

(1) CODE FOR VACUUM CLEANER

```
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 or B): ") # User input for location  
    status_input = input("Enter status of " + location_input + " (0 for Clean, 1 for Dirty): ") # User input  
    for status  
    status_input_complement = input("Enter status of the other room (0 for Clean, 1 for Dirty): ")  
  
    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.")  
            # Suck the dirt and mark it as clean  
            goal_state['A'] = '0'  
            cost += 1 # Cost for suck  
            print("Cost for CLEANING A: " + str(cost))  
            print("Location A has been cleaned.")  
  
            if status_input_complement == '1': # If B is Dirty  
                print("Location B is Dirty.")  
                print("Moving right to Location B.")  
                cost += 1 # Cost for moving right  
                print("Cost for moving RIGHT: " + str(cost))  
                # Suck the dirt and mark it as clean  
                goal_state['B'] = '0'
```

```
cost += 1 # Cost for suck
print("Cost for SUCK: " + str(cost))
print("Location B has been cleaned.")
else:
    print("No action required; Location B is already clean.")
```

```
else:
    print("Location A is already clean.")
    if status_input_complement == '1': # If B is Dirty
        print("Location B is Dirty.")
        print("Moving right to Location B.")
        cost += 1 # Cost for moving right
        print("Cost for moving RIGHT: " + str(cost))
        # Suck the dirt and mark it as clean
        goal_state['B'] = '0'
        cost += 1 # Cost for suck
        print("Cost for SUCK: " + str(cost))
        print("Location B has been cleaned.")
    else:
        print("No action required; 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: " + str(cost))
        print("Location B has been cleaned.")
```

```

if status_input_complement == '1': # If A is Dirty
    print("Location A is Dirty.")
    print("Moving left to Location A.")
    cost += 1 # Cost for moving left
    print("Cost for moving LEFT: " + str(cost))
    # Suck the dirt and mark it as clean
    goal_state['A'] = '0'
    cost += 1 # Cost for suck
    print("Cost for SUCK: " + str(cost))
    print("Location A has been cleaned.")
else:
    print("No action required; Location A is already clean.")

```

```

else:
    print("Location B is already clean.")
    if status_input_complement == '1': # If A is Dirty
        print("Location A is Dirty.")
        print("Moving left to Location A.")
        cost += 1 # Cost for moving left
        print("Cost for moving LEFT: " + str(cost))
        # Suck the dirt and mark it as clean
        goal_state['A'] = '0'
        cost += 1 # Cost for suck
        print("Cost for SUCK: " + str(cost))
        print("Location A has been cleaned.")
    else:
        print("No action required; Location A is already clean.")

```

```

# Done cleaning
print("GOAL STATE: ")
print(goal_state)

```

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

```
# Call the function
```

```
vacuum_world()
```

OUTPUT:

```
Enter location of Vacuum (A or B): A
Enter status of A (0 for Clean, 1 for Dirty): 1
Enter status of the other room (0 for Clean, 1 for Dirty): 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 required; Location B is already clean.
GOAL STATE:
{'A': '0', 'B': '0'}
Performance Measurement: 1

=== Code Execution Successful ===
```