Date: **Ex No:** 1.1	**Title of the Lab** Vacuum Cleaner World	**Name:** Yuvraj Singh Chauhan **Registration Number:** RA1911027010058 **Section:** N1 **Lab Batch:** 1 **Day Order:** 3

AIM:

To implement a vacuum cleaner world problem which returns a sequence of actions that leads to the goal state, along with the path cost.

Description of the Concept or Problem given:

Developed a simple reflex agent program in Python for the vacuum-cleaner world problem. This program defines the States, Goal State, Goal Test, Actions, Transition Model, and Path Cost. For each possible initial state, the program returns a sequence of actions that leads to the goal state, along with the path cost.

Manual Solution:

Location need to be entered as A/B (in capitals) where A and B are adjacent rooms respectively.

Status need to be entered as 0/1 where 0 means clean while 1 means dirty.

The location first mentioned will be the place where the vacuum cleaner will be deployed. For every cleaning and moving cost will be incremented.

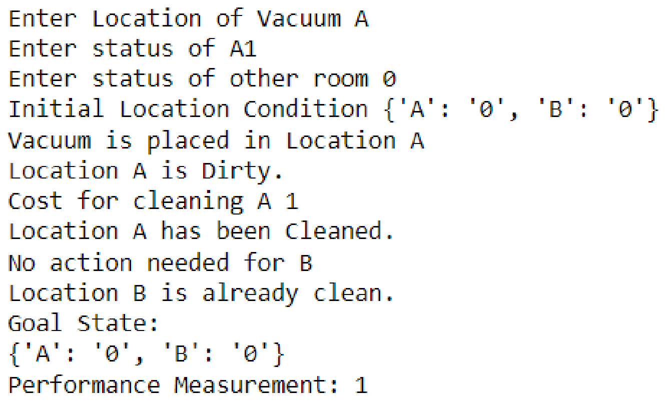
Program Implementation [Coding]

```
def vacuum_cleaner():

    goal = {'A': '0', 'B': '0'}

    cost = 0
```

```
location_input = input("Enter Location of Vacuum ")
status_input = input("Enter status of " + location_input )
status_input_complement = input("Enter status of other room ")
print("Initial Location Condition " + str(goal))

if location_input == 'A':
    print("Vacuum is placed in Location A ")
    if status_input == '1':
        print("Location A is Dirty.")
        goal['A'] = '0'
        cost += 1
        print("Cost for cleaning A " + str(cost))
        print("Location A has been Cleaned.")

        if status_input_complement == '1':
            print("Location B is Dirty.")
            print("Moving right to the Location B. ")
            cost += 1
            print("Cost for moving right " + str(cost))
            goal['B'] = '0'
            cost += 1
            print("Cost for sucking B " + str(cost))
            print("Location B has been Cleaned. ")
        else:
            print("No action needed for B ")
            print("Location B is already clean.")

    if status_input == '0':
        print("Location A is already clean ")
        if status_input_complement == '1':
            print("Location B is Dirty.")
            print("Moving right to the Location B. ")
            cost += 1
            print("Cost for moving right " + str(cost))
            goal['B'] = '0'
            cost += 1
            print("Cost for sucking B " + str(cost))
            print("Location B has been Cleaned. ")
        else:
            print("No action needed for B")
            print(cost)
            print("Location B is already clean.")

else:
    print("Vacuum is placed in location B")
    if status_input == '1':
        print("Location B is Dirty.")
        goal['B'] = '0'
        cost += 1
        print("Cost for cleaning " + str(cost))
        print("Location B has been Cleaned.")

        if status_input_complement == '1':
            print("Location A is Dirty.")
            print("Moving left to the Location A. ")
            cost += 1
            print("Cost for moving left " + str(cost))
            goal['A'] = '0'
```

```
                cost += 1
                print("Cost for sucking A " + str(cost))
                print("Location A has been Cleaned.")

        else:
            print(cost)
            print("Location B is already clean.")

            if status_input_complement == '1':
                print("Location A is Dirty.")
                print("Moving left to the Location A. ")
                cost += 1
                print("Cost for moving left " + str(cost))
                goal_state['A'] = '0'
                cost += 1
                print("Cost for sucking A " + str(cost))
                print("Location A has been Cleaned. ")
            else:
                print("No action needed for A")
                print("Location A is already clean.")

    print("Goal State: ")
    print(goal)
    print("Performance Measurement: " + str(cost))

vacuum_cleaner()
```

Screenshots of the Outputs:



```
Enter Location of Vacuum A
Enter status of A1
Enter status of other room 0
Initial Location Condition {'A': '0', 'B': '0'}
Vacuum is placed in Location A
Location A is Dirty.
Cost for cleaning A 1
Location A has been Cleaned.
No action needed for B
Location B is already clean.
Goal State:
{'A': '0', 'B': '0'}
Performance Measurement: 1
```

Signature of the Student

[YUVRAJ SINGH CHAUHAN]