2. Implement vacuum cleaner agent.

Code:

```
def vacuum_world():
  # Initializing goal_state
  # 0 indicates Clean and 1 indicates Dirty
  goal_state = {'A': '0', 'B': '0'}
  cost = 0
  print("Sonal,1BM22CS286")
  location_input = input("Enter Location of Vacuum (A or B): ").strip().upper() # User input of
location
  status_input = input(f"Enter status of {location_input} (0 for Clean, 1 for Dirty): ").strip()
  status_input_complement = input("Enter status of other room (0 for Clean, 1 for Dirty):
").strip()
  print("Initial Location Condition: " + str(goal 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' # Clean A
       cost += 1 # Cost for sucking
       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 Location B.")
         cost += 1 # Cost for moving right
         print("Cost for moving RIGHT: " + str(cost))
         goal_state['B'] = '0' # Clean B
         cost += 1 # Cost for sucking
         print("Cost for SUCK: " + str(cost))
         print("Location B has been Cleaned.")
       else:
         print("Location B is already clean.")
    else:
       print("Location A is already clean.")
       if status_input_complement == '1':
         print("Location B is Dirty.")
         print("Moving RIGHT to Location B.")
```

```
cost += 1 # Cost for moving right
       print("Cost for moving RIGHT: " + str(cost))
       goal_state['B'] = '0' # Clean B
       cost += 1 # Cost for sucking
       print("Cost for SUCK: " + str(cost))
       print("Location B has been Cleaned.")
    else:
       print("Location B is already clean.")
elif location input == 'B':
  print("Vacuum is placed in Location B")
  if status input == '1':
    print("Location B is Dirty.")
    goal_state['B'] = '0' # Clean B
    cost += 1 # Cost for sucking
    print("Cost for CLEANING B: " + str(cost))
    print("Location B has been Cleaned.")
    if status_input_complement == '1':
       print("Location A is Dirty.")
       print("Moving LEFT to Location A.")
       cost += 1 # Cost for moving left
       print("Cost for moving LEFT: " + str(cost))
       goal_state['A'] = '0' # Clean A
       cost += 1 # Cost for sucking
       print("Cost for SUCK: " + str(cost))
       print("Location A has been Cleaned.")
    else:
       print("Location A is already clean.")
    print("Location B is already clean.")
    if status_input_complement == '1':
       print("Location A is Dirty.")
       print("Moving LEFT to Location A.")
       cost += 1 # Cost for moving left
       print("Cost for moving LEFT: " + str(cost))
       goal_state['A'] = '0' # Clean A
       cost += 1 # Cost for sucking
       print("Cost for SUCK: " + str(cost))
       print("Location A has been Cleaned.")
       print("Location A is already clean.")
```

```
# Done cleaning
print("GOAL STATE: ")
print(goal_state)
print("Performance Measurement: " + str(cost))

# Output
vacuum_world()
```

Output:

```
→ Sonal,1BM22CS286
    Enter Location of Vacuum (A or B): A
    Enter status of A (0 for Clean, 1 for Dirty): 1
    Enter status of other room (0 for Clean, 1 for Dirty): 1
    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.
    Location B is Dirty.
    Moving right to 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
```

Observation book screenshots:

OBSCIVATION BOOK CONCONCIONS.
Page 5
Tulkalize task input (either A or B)
SALO'B' O'S
and shall and a doo
Intralize good state Intralize took Totalize took
encational, elean of all
Intralize cost of input (elthree A or B) Intralize cost of input (elthree A or B) Input: location input (clean or dirty for the Input: states input (clean or dirty for the Current room) Current complement (clean or
cittle dements
the status input office room)
Input: states input (alean or author) Tuput: status input complement (clean or dirry for the other moon)
A Lines Alleria al a su lay had
a 4 slatus Exput = 1:00
of slatus expet (1) = 0)
a. y
location input a di slatus input ii Incase cost by 1 ii Trease cost by 1 ii organisment = 1'
e stangement
b. Sy states input complement = 1 b. Move vacuum to Roam B
The state of the s
ii. Clean Room & Cast goal state (B) = 0) Vol
III. Clean Room Com
iv. Increase cost by 1.
c. Else: Room B is already clean, do
nothing
- det 1
fle # il location input = 1's:
le el sterler e sumit == 1.
1. Clean Room B (set goal state (B)=0)
L. Clean Store of
ii. Increase cost by a
h 11 status input complement == 1:
b. 1 status input complement == 1':
i. More vacuum to Room A
ii. Increase cost by 1
The second of th
iii. clean Room A (set goal state[A]=0)
iv. Increase cost by 1.
else
C. Ette: Room A is alreasy clean do nothing
Print and state and it is
Print good state and performance (cost)