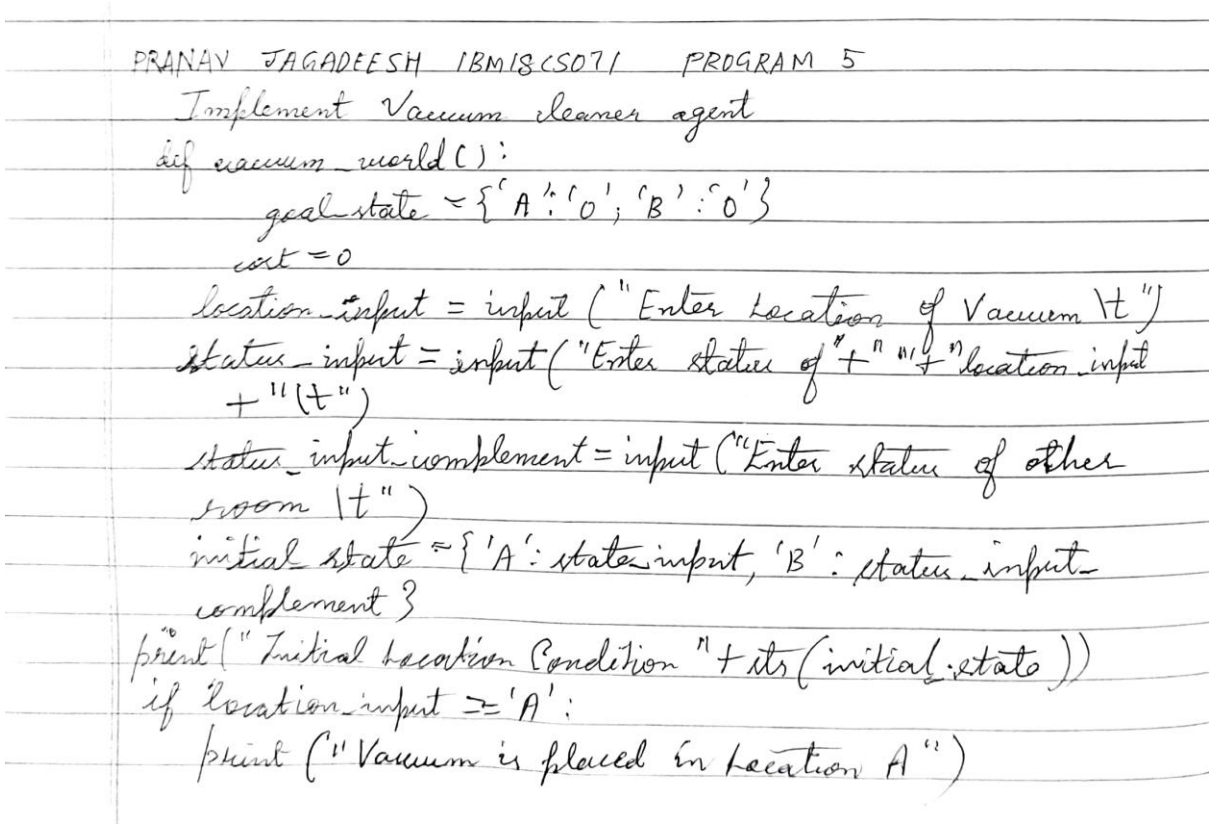
Program 5: Implement vacuum cleaner agent.

Write-up:



```
PRANAV JAGADEESH 1BM18CS071   PROGRAM 5
  Implement Vacuum cleaner agent
def vacuum_world():
    goal_state = {'A':'0','B':'0'}
    cost = 0
  location_input = input("Enter location of Vacuum \t")
  status_input = input("Enter status of "+" "+location_input
    +"(t")
  status_input_complement = input("Enter status of other
  room \t")
  initial_state = {'A': status_input, 'B': status_input_
  complement}
print("Initial location Condition "+str(initial_state))
if location_input == 'A':
    print("Vacuum is placed in location A")
```

```
if status_input == '1':
    print("Location A is Dirty.")
    goal_state['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_state['B'] = '0'
        cost += 1
        print("Cost for suck " + str(cost))
        print("Location B has been Cleaned.")
    else:
        print("No action" + str(cost))
        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_state['B'] = '0'
    cost += 1
    print("No action " + str(cost))
    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_state['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_state['A'] = '0'
    cost += 1
    print("Cost for suck " + str(cost))
    print("Location A has been cleaned.")
else
    print(cost)
    print("Location B is already clean.")
    if status_input_complement == '0':
        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 SUCK " + str(cost))
        print("Location A has been cleaned.")
    else:
        print("No action " + str(cost))
        print("Location A is already clean.")
```

```
print ("GOAL STATE:")
print (goal_state)
print ("Performance Measurement: " + str(cost))
vacuum_world()
```

