

Lab 2: Vaccume world

Code:

```
from ctypes import c_char
```

```
cost=0
```

```
status=[0,0];
```

```
A=0;
```

```
B=1;
```

```
def suck(location):
```

```
    global cost
```

```
    if(status[location]==0):
```

```
        print("Room is already clean.")
```

```
    else:
```

```
        status[location]=0
```

```
        cost+=1
```

```
        print("Room cleaned.")
```

```
        if(location==A):
```

```
            print("A still dirty? (1 for yes, 0 for no)")
```

```
            status[A]=int(input())
```

```
        elif(location==B):
```

```
            print("A still dirty? (1 for yes, 0 for no)")
```

```
            status[B]=int(input())
```

```
def left(location):
```

```
    print("Moved to room A.")
```

```
    return A
```

```
def right(location):
```

```
    print("Moved to room B.")
```

```
    return B
```

```
def vaccume_cleaner(location):
```

```
    global cost
```

```
    global status
```

```
    global A,B
```

```
    if(status[A]==0 and status[B]==0):
```

```
        print("All rooms cleaned. Cost: " + str(cost))
```

```
        return
```

```
    if(status[location]==1):
```

```
        suck(location)
```

```
    else:
```

```
        print("Room is already clean.")
```

```
    if(location==A):
```

```
        new_loc=right(location)
```

```
        vaccume_cleaner(new_loc)
```

```
    elif(location==B):
```

```
        new_loc=left(location)
```

```
        vaccume_cleaner(new_loc)
```

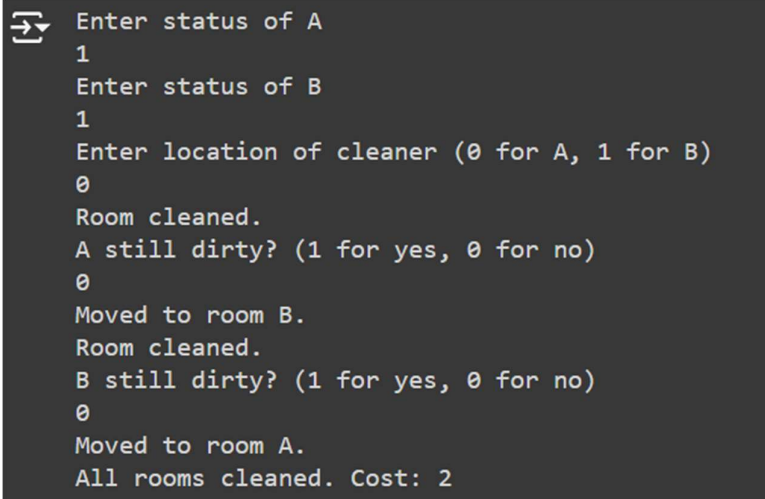
```
def main():
```

```
global A,B
global status
print("Enter status of A")
status[A]=int(input())
print("Enter status of B")
status[B]=int(input())
print("Enter location of cleaner (0 for A, 1 for B)")
location=int(input())
vaccume_cleaner(location)
```

main()

Output:

1. State {A : 1, B : 1}

A terminal window with a dark background and light-colored text. It shows the execution of a program. The user enters '1' for the status of A, '1' for the status of B, and '0' for the location of the cleaner. The program outputs 'Room cleaned.', asks 'A still dirty? (1 for yes, 0 for no)', and the user enters '0'. It then outputs 'Moved to room B.', 'Room cleaned.', asks 'B still dirty? (1 for yes, 0 for no)', and the user enters '0'. Finally, it outputs 'Moved to room A.' and 'All rooms cleaned. Cost: 2'.

```
➡ Enter status of A
1
Enter status of B
1
Enter location of cleaner (0 for A, 1 for B)
0
Room cleaned.
A still dirty? (1 for yes, 0 for no)
0
Moved to room B.
Room cleaned.
B still dirty? (1 for yes, 0 for no)
0
Moved to room A.
All rooms cleaned. Cost: 2
```

2. State {A : 0, B : 0}

```
➡ Enter status of A
0
Enter status of B
0
Enter location of cleaner (0 for A, 1 for B)
0
All rooms cleaned. Cost: 0
```

3. State {A : 1, B : 0}

```
➡ Enter status of A
1
Enter status of B
0
Enter location of cleaner (0 for A, 1 for B)
0
Room cleaned.
A still dirty? (1 for yes, 0 for no)
0
Moved to room B.
All rooms cleaned. Cost: 1
```

4. State {A : 0, B : 1}

```
➡ Enter status of A
0
Enter status of B
1
Enter location of cleaner (0 for A, 1 for B)
0
Room is already clean.
Moved to room B.
Room cleaned.
B still dirty? (1 for yes, 0 for no)
0
Moved to room A.
All rooms cleaned. Cost: 1
```