

1/ Oct 19

## Lab-2

A = Right

B = right



### Algorithm ;

→ S-1 : Create Two rooms Using 2D list  
1 - dirt      0 - clean  
[0, 0, 1] etc.

→ S-2 : Start With Any State for initial State  
Can

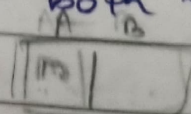
→ S-3 : if current Position has dirt (1)  
clean it by setting it to 0

if room1 [position [0]] [position [1]] == 1:  
room1 [position [0]] [position [1]] = 0 ;  
Print ("Cleaned.")

→ S-4 → move vacuum cleaner to next state  
When one room is done  
(if room A is dirt, suck it and move room B)

→ S-5 → keep checking for dirt & cleaning  
until all dirt is removed

→ S-6 → if both rooms are clean Exit the  
loop & Print that both rooms are clean



Percept Sequence

[A, clean]

[A, dirty]

right

[B, clean]

suck

[B, dirty]

left

[A, clean] [A, clean]

suck

[A, clean] [A, dirty]

left

[B, clean] [B, clean]

suck

[B, clean] [B, dirty]

right

[A, clean] [B, clean]

suck

suck

~~Done~~Program

class room:

def \_\_init\_\_(self, a):

self.state = a

def suck(self):

self.state = "clean"

n = 2

roomList = []

for i in range(n):

a = str(input(f"enter room {i+1} state: "))

roomList.append(room(a))

start = int(input("enter starting room number: "))

Print ("Before cleaning")

Print ("room state")

for i in range(len(roomList)):

Print (f"{i+1} {roomList[i].state}")



```

Count = 0
while Count < len (room list):
    if (room list [start].state.lower () == "dirty")
        room list [start].state = "clean"
        start = (start + 1) % len (room list)
        Count += 1

```

```

Print ("In")
Print ("After cleaning")
Print ("Room list state")
for i in range (len (room list)):
    Print (f" {i+1} {room list [i].state}")

```

Output:-

Enter room 1 state: dirty  
 Enter room 2 state: clean  
 Enter starting room number: 1  
 Before cleaning  
 Room state

1. dirty
2. clean

After cleaning  
 Room state

1. clean
2. clean

4 rooms

$n = 4$

roomList = []

for i in range(n):

a = str(input(f"Enter room {i+1} state  
(dirty/clean):"))

roomList.append(room(a))

start = int(input("Enter starting room number  
(1 to 4):")) - 1

Print ("Before cleaning")

Print ("room \ state")

for i in range(len(roomList)):

Print (f"{i+1} \ {room[i].state}")

count = 0

Output:

Enter room 1 state (dirty/clean): dirty

Enter room 2 state (dirty/clean): clean

Enter room 3 state (dirty/clean): dirty

Enter room 4 state (dirty/clean): clean

Enter starting room number (1 to 4): 1

Before cleaning

Room state:

1. dirty

2. clean

3. dirty

4. clean

after cleaning

1. clean

2. clean

11/10/24