```
 ['-' '-' '-']]
Enter the Row Value (0-2): 2
Enter the Column Value (0-2): 1
[['X' 'O' 'X']
 ['X' 'O' 'X']
 ['-' 'O' '-']]
The winner is: O
```

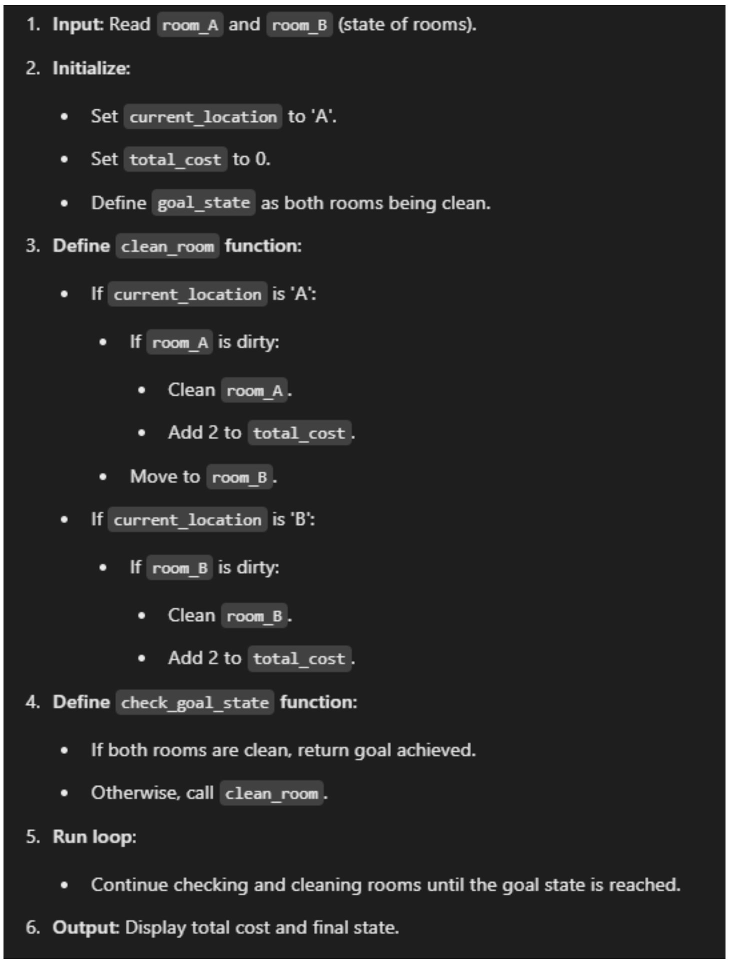
LAB 2

```
#2QUADS
room_A = int(input("enter the state"))
room_B =  int(input("enter the state"))
current_location = 'A'

goal_state = ['A', 0, 'B', 0]

total_cost = 0

# Step 4: Define a function to clean the room
def clean_room():
    global room_A, room_B, current_location, total_cost

    # Check the current location
    if current_location == 'A':
        print("Vacuum is placed in Location A")
        if room_A == 1:
            print("Location A is Dirty")
            print("Cleaning Location A...")
            room_A = 0  # Clean the room
            total_cost += 2
            print("Location A has been Cleaned")
            print("COST for SUCK in Location A: 2 units")
        else:
            print("Location A is already clean")

        # Move to Location B (no cost for moving)
        print("Moving right to Location B")
        current_location = 'B'

    if current_location == 'B':
        print("Vacuum is placed in Location B")
        if room_B == 1:
            print("Location B is Dirty")
            print("Cleaning Location B...")
            room_B = 0  # Clean the room
            total_cost += 2  # Cost of cleaning (SUCK action)
            print("Location B has been Cleaned")
            print("COST for SUCK in Location B: 2 units")
        else:
            print("Location B is already clean")

# Step 5: Keep checking for the goal state
def check_goal_state():
    if room_A == goal_state[1] and room_B == goal_state[3]:
        print("Both rooms are clean! Goal state reached:", goal_state)
        return True  # Goal state reached, stop the vacuum cleaner
    else:
        clean_room()  # Continue cleaning until the goal state is reached
        return False

# Step 6: Run the vacuum cleaner
goal_reached = False  # A flag to track if the goal is reached

while not goal_reached:  # Loop until the goal state is reached
    goal_reached = check_goal_state()

# Final output after cleaning
print("Final goal state:", goal_state)
print("Total cost incurred:", total_cost, "units")
```

```
enter the state1
enter the state0
Vacuum is placed in Location A
Location A is Dirty
Cleaning Location A...
Location A has been Cleaned
COST for SUCK in Location A: 2 units
Moving right to Location B
Vacuum is placed in Location B
```

```
Location B is already clean
Both rooms are clean! Goal state reached: ['A', 0, 'B', 0]
Final goal state: ['A', 0, 'B', 0]
Total cost incurred: 2 units
```

