

AI LAB 3- Toy's Problem (Vacuum Problem)

Name- Priyanshu Rishi

Reg No- RA1911042010041

Batch-CSBS(R1)

Code:

```
def vacuum_cleaner():
    final = {'A': '0', 'B': '0'} #0-clean 1-dirty
    steps = 0
    location= input("Enter Location of Vacuum")
    status= input("Enter status of " + location)
    status_temp = input("Enter status of other room")
    print("Initial Location Condition" + str(final))

    if location == 'A':
        print("Vacuum is placed in Location A")
        if status== '1':
            print("Location A is Dirty.")
            final['A'] = '0'
            steps += 1
            print("Cost for CLEANING A " + str(steps))
            print("Location A has been Cleaned.")

        if status_temp == '1':
            print("Location B is Dirty.")
            print("Moving right to the Location B. ")
            steps += 1
            print("COST for moving RIGHT" + str(steps))
            final['B'] = '0'
            steps += 1
```

```
print("COST for SUCK " + str(steps))  
print("Location B has been Cleaned. ")  
else:  
    print("No action" + str(steps))  
    print("Location B is already clean.")
```

```
if status == '0':  
    print("Location A is already clean ")  
    if status_temp == '1':  
        print("Location B is Dirty.")  
        print("Moving RIGHT to the Location B. ")  
        steps += 1  
        print("COST for moving RIGHT " + str(steps))  
        final['B'] = '0'  
        steps += 1  
        print("Cost for SUCK" + str(steps))  
        print("Location B has been Cleaned. ")  
    else:  
        print("No action " + str(steps))  
        print(steps)  
        print("Location B is already clean.")
```

```
else:  
    print("Vacuum is placed in location B")  
    if status == '1':  
        print("Location B is Dirty.")  
        final['B'] = '0'  
        steps += 1  
        print("COST for CLEANING " + str(steps))  
        print("Location B has been Cleaned.")
```

```
if status_temp == '1':  
    print("Location A is Dirty.")  
    print("Moving LEFT to the Location A. ")  
    steps += 1  
    print("COST for moving LEFT" + str(steps))  
    final['A'] = '0'  
    steps += 1  
    print("COST for SUCK " + str(steps))  
    print("Location A has been Cleaned.")
```

else:

```
print(steps)  
print("Location B is already clean.")
```

```
if status_temp == '1':  
    print("Location A is Dirty.")  
    print("Moving LEFT to the Location A. ")  
    steps += 1  
    print("COST for moving LEFT " + str(steps))
```

```
    final['A'] = '0'  
    steps += 1  
    print("Cost for SUCK " + str(steps))  
    print("Location A has been Cleaned. ")
```

else:

```
    print("No action " + str(steps))  
  
    print("Location A is already clean.")
```

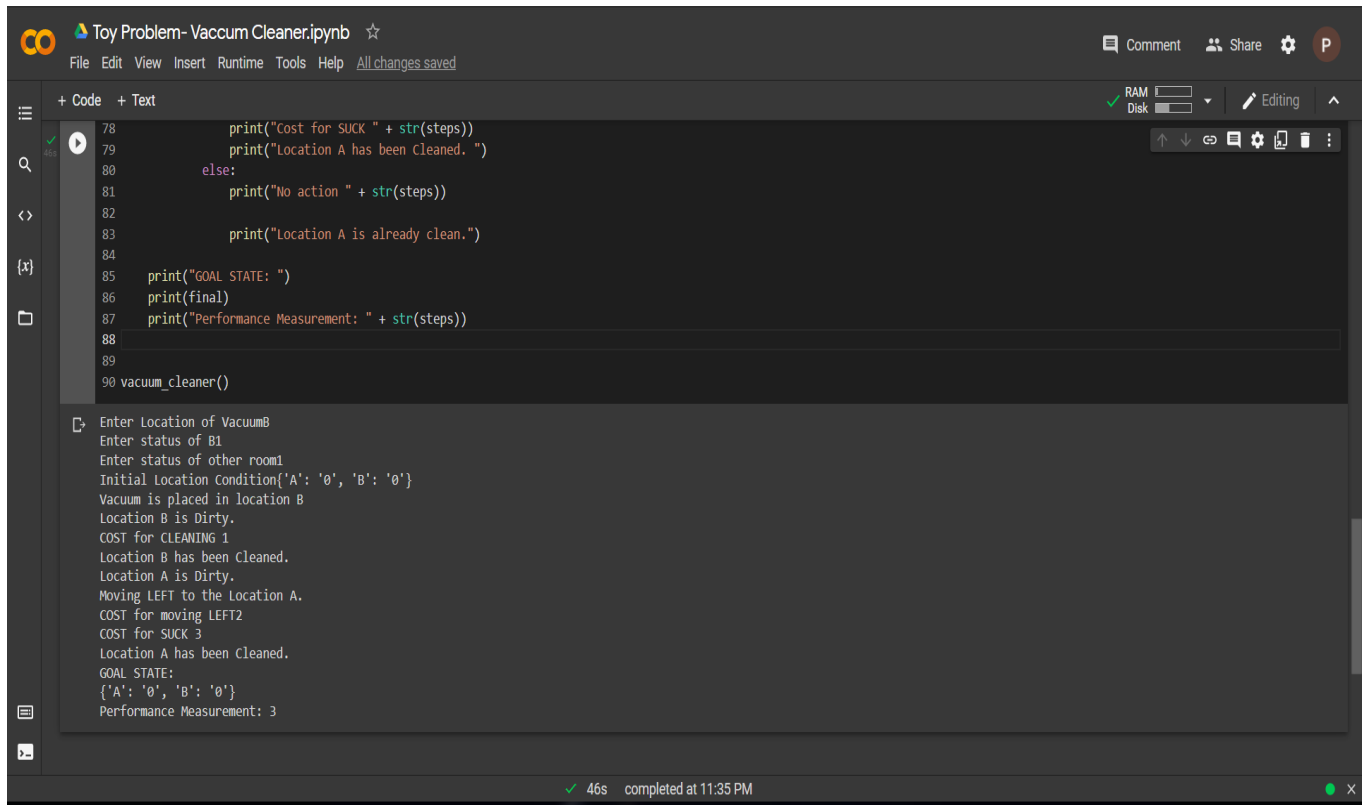
```
print("GOAL STATE: ")
```

```
print(final)
```

```
print("Performance Measurement: " + str(steps))
```

```
vacuum_cleaner()
```

Output:



The screenshot shows a Jupyter Notebook titled "Toy Problem- Vacuum Cleaner.ipynb". The code cell contains the following Python code:

```
78     print("Cost for SUCK " + str(steps))
79     print("Location A has been Cleaned. ")
80     else:
81         print("No action " + str(steps))
82
83         print("Location A is already clean.")
84
85     print("GOAL STATE: ")
86     print(final)
87     print("Performance Measurement: " + str(steps))
88
89
90 vacuum_cleaner()
```

The output cell shows the following text:

```
Enter Location of VacuumB
Enter status of B1
Enter status of other room1
Initial Location Condition{'A': '0', 'B': '0'}
Vacuum is placed in location B
Location B is Dirty.
COST for CLEANING 1
Location B has been Cleaned.
Location A is Dirty.
Moving LEFT to the Location A.
COST for moving LEFT2
COST for SUCK 3
Location A has been Cleaned.
GOAL STATE:
{'A': '0', 'B': '0'}
Performance Measurement: 3
```

The bottom status bar indicates the notebook was completed in 46s at 11:35 PM.