

Abubakkar Abdullah

20P-0045

Lab-Task-2

Section BCS-6C

-----

Question 1:

```

In [*]: import time

NO_SMOKE           = 0x00
SMOKE              = 0x01
NO_HIGH_TEMPERATURE = 0x02
HIGH_TEMPERATURE   = 0x03
SPRINKER_ON        = 0x04
SPRINKER_OFF       = 0x05
ALARM_ON           = 0x06
ALARM_OFF          = 0x07
ALARM_FIRE_DEP_ON  = 0x08
ALARM_FIRE_DEP_OFF = 0x09

GOAL_STATE = { 0x01: NO_HIGH_TEMPERATURE, 0x02: NO_SMOKE }

GOAL_TEST = [
    { "0x01": NO_HIGH_TEMPERATURE, "0x02": SMOKE },
    { "0x01": NO_HIGH_TEMPERATURE, "0x02": NO_SMOKE },
    { "0x01": HIGH_TEMPERATURE, "0x02": SMOKE } ]

def action(state):
    ACTIVE_ACTION = []

    if (state["0x01"] == HIGH_TEMPERATURE): ACTIVE_ACTION += [ALARM_FIRE_DEP_ON, ALARM_ON, SPRINKER_ON]
    if (state["0x01"] == NO_HIGH_TEMPERATURE):
        if (state["0x02"] == SMOKE):
            ACTIVE_ACTION += [ALARM_FIRE_DEP_OFF, ALARM_ON, SPRINKER_ON]
        if (state["0x02"] == NO_SMOKE):
            ACTIVE_ACTION += [ALARM_FIRE_DEP_OFF, ALARM_OFF, SPRINKER_OFF]

    return ACTIVE_ACTION

def action_to_screen(actions):
    CALL_1, CALL_2, CALL_3 = '', '', ''

    if (actions[0] == ALARM_FIRE_DEP_ON): CALL_1 = 'action: [emergency] calling fire department'
    else: CALL_1 = 'action: no emergency'

    if (actions[1] == ALARM_ON): CALL_2 = 'action: alarm on'
    else: CALL_2 = 'action: alarm off'

    if (actions[2] == SPRINKER_ON): CALL_3 = 'action: sprinkler on'
    else: CALL_3 = 'action: sprinkler off'

    print(CALL_1, CALL_2, CALL_3, '\n', sep="\n")

```

System contains  
the,

- Alarm
- Temperature  
Detector
- Smoke Detector

- Sprinkling System
- Calling The FireDept

```
def action_to_screen(actions):  
    CALL_1, CALL_2, CALL_3 = '', '', ''  
  
    if (actions[0] == ALARM_FIRE_DEP_ON): CALL_1 = 'action: [emergency] calling fire department'  
    else: CALL_1 = 'action: no emergency'  
  
    if (actions[1] == ALARM_ON): CALL_2 = 'action: alarm on'  
    else: CALL_2 = 'action: alarm off'  
  
    if (actions[2] == SPRINKER_ON): CALL_3 = 'action: sprinkler on'  
    else: CALL_3 = 'action: sprinkler off'  
  
    print(CALL_1, CALL_2, CALL_3, '\n', sep="\n")  
  
def main():  
    while True:  
        for state in GOAL_TEST:  
            action_to_screen(action(state))  
            time.sleep(1)  
  
if __name__ == '__main__':  
    main()
```

•

•

Action appending  
on the screen.

# Main:

```
def main():  
    while True:  
        for state in GOAL_TEST:  
            action_to_screen(action(state))  
            time.sleep(1)  
  
if __name__ == '__main__':  
    main()
```

•

```
In [*]: import time  
  
NO_SMOKE           = 0x00  
SMOKE              = 0x01  
NO_HIGH_TEMPERATURE = 0x02  
HIGH_TEMPERATURE   = 0x03  
SPRINKER_ON        = 0x04  
SPRINKER_OFF       = 0x05  
ALARM_ON            = 0x06  
ALARM_OFF           = 0x07  
ALARM_FIRE_DEP_ON  = 0x08  
ALARM_FIRE_DEP_OFF = 0x09  
  
GOAL_STATE         = { 0x01: NO_HIGH_TEMPERATURE, 0x02: NO_SMOKE }  
  
GOAL_TEST          = [  
    { "0x01": NO_HIGH_TEMPERATURE, "0x02": SMOKE },  
    { "0x01": NO_HIGH_TEMPERATURE, "0x02": NO_SMOKE },  
    { "0x01": HIGH_TEMPERATURE, "0x02": SMOKE }]
```

# Working:

•

```
action: no emergency  
action: alarm on  
action: sprinkler on
```

•

```
action: no emergency  
action: alarm off  
action: sprinkler off
```

•

```
action: [emergency] calling fire department  
action: alarm on  
action: sprinkler on
```

•

```
action: no emergency  
action: alarm on  
action: sprinkler on
```

•

```
action: no emergency  
action: alarm off  
action: sprinkler off
```

# Question 2:

```
In [*]: import random
import pandas as pd
def Automatic_watering_system():
    userdata = {
        "username": ['Mahad'],
        "password": ['Alpha009']
    }
    cost = 0
    df = pd.DataFrame(userdata)
    status = random.randint(0, 2)
    print("Welcome To Automatic Watering System Please Provide Your Login Details")
    user = input('Username : ')
    pasw = input('Password : ')

    def deactivating_watering_System():
        print("Deactivating the Watering System")

    def stoping_watering_System():
        print("Watering System is off stopped watering the plants")

    def Watering_System():
        print("Watering System Activated Watering the Plants")

    matching_creds = (len(df[(df.username == user) & (df.password == pasw)]) > 0)
    if matching_creds:
        if status == 0:
            print("Moisture Level in the soil is Dry")
            print("Activating Watering System")
            cost += 1
            Watering_System()
            cost += 1
            print(f"Watered the Plants. Cost: {cost}")

        elif status == 1:
            print("Moisture Level in the soil is Moist")
            print("Turning Watering System Off")
            cost += 1
            stoping_watering_System()
            cost += 1
            print(f"Soil is Moist Plants are already Watered System off Cost: {cost}")

        elif status == 2:
            print("Moisture Level in the soil is Wet")
            print("Deactivating Watering System ")
            cost += 1
            deactivating_watering_System()
            cost += 1
            print(f"Soil is Wet Watering System is deactivated to prevent waterlogging. Cost: {cost}")
        else:
            print("Watering Sensor off")
            print("Performance measurement:", cost)

    else:
        print('\nYour account is not registered yet!')
        print('please contact admin')
    Automatic_watering_system()
```

END