Uploaded By: Supreet Ganguly

Reg no: RA1811003010984 & RA1811003010981

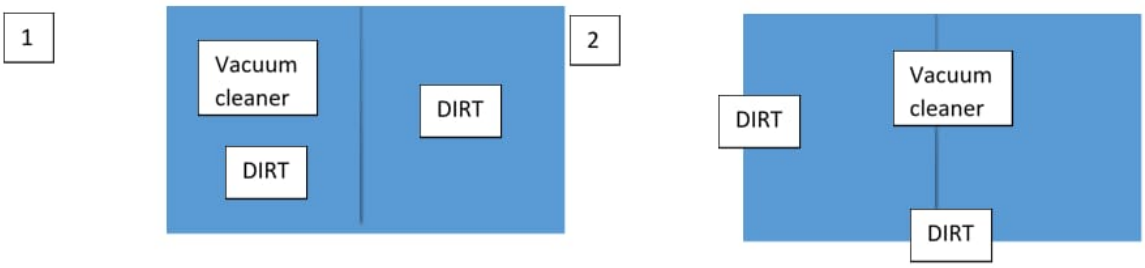
Expt2: Vaccum Cleaner Problem

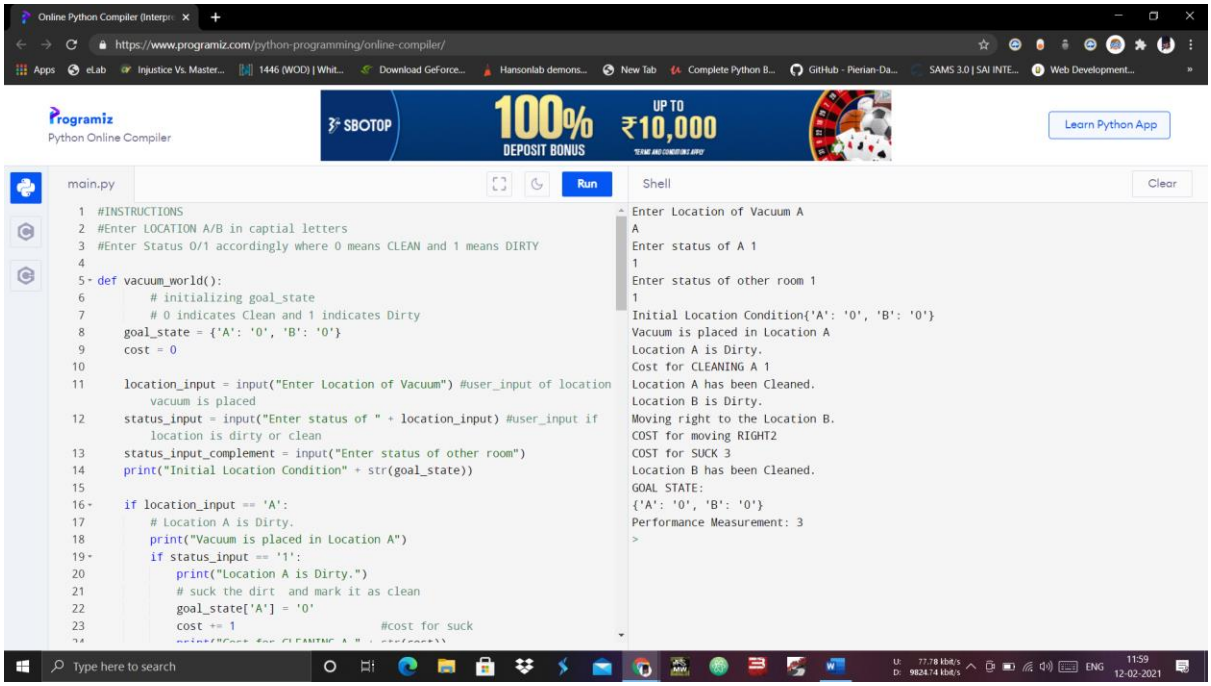
```
#INSTRUCTIONS
#Enter LOCATION A/B in captial letters
#Enter Status O/1 accordingly where 0 means CLEAN and 1 means DIRTY

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") #user_input of location vacuum is placed
  status_input = input("Enter status of " + location_input) #user_input if location is dirty or clean
  status_input_complement = input("Enter status of other room")
  print("Initial Location Condition" + str(goal_state))

  if location_input == 'A':
    # Location A is Dirty.
    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 the 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" + str(cost))
        # suck and mark clean
        print("Location B is already clean.")

if status_input == '0':
    print("Location A is already clean ")
    if status_input_complement == '1':# if B is Dirty
        print("Location B is Dirty.")
        print("Moving RIGHT to the 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 " + str(cost))
```

```
            print(cost)
            # suck and mark clean
            print("Location B is already clean.")

else:
    print("Vacuum is placed in location B")
    # Location B is Dirty.
    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 " + 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 the Location A. ")
            cost += 1  # cost for moving right
            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(cost)
        # suck and mark clean
        print("Location B is already clean.")
```

```
        if status_input_complement == '1':  # if A is Dirty
            print("Location A is Dirty.")
            print("Moving LEFT to the Location A. ")
            cost += 1  # cost for moving right
            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 " + str(cost))
            # suck and mark clean
            print("Location A is already clean.")

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


vacuum_world()
```




Online Python Compiler (Interpr
https://www.programiz.com/python-programming/online-compiler/
Apps
eLab
Injustice Vs. Master...
1446 (WOD) | Whit...
Download GeForce...
Hansonlab demons...
New Tab
Complete Python B...
GitHub - Pierian-Da...
SAMS 3.0 | SAI INTE...
Web Development...
Programiz
Python Online Compiler
SBOTOP
100%
DEPOSIT BONUS
UP TO
₹10,000
Learn Python App
main.py
Run
Shell
Clear
1 #INSTRUCTIONS
2 #Enter LOCATION A/B in captial letters
3 #Enter Status 0/1 accordingly where 0 means CLEAN and 1 means DIRTY
4
5 def vacuum_world():
6 # initializing goal_state
7 # 0 indicates Clean and 1 indicates Dirty
8 goal_state = {'A': '0', 'B': '0'}
9 cost = 0
10
11 location_input = input("Enter Location of Vacuum") #user_input of location vacuum is placed
12 status_input = input("Enter status of " + location_input) #user_input if location is dirty or clean
13 status_input_complement = input("Enter status of other room")
14 print("Initial Location Condition" + str(goal_state))
15
16 if location_input == 'A':
17 # Location A is Dirty.
18 print("Vacuum is placed in Location A")
19 if status_input == '1':
20 print("Location A is Dirty.")
21 # suck the dirt and mark it as clean
22 goal_state['A'] = '0'
23 cost += 1 #cost for suck
Enter Location of Vacuum A
A
Enter status of A 1
1
Enter status of other room 1
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 the Location B.
COST for moving RIGHT2
COST for SUCK 3
Location B has been Cleaned.
GOAL STATE:
{'A': '0', 'B': '0'}
Performance Measurement: 3
>
Type here to search
U: 77.78 kbit/s
D: 9824.74 kbit/s
ENG
11:59
12-02-2021