

PROGRAM TITLE-6

VACCUM CLEANER

AIM:

To write and execute the python program for Vacuum cleaner.

PROCEDURE:

1. Initialize the environment: Create a grid environment with clean and dirty cells. Initialize the vacuum cleaner's position randomly.
2. Clean the environment:

While there are still dirty cells in the environment:

Check if the current cell is dirty. If yes, clean it.

Move the vacuum cleaner to a neighbouring cell. Choose the next cell based on a predefined strategy (e.g., random, systematic, or intelligent movement).
3. Random movement: Move the vacuum cleaner randomly to any neighbouring cell.
4. Systematic movement: Move the vacuum cleaner in a systematic way (e.g., row by row or column by column) until all cells are cleaned.
5. Intelligent movement: Use sensors or heuristics to detect and navigate towards dirty cells efficiently.
6. Implement Simple Reflex Agent:
 - Design a simple reflex agent that makes decisions based on the current percept (sensory input). For example, the agent could move to the left if there's dirt on the left, move to the right if there's dirt on the right, and clean the current location if it's dirty.
7. Define Environment Dynamics:
 - Implement the dynamics of the environment, including the generation of dirt and the vacuum cleaner's movement. Define the conditions under which dirt appears and how the vacuum cleaner's position changes.
8. Simulate the Environment:
 - Run a simulation of the environment, allowing the vacuum cleaner to make decisions based on its percept and perform actions accordingly. Print or visualize the state of the environment at each step, including the location of dirt and the vacuum cleaner's position.

CODING:

```
import random
```

```
def display(room):
```

```
    print(room)
```

```
room = [
```

```
    [1, 1, 1, 1],
```

```
    [1, 1, 1, 1],
```

```
    [1, 1, 1, 1],
```

```
    [1, 1, 1, 1],
```

```
]
```

```
print("All the rooom are dirty")
```

```
display(room)
```

```
x =0
```

```
y= 0
```

```
while x < 4:
```

```
    while y < 4:
```

```
        room[x][y] = random.choice([0,1])
```

```
        y+=1
```

```
    x+=1
```

```
    y=0
```

```
print("Before cleaning the room I detect all of these random dirts")
```

```
display(room)
```

```
x =0
```

```
y= 0
```

```
z=0
```

```
while x < 4:
```

```
    while y < 4:
```

```

if room[x][y] == 1:

    print("Vaccum in this location now,"x, y)

    room[x][y] = 0

    print("cleaned", x, y)

    z+=1

    y+=1

    x+=1

    y=0

pro= (100-((z/16)*100))

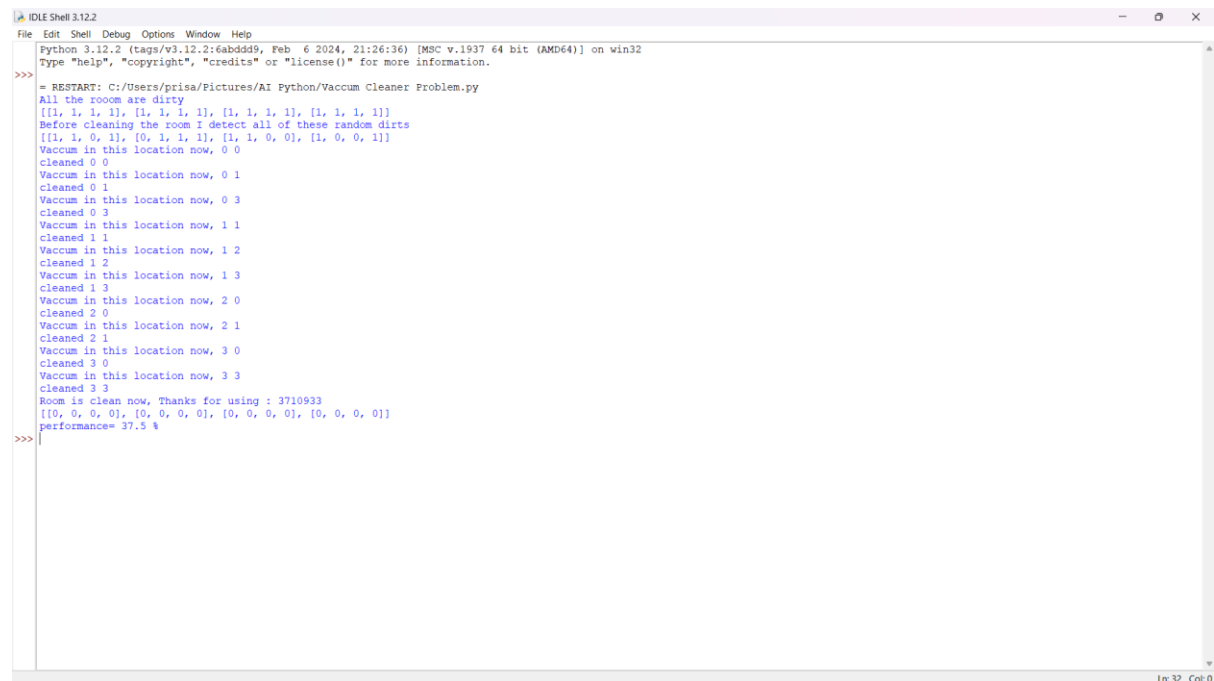
print("Room is clean now, Thanks for using : 3710933")

display(room)

print('performance=',pro,'%')

```

OUTPUT:



```

Python 3.12.2 (tags/v3.12.2:6abddd9, Feb  6 2024, 21:26:36) [MSC v.1937 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:/Users/prisa/Pictures/AI Python/Vaccum Cleaner Problem.py
All the room are dirty
[[1, 1, 1, 1], [1, 1, 1, 1], [1, 1, 1, 1], [1, 1, 1, 1]]
Before cleaning the room I detect all of these random dirt
[[1, 1, 0, 1], [0, 1, 1, 1], [1, 1, 0, 0], [1, 0, 0, 1]]
Vaccum in this location now, 0 0
cleaned 0 0
Vaccum in this location now, 0 1
cleaned 0 1
Vaccum in this location now, 0 3
cleaned 0 3
Vaccum in this location now, 1 1
cleaned 1 1
Vaccum in this location now, 1 2
cleaned 1 2
Vaccum in this location now, 1 3
cleaned 1 3
Vaccum in this location now, 2 0
cleaned 2 0
Vaccum in this location now, 2 1
cleaned 2 1
Vaccum in this location now, 3 0
cleaned 3 0
Vaccum in this location now, 3 3
cleaned 3 3
Room is clean now, Thanks for using : 3710933
[[0, 0, 0, 0], [0, 0, 0, 0], [0, 0, 0, 0], [0, 0, 0, 0]]
performance= 37.5 %
>>>

```

RESULT:

Thus the result has been successfully executed and verified.