1. Vacuum

import random

# Define environment

rooms = {

'A': random.choice(['Clean', 'Dirty']),

'B': random.choice(['Clean', 'Dirty']),

'C': random.choice(['Clean', 'Dirty']),

'D': random.choice(['Clean', 'Dirty'])

}

agent\_position = random.choice(['A', 'B', 'C', 'D'])

def display\_state():

print(f"Agent is in Room {agent\_position}")

for room, status in rooms.items():

print(f"Room {room}: {status}")

print()

# Rule-based agent logic with smart movement

def vacuum\_agent():

global agent\_position

steps = 0

# Keep track of cleaned rooms

cleaned\_rooms = set()

while len(cleaned\_rooms) < 4:

display\_state()

# Clean current room if dirty

if rooms[agent\_position] == 'Dirty':

print(f"🧹 Cleaning Room {agent\_position}")

rooms[agent\_position] = 'Clean'

cleaned\_rooms.add(agent\_position)

else:

print(f"✅ Room {agent\_position} is already clean.")

cleaned\_rooms.add(agent\_position)

# Decide next move only if not all rooms are clean

if len(cleaned\_rooms) < 4:

# Move to the next room that is still dirty

for next\_room in ['A', 'B', 'C', 'D']:

if next\_room not in cleaned\_rooms:

agent\_position = next\_room

break

steps += 1

print(f"Step {steps} complete.\n")

print("🎉 All rooms are clean!")

display\_state()

vacuum\_agent()output:

