VACCUM CLEANER AGENT

NAME: VENKATESH VINAY CHANDLE USN: 1BM22CS325

INPUT:

```
import random


class VacuumCleanerEnvironment:
    def __init__(self):
        self.grid = [random.choice([0, 1]), random.choice([0, 1])]
        self.agent_position = 0

    def is_dirty(self):
        return self.grid[self.agent_position] == 1

    def clean(self):
        if self.is_dirty():
            print(f"Cleaning location {'A' if self.agent_position == 0 else 'B'}")
            self.grid[self.agent_position] = 0

    def display(self):
        print(f"Current grid: A: {'Dirty' if self.grid[0] == 1 else 'Clean'}, B: {'Dirty' if self.grid[1] == 1 else 'Clean'}")

    def move(self):
        self.agent_position = 1 - self.agent_position
        print(f"Agent moved to location {'A' if self.agent_position == 0 else 'B'}")

    def is_done(self):
        return all(cell == 0 for cell in self.grid)

class VacuumCleanerAgent:
    def __init__(self, environment):
```

```
        self.environment = environment

    def run(self):
        while not self.environment.is_done():
            print(f"Agent is at location {'A' if self.environment.agent_position == 0 else 'B'}")
            self.environment.display()

            if self.environment.is_dirty():
                self.environment.clean()

            if not self.environment.is_done():
                self.environment.move()

env = VacuumCleanerEnvironment()
agent = VacuumCleanerAgent(env)
print("Venkatesh Vinay Chandle, 1BM22CS325")
print("Starting Vacuum Cleaner Agent Simulation...")
agent.run()
print("\nFinal State of the Grid:")
env.display()
```

OUTPUT:

```
Venkatesh Vinay Chandle, 1BM22CS325
Starting Vacuum Cleaner Agent Simulation...
Agent is at location A
Current grid: A: Clean, B: Dirty
Agent moved to location B
Agent is at location B
Current grid: A: Clean, B: Dirty
Cleaning location B

Final State of the Grid:
Current grid: A: Clean, B: Clean
```

Lab-02

2. Implement Vaccum Cleaner Agent

Pseudocode

```
Function vaccum_world(){
    initialize goal_state = {'A':'0','B':'0'}
    initialize cost = 0
    Input location
    Input status for location
    Input status for other location
    Print Initial condition for Location, goal state
If location input = 'A' and status_input = '1' then, {
    print location A is dirty
    goal_state['A'] = '0'
    cost = 1
    print cost for cleaning 'A' - cost
If status for other location = '1' then {
    print location B is Dirty
    cost = 1
    print cost for moving right as cost }
else
    print "Vaccum is placed in Location B"
    If status_input == 1 then Location B is dirty
else {
    print Location A is already clean
    If status of other location = 1 then {
        print Location B is dirty
        cost = 1
        Print cost for moving right = cost
        cost = 1
        Print total cost of cleaning, cost
} }
Else { print vaccum is at Location B }
    If status == '1' then print 'Location B is Dirty'
    cost = 1
    cost for cleaning, cost
    If status of other location = 1, then {
        print Location = 1 then,
        cost = 1
        print cost for moving Left, cost
        goal state['A'] = '0'
        cost = 1
        print cost for cleaning, cost }
else {
    print location B is already clean
    If status other location = '1' then {
        print Location A is dirty
        cost = 1
        print cost for moving Left. cost
        goal state ['A'] = '0'
        cost = 1
        print cost for clean, cost
    }
    print performance Measurement cost.
}
```

Output

Locations: A-0 B-1

Enter Location of vaccum: 1 (at B)

Enter status of Room (0 for clean, 1 for dirty): 1

Enter status of other room (0 for clean, 1 for dirty): 0

Initial Location condition

Vaccum is placed in B

Location B is Dirty

Cost for cleaning: 1