PSEUDOCODE



1. **Input:** Read `room_A` and `room_B` (state of rooms).
2. **Initialize:**
   - Set `current_location` to 'A'.
   - Set `total_cost` to 0.
   - Define `goal_state` as both rooms being clean.
3. **Define** `clean_room` **function:**
   - If `current_location` is 'A':
     - If `room_A` is dirty:
       - Clean `room_A`.
       - Add 2 to `total_cost`.
     - Move to `room_B`.
   - If `current_location` is 'B':
     - If `room_B` is dirty:
       - Clean `room_B`.
       - Add 2 to `total_cost`.
4. **Define** `check_goal_state` **function:**
   - If both rooms are clean, return goal achieved.
   - Otherwise, call `clean_room`.
5. **Run loop:**
   - Continue checking and cleaning rooms until the goal state is reached.
6. **Output:** Display total cost and final state.

```
#4 quads
room_A = int(input("Enter the state for room A  "))
room_B = int(input("Enter the state for room B  "))
room_C = int(input("Enter the state for room C  "))
room_D = int(input("Enter the state for room D  "))

current_location = 'A'

goal_state = ['A', 0, 'B', 0, 'C', 0, 'D', 0]

total_cost = 0

def clean_room():
    global room_A, room_B, room_C, room_D, current_location, total_cost

    if current_location == 'A':
        print("Vacuum is placed in Location A")
        if room_A == 1:
            print("Location A is Dirty")
            print("Cleaning Location A...")
            room_A = 0
            total_cost += 2
            print("Location A has been Cleaned")
            print("COST for SUCK in Location A: 2 units")
        else:
            print("Location A is already clean")

        print("Moving right to Location B")
        current_location = 'B'

    elif current_location == 'B':
        print("Vacuum is placed in Location B")
        if room_B == 1:
            print("Location B is Dirty")
            print("Cleaning Location B...")
            room_B = 0
            total_cost += 2
            print("Location B has been Cleaned")
            print("COST for SUCK in Location B: 2 units")
        else:
            print("Location B is already clean")

        print("Moving down to Location C")
        current_location = 'C'

    elif current_location == 'C':
        print("Vacuum is placed in Location C")
        if room_C == 1:
            print("Location C is Dirty")
            print("Cleaning Location C...")
            room_C = 0
            total_cost += 2
            print("Location C has been Cleaned")
            print("COST for SUCK in Location C: 2 units")
        else:
            print("Location C is already clean")

        print("Moving right to Location D")
        current_location = 'D'

    elif current_location == 'D':
        print("Vacuum is placed in Location D")
        if room_D == 1:
            print("Location D is Dirty")
            print("Cleaning Location D...")
            room_D = 0
            total_cost += 2
            print("Location D has been Cleaned")
            print("COST for SUCK in Location D: 2 units")
        else:
            print("Location D is already clean")

def check_goal_state():
    if room_A == goal_state[1] and room_B == goal_state[3] and room_C == goal_state[5] and room_D == goal_state[7]:
        print("All rooms are clean! Goal state reached:", goal_state)
        return True
    else:
        clean_room()
        return False

goal_reached = False

while not goal_reached:
    goal_reached = check_goal_state()

print("Final goal state:", goal_state)
print("Total cost incurred:", total_cost, "units")
```

```
Enter the state for room A  1
Enter the state for room B  0
Enter the state for room C  1
Enter the state for room D  1
Vacuum is placed in Location A
Location A is Dirty
Cleaning Location A...
Location A has been Cleaned
COST for SUCK in Location A: 2 units
Moving right to Location B
Vacuum is placed in Location B
Location B is already clean
Moving down to Location C
Vacuum is placed in Location C
Location C is Dirty
Cleaning Location C...
Location C has been Cleaned
COST for SUCK in Location C: 2 units
Moving right to Location D
Vacuum is placed in Location D
Location D is Dirty
Cleaning Location D...
Location D has been Cleaned
COST for SUCK in Location D: 2 units
All rooms are clean! Goal state reached: ['A', 0, 'B', 0, 'C', 0, 'D', 0]
Final goal state: ['A', 0, 'B', 0, 'C', 0, 'D', 0]
Total cost incurred: 6 units
```