

## مسئله ی دنیای جارو برقی با عامل ساده

در این مسئله ما قصد داریم که یک جاروبرقی با عامل ساده ایجاد کنیم که در یک محیط (دارای دو اتاق راست و چپ) قرار می گیرد و بررسی می کند که آیا این اتاق تمیز است یا آلوده و در صورت آلوده بودن اقدام به تمیز نمودن اتاق می نماید و در آخر امتیاز معیار کارایی عامل را باز می گرداند.

این مسئله شامل یک تابع می شود که از ورودی یک مقدار برای مکان جاروبرقی دریافت می کند که می تواند در یکی از اتاق های چپ یا راست باشد. برای این منظور یکی از مقادیر A یا B را دریافت می کند.

سپس یک مقدار وضعیت 0 یا 1 برای اتاقی که جاروبرقی در آن قرار دارد و یکی دیگر برای اتاق دیگری دریافت می کند که 0 معادل تمیز بودن و 1 معادل کثیف بودن اتاق می باشد.

جاروبرقی پس از بررسی اتاق اولیه و تمیز نمودن آن در صورت آلوده بودن ، به اتاق دیگر می رود و آن را نیز در صورت آلوده بودن تمیز می نماید و عملیات جاروبرقی خاتمه می یابد.

برای بررسی حالت های مختلف این مسئله (که کلاً شامل 8 حالت می باشد) ، از دستورات شرطی تودرتو استفاده می کنیم و به ازای هر عمل جاروبرقی یک مقدار هزینه به متغیری با این نام نسبت می دهیم و در انتها میزان این هزینه نمایش داده می شود.

از آن جا که پیاده سازی این برنامه برای اجرای یک مرتبه ی عامل جاروبرقی در یک شرایط مشخص می باشد و ممکن است که ما بخواهیم این عملیات برای چند وضعیت مختلف از محیط صورت بپذیرد و میزان امتیاز میانگین آن نمایش داده شود ، می توانیم با قرار دادن این برنامه در یک حلقه ی for و دریافت وضعیت های جدید برای محیط ، مسئله را تکرار و حل نماییم و میزان امتیاز را با استفاده از هزینه ی کل محاسبه شده تقسیم بر تعداد چرخش حلقه محاسبه کنیم. همچنین می توانیم با استفاده از یک فایل ورودی وضعیت های مختلف محیط مسئله را دریافت نماییم و با توجه وضعیت های دریافت شده از فایل ، مسئله را تکرار و حل بنماییم و میانگین امتیاز را با استفاده از تقسیم هزینه بر تعداد خطوط فایل ورودی به دست بیاوریم.

```
#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 the Location of Vacuum:\n") #user_input
    of Location vacuum is placed
    status_input = input("Enter the Status of " + location_input + ":\n")
    #user_input if Location is Dirty or Clean
    status_input_complement = input("Enter the Status of other room:\n")
    print("Initial Location Condition is: " + str(goal_state))
    if location_input == 'A':
        current_state = {'A': status_input, 'B': status_input_complement}
        print("Current World State is: " + str(current_state))
        print("Vacuum is placed in Location A.")
        #Location A is Dirty.
        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 Location A is: " + 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 to the Location B is: " +
str(cost))
                #Suck the dirt and mark it as Clean
```

```

        goal_state['B'] = '0'
        cost += 1                                #Cost for Suck
        print("COST for CLEANING Location B is: " + str(cost))
        print("Location B has been Cleaned.")
    else:
        #Mark it as Clean
        print("Location B is already Clean.")
        print("No action, so Total COST is: " + str(cost))
if status_input == '0':
    #Mark it as Clean
    print("Location A is already Clean.")
    print("No action, so Total COST is: " + str(cost))
    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 is: " + str(cost))
        #Suck the dirt and mark it as Clean
        goal_state['B'] = '0'
        cost += 1                                #Cost for Suck
        print("Cost for CLEANING Location B is: " + str(cost))
        print("Location B has been Cleaned.")
    else:
        #Mark it as Clean
        print("Location B is already Clean.")
        print("No action, so Total COST is: " + str(cost))
else:
    current_state = {'A': status_input_complement, 'B': status_input}
    print("Current World State is: " + str(current_state))
    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 Location B is: " + 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 is: " + str(cost))
            #Suck the dirt and mark it as Clean
            goal_state['A'] = '0'
            cost += 1                                #Cost for Suck
            print("Cost for CLEANING Location A is: " + str(cost))
            print("Location A has been Cleaned.")
        else:
            # Mark it as Clean
            print("Location A is already Clean.")
            print("No action, so Total COST is: " + str(cost))
    else:
        #Mark it as Clean
        print("Location B is already Clean.")
        print("No action, so Total COST is: " + str(cost))
        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 is: " + str(cost))
    #Suck the dirt and mark it as Clean
    goal_state['A'] = '0'
    cost += 1                                #Cost for Suck
    print("Cost for CLEANING Location A is: " + str(cost))
    print("Location A has been Cleaned.")
else:
    #Mark it as Clean
    print("Location A is already Clean.")
    print("No action, so Total COST is: " + str(cost))
#Done Cleaning
print("GOAL STATE:")
print(goal_state)
print("Performance Measurement is: " + str(cost))

```

با استفاده از این بخش تابع را فراخوانی و اجرا می نماییم:

```

if __name__ == '__main__':
    vacuum_world()

```