(2) Vacuum cleaner Algorithm

(i) A vacuum cleaner (agent) performs a suck operation if the room is dirty and moves to next location and checks the status.

(ii) We consider 2 rooms A and B (a grid 2×1), the agent can percieve whether its current room is clean or dirty.

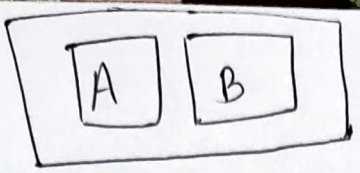
(iii) Thus the agent performs 3 operations : move left, move right, suck.

(iv) Input : Bodean data (0 or 1) whether the current room is clean or dirty . Clean = 1 , Dirty = 0

(v)
```
while (environment != clean)

    if (current room = dirty){
        clean();
    }
    else if (current_room = clean)
    {
    }

if (agent in room A)
  {   move_right();
  }
else if (agent in room B)
  {  move_left();
  }
}
```



DATE: PAGE:

Percept Sequence:	Action
[A, clean]	move right
[A, Dirty]	suck operation
[B, clean]	move left
[B, Dirty]	suck operation
[A, clean] [B, Dirty]	suck operation
[A, clean] [B, clean]	move left
[B, clean] [A, clean]	move right
[B, clean] [A, dirty]	suck operation
[A, clean] [B, Dirty] [B, clean]	move left
[A, clean] [B, clean] [A, ~~dirty~~ clean]	terminate operation ~~suck operation~~

Code:

```
import random

class VacuumCleaner:
    def __init__(self):
        self.rooms = {'A': 'dirty', 'B': 'dirty', 'C': 'dirty',
                      'D': 'dirty'}
        self.position = 'A'
        self.room_order = ['A', 'B', 'C', 'D']
        self.current_index = 0

    def status(self):
        print(f"\nCurrent Position: {self.position}")
        for room, state in self.rooms.items():
            print(f"Room {room}: {state}")

    def suck(self):
        if self.rooms[self.position] == 'dirty':
            self.rooms[self.position] = 'clean'
```

DATE: PAGE:

```
        print(f"Sucking in room {self.position}. Room{self.position}
            is now clean.")
    else:
        print(f"Room{self.position} is already clean.")


def move_clockwise(self):
    self.current_index = (self.current_index+1) % len(self.ro
    self.position = self.room_order[self.current_index]
    print(f"Moved clockwise to room{self.position}.")

def move_counterclockwise(self):
    self.current_index = (self.current_index-1) %
                                    len(self.room_order)
    self.position = self.room_order[self.current_index]
    print(f"Moved counterclockwise to room{self.position}.")

def run(self):
    while 'dirty' in self.rooms.values():
        self.status()
        if self.rooms[self.position] == 'dirty':
            self.suck()
        else:
            if self.position == 'D':
                self.move_counterclockwise()
            else:
                self.move_clockwise()

    print("\nAll rooms are clean!")
```

DATE: PAGE:

```
if __name__ == "__main__":
    vacuum = VacuumCleaner()
    vacuum.run()
```

Output:

Current position : A
Room A : dirty
Room B : dirty
Room C : dirty
Room D : dirty

Sucking in room A. Room A is now clean.

Current position : A
Room A : clean
Room B : dirty
Room C : dirty
Room D : dirty
Moved clockwise to room B.

Current position : B
Room A : clean
Room B : dirty
Room C : dirty
Room D : dirty

Sucking in room B. Room B is now clean

Current position : B
Room A : clean
Room B : clean

DATE:

PAGE:

Room C : dirty
Room D : dirty
Moved clockwise to room C.

Current Position : C
Room A : clean
Room B : clean
Room C : dirty
Room D : dirty

Sucking in room C . Room C is now clean

Current Position : C
Room A : clean
Room B : clean
Room C : clean
Room D : dirty

Move clockwise to room D

Current Position : D
Room A : clean
Room B : clean
Room C : clean
Room D : dirty

Sucking in room D . Room D is now clean

All rooms are clean!