_position = self.find_next_dirty()
        if next_position:
            print("Moving to next dirty position {next_position}")
            self.position = next_position
        else:
            print("No dirty room. Cleaning complete")
            break

Def find_next_dirty(self):

    for i in range(len(self.environment)):
        for j in range(len(self.environment[i])):
            if self.environment[i][j] == 'D'
                return(i,j)

    return None
```

```
def display_environment(self):
    for row in self.environment:
        print(" ".join(row))
    print(f"\nTotal cleaned cells: {self.clean...

Initial_environment = [
    ['D', 'D']
]

agent = VacuumCleanerAgent(initial_envir...
print("Initial Environment")
agent.display_environment()
agent.clean()
print("Final Environment")
agent.display_environment()
```

Output:

```
Initial Environment
D D
total cleaned steps: 0
Cleaned position (0,0)
Moving to next dirty position (0,1)
cleaned position (0,1)
No more dirty cells

final Environment
C C
total cleaned cells: 2
```

Date __/__/__
Page _____

Initial Environment

D D
D D

total cleaned cells : 0
cleaned position (0, 0)
moving to next dirty position (0, 1)
cleaned position (0, 1)
Moving to next dirty position (1, 0)
cleaned position (1, 1)
no more dirty rooms

Final Environment

C C
C C