Output: Enter Location of Vacuum A
Enter status of A 1
Enter status of other room 1
Initial location condition {'A': '1', 'B': '1'}
Vacuum is placed in location A
Location A is Dirty
Cost for CLEANING A 1
Location A has been cleaned.
Location B is Dirty.
Moving right to the location B.
Cost for moving RIGHT 2
Cost for Suck: 3
Location B has been cleaned.
GOAL STATE:
{'A': '0', 'B': '0'}
Performance Measurement: 3

Output 2: Enter Location of Vacuum: B
Enter status of B 1
Enter status of other Room 1
Initial Location Condition {'A': '1', 'B': '1'}
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 Left 2
Cost for suck 3
Location A has been cleaned
Goal State:
{'A': '0', 'B': '0'}
Performance Measurement: 3

Output 3: Enter location of Vacuum A
Enter status of A 0
Enter status of other room 0
Initial location condition {'A': '0', 'B': '0'}
Vacuum is placed in location A
Location A is already clean
No action 0
0
Location B is already clean.
GOAL state:
{'A': '0', 'B': '0'}
Performance Measurement: 0

Output 4: Enter Location of Vacuum B
Enter status of B 0
Enter status of other room 0
Initial location condition {'A': '0', 'B': '0'}
Vacuum is placed in location B
0
Location B is already clean
No action 0
Location A is already clean
Goal state:
{'A': '0', 'B': '0'}
Performance Measurement: 0

Program:

```
def vacuum_world():

    goal_state = {'A': '0', 'B': '0'}
    cost = 0

    location_input = input("Enter Location of Vacuum \t")
    status_input = input("Enter status of"+" " + location_input + "\t")
    status_input_complement = input("Enter status of other room \t")
    initial_state = {'A' : status_input , 'B' : status_input_complement}
    print("Initial Location Condition" + str(initial_state))

    if location_input == 'A':

        print("Vacuum is placed in Location A")
        if status_input == '1':
            print("Location A is Dirty.")

            goal_state['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_state['B'] = '0'
            cost += 1
            print("COST for SUCK " + str(cost))
            print("Location B has been Cleaned. ")
        else:
            print("No action" + str(cost))

            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_state['B'] = '0'
            cost += 1
            print("Cost for SUCK" + str(cost))
            print("Location B has been Cleaned. ")
        else:
            print("No action " + str(cost))
            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_state['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_state['A'] = '0'
        cost += 1
        print("COST for SUCK " + 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 SUCK " + str(cost))
            print("Location A has been Cleaned. ")
        else:
            print("No action " + str(cost))

            print("Location A is already clean.")


    print("GOAL STATE: ")
    print(goal_state)
    print("Performance Measurement: " + str(cost))


vacuum_world()
```

Output:1

```
Enter Location of Vacuum    A
Enter status of A   1
Enter status of other room  1
Initial Location Condition{'A': '1', 'B': '1'}
Vacuum is placed in Location A
Location A is Dirty.
Cost for CLEANING A 1
Location A has been Cleaned.
Location B is Dirty.
Moving right to the Location B.
COST for moving RIGHT2
COST for SUCK 3
Location B has been Cleaned.
GOAL STATE:
{'A': '0', 'B': '0'}
Performance Measurement: 3
```

Output:2

```
Enter Location of Vacuum    B
Enter status of B   1
Enter status of other room  1
Initial Location Condition{'A': '1', 'B': '1'}
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
```

Output:3

```
Enter Location of Vacuum    A
Enter status of A   0
Enter status of other room  0
Initial Location Condition{'A': '0', 'B': '0'}
Vacuum is placed in Location A
Location A is already clean
No action 0
0
Location B is already clean.
GOAL STATE:
{'A': '0', 'B': '0'}
Performance Measurement: 0
```

Output:4

```
Enter Location of Vacuum    B
Enter status of B   0
Enter status of other room  0
Initial Location Condition{'A': '0', 'B': '0'}
Vacuum is placed in location B
0
Location B is already clean.
No action 0
Location A is already clean.
GOAL STATE:
{'A': '0', 'B': '0'}
Performance Measurement: 0
```