

## EXPERIMENT 5: Vacuum cleaner problem

Aim: Implement an Algorithm in Python to solve the vacuum cleaner problem.

Code:

```
class VacuumCleaner:
    def __init__(self):
        self.position = 0
        self.environment = ['Clean', 'Dirty', 'Clean', 'Dirty', 'Clean'] # Example environment

    def sense(self):
        return self.environment[self.position]

    def clean(self):
        self.environment[self.position] = 'Clean'

    def move(self):
        if self.position < len(self.environment) - 1:
            self.position += 1

    def run(self):
        while True:
            status = self.sense()
            if status == 'Dirty':
                print("Vacuum cleaner is cleaning.")
                self.clean()
            else:
                print("No dirt found.")
                self.move()
            print("Moving to the next position.")
            print("Environment status:", self.environment)
            print("Current position:", self.position)
            print()

            # Check if all positions are clean
            if all(pos == 'Clean' for pos in self.environment):
                print("All positions are clean. Task completed!")
                break
```

```
# Example usage:  
vacuum = VacuumCleaner()  
vacuum.run()
```

Output:

No dirt found.

Moving to the next position.

Environment status: ['Clean', 'Dirty', 'Clean', 'Dirty', 'Clean']

Current position: 1

Vacuum cleaner is cleaning.

Moving to the next position.

Environment status: ['Clean', 'Clean', 'Clean', 'Dirty', 'Clean']

Current position: 2

No dirt found.

Moving to the next position.

Environment status: ['Clean', 'Clean', 'Clean', 'Dirty', 'Clean']

Current position: 3

Vacuum cleaner is cleaning.

Moving to the next position.

Environment status: ['Clean', 'Clean', 'Clean', 'Clean', 'Clean']

Current position: 4

All positions are clean. Task completed!

Result: Code has been Implemented successfully.