```
def vacuum_world():
    # Initializing goal_state
    # 0 indicates Clean and 1 indicates Dirty
    goal_state = {'A': '0', 'B': '0'}
    cost = 0
    location_input = input("Enter Location of Vacuum (A/B): ").strip().upper()
    status_input = input(f"Enter status of {location_input} (0 for Clean, 1 for
    other_location = 'B' if location_input == 'A' else 'A'
    status_input_complement = input(f"Enter status of {other_location} (0 for C
    print("Vignesh B 1BM22CS326 ")
    print("Initial Location Condition:", goal_state)

    if location_input == 'A':
        print("Vacuum is placed in Location A")
        if status_input == '1':
            print("Location A is Dirty.")
            # Suck the dirt and mark it as clean
            goal_state['A'] = '0'
            cost += 1  # Cost for suck
            print("Cost for CLEANING A:", 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:", cost)
                # Suck the dirt and mark it as clean
                goal_state['B'] = '0'
                cost += 1  # Cost for suck
                print("COST for SUCK:", 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:", cost)
                # Suck the dirt and mark it as clean
                goal_state['B'] = '0'
                cost += 1  # Cost for suck
                print("COST for SUCK:", cost)
                print("Location B has been Cleaned.")
```

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

    else:  # Vacuum is placed in Location B
        print("Vacuum is placed in Location B")
        if status_input == '1':
            print("Location B is Dirty.")
            # Suck the dirt and mark it as clean
            goal_state['B'] = '0'
            cost += 1  # Cost for suck
            print("COST for CLEANING B:", 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:", cost)
                # Suck the dirt and mark it as clean
                goal_state['A'] = '0'
                cost += 1  # Cost for suck
                print("COST for SUCK:", cost)
                print("Location A has been Cleaned.")
            else:
                print("Location A is already clean.")

        else:
            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:", cost)
                # Suck the dirt and mark it as clean
                goal_state['A'] = '0'
                cost += 1  # Cost for suck
                print("COST for SUCK:", cost)
                print("Location A has been Cleaned.")
            else:
                print("Location A is already clean.")

    # Done cleaning
    print("GOAL STATE:", goal_state)
    print("Performance Measurement:", cost)

# Call the function
vacuum_world()
```

```
Enter Location of Vacuum (A/B): A
Enter status of A (0 for Clean, 1 for Dirty): 0
Enter status of B (0 for Clean, 1 for Dirty): 1
Vignesh B 1BM22CS326
Initial Location Condition: {'A': '0', 'B': '0'}
Vacuum is placed in Location A
Location A is already clean.
Location B is Dirty.
Moving RIGHT to Location B.
COST for moving RIGHT: 1
COST for SUCK: 2
Location B has been Cleaned.
GOAL STATE: {'A': '0', 'B': '0'}
Performance Measurement: 2
```