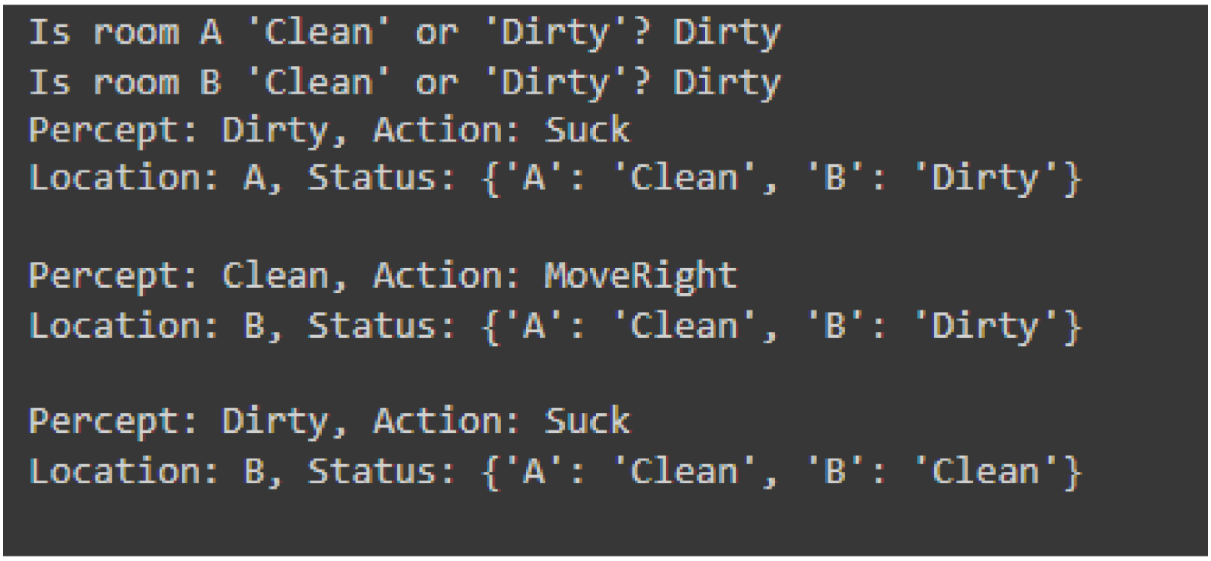
Sanghamitra R
1BM22CS237

AI Lab 2: Vacuum World

```
# Agent function table
agent_table = {
    ('Clean', 'A'): 'MoveRight',
    ('Clean', 'B'): 'MoveRight',
    ('Clean', 'C'): 'MoveRight',
    ('Clean', 'D'): 'MoveLeft',
    ('Dirty', 'A'): 'Suck',
    ('Dirty', 'B'): 'Suck',
    ('Dirty', 'C'): 'Suck',
    ('Dirty', 'D'): 'Suck',
}

# Vacuum cleaner class
class VacuumCleaner:
    def __init__(self, location='A', status_a='Clean',
status_b='Clean', status_c='Clean', status_d='Clean'):
        self.location = location
        self.status = {'A': status_a, 'B': status_b, 'C': status_c,
'D': status_d}

    def percept(self):
        return self.status[self.location]

    def act(self, action):
        if action == 'MoveRight':
            if self.location == 'A':
                self.location = 'B'
            elif self.location == 'B':
                self.location = 'C'
            elif self.location == 'C':
                self.location = 'D'
        elif action == 'MoveLeft':
            if self.location == 'B':
                self.location = 'A'
            elif self.location == 'C':
                self.location = 'B'
            elif self.location == 'D':
                self.location = 'C'
        elif action == 'Suck':
            self.status[self.location] = 'Clean'

    def all_clean(self):
        return all(status == 'Clean' for status in
self.status.values())

# Table-driven agent function
def table_driven_agent(percept):
    return agent_table.get(percept, 'NoOp')

# Main simulation loop
if __name__ == "__main__":
    status_a = input("Is room A 'Clean' or 'Dirty'?
").strip().capitalize()
    status_b = input("Is room B 'Clean' or 'Dirty'?
").strip().capitalize()
    status_c = input("Is room C 'Clean' or 'Dirty'?
").strip().capitalize()
    status_d = input("Is room D 'Clean' or 'Dirty'?
").strip().capitalize()
    vacuum = VacuumCleaner(status_a=status_a, status_b=status_b,
status_c=status_c, status_d=status_d)

    while not vacuum.all_clean():  # Run until all rooms are clean
        current_percept = vacuum.percept()
        action = table_driven_agent((current_percept, vacuum.location))
        print(f"Percept: {current_percept}, Action: {action}")

        if action != 'NoOp':
            vacuum.act(action)

        print(f"Location: {vacuum.location}, Status:
{vacuum.status}\n")

    print("All rooms are clean!")
```

Output:



```
Is room A 'Clean' or 'Dirty'? Dirty
Is room B 'Clean' or 'Dirty'? Dirty
Percept: Dirty, Action: Suck
Location: A, Status: {'A': 'Clean', 'B': 'Dirty'}

Percept: Clean, Action: MoveRight
Location: B, Status: {'A': 'Clean', 'B': 'Dirty'}

Percept: Dirty, Action: Suck
Location: B, Status: {'A': 'Clean', 'B': 'Clean'}
```