

Program 3

Assume that there are 3 floors and 4 rooms in each floor. Design the vacuum cleaner to ensure the rooms are clean. You may make suitable assumption for initial state.

Program:

```
# Given M x N grid(floor) create an agent that moves around the grid until the entire grid is clean
```

```
floor = [[1, 0, 0, 0], # '1' represents dirty and '0' represents clean  
         [0, 1, 0, 1],  
         [1, 0, 1, 1]]
```

```
def clean(floor):  
    m = len(floor[0]) # no of cols  
    n = len(floor) # no of rows  
    no_of_tiles = m * n
```

```
    tiles_checked = 0
```

```
    row = 0  
    col = 0
```

```
    while tiles_checked < no_of_tiles:  
        # Current position  
        print_floor(floor, row, col)
```

```
        # Suck if dirty
```

```
        if floor[row][col] == 1:  
            floor[row][col] = 0  
            print('Sucked the dirt')
```

```
        else:  
            print('Already Clean')
```

```
        # Next tile
```

```
        if row % 2 == 0: # Even rows the bot moves right to the next tile  
            if col < m-1:  
                col += 1
```

```
            else:  
                row += 1 # Move to next row if we reached the last col
```

```
        elif row % 2 == 1: # Odd rows the bot moves left to the next tile  
            if 0 < col:  
                col -= 1
```

```
            else:
```

```
                row += 1 # Move to next row if we reached the last col
```

```
    tiles_checked += 1
```

```
    print('-----')
```

```
print('Cleaned!!!')
```

Output

```
['VC', 0, 0, 0]
[0, 1, 0, 1]
[1, 0, 1, 1]
Sucked the dirt
-----
[0, 'VC', 0, 0]
[0, 1, 0, 1]
[1, 0, 1, 1]
Already Clean
-----
[0, 0, 'VC', 0]
[0, 1, 0, 1]
[1, 0, 1, 1]
Already Clean
-----
[0, 0, 0, 'VC']
[0, 1, 0, 1]
[1, 0, 1, 1]
Already Clean
-----
[0, 0, 0, 0]
[0, 1, 0, 'VC']
[1, 0, 1, 1]
Sucked the dirt
-----
[0, 0, 0, 0]
[0, 1, 'VC', 0]
[1, 0, 1, 1]
Already Clean
-----
[0, 0, 0, 0]
[0, 'VC', 0, 0]
[1, 0, 1, 1]
Sucked the dirt
-----
[0, 0, 0, 0]
['VC', 0, 0, 0]
[1, 0, 1, 1]
Already Clean
-----
[0, 0, 0, 0]
[0, 0, 0, 0]
['VC', 0, 1, 1]
Sucked the dirt
-----
[0, 0, 0, 0]
[0, 0, 0, 0]
[0, 'VC', 1, 1]
Already Clean
-----
[0, 0, 0, 0]
[0, 0, 0, 0]
[0, 0, 'VC', 1]
Sucked the dirt
```

```
-----  
[0, 0, 0, 0]  
[0, 0, 0, 0]  
[0, 0, 'VC', 1]  
Sucked the dirt  
-----  
[0, 0, 0, 0]  
[0, 0, 0, 0]  
[0, 0, 0, 'VC']  
Sucked the dirt  
-----  
Cleaned!!!
```