Lab Task: 02

————————————786——————————————

Name: Muhammad Sherjeel Akhtar

Roll No: 20p-0101

Subject: Artificial Intelligence Lab

Submitted To Respected
Ma’am: Hurmat Hidayat

Section: BCS-6C

Question 1:

Imagine a single-floor office building with a fire alarm system that is controlled by a simple reflex agent. The system has smoke detectors and temperature sensors placed throughout the building to detect any signs of fire.
The agent's rules are as follows:

The goal of the agent is to keep the building and its occupants safe by quickly and efficiently responding to any signs of fire. Write a program to develop a simple reflex Agent.

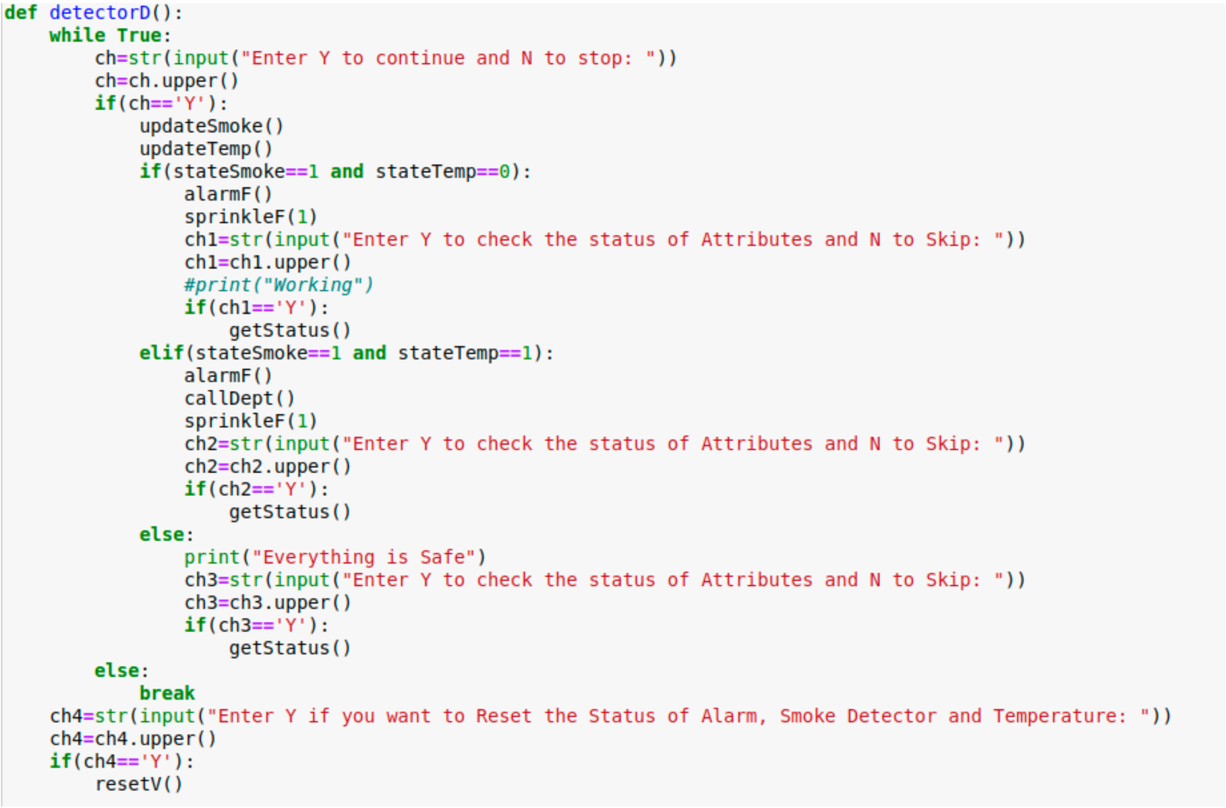
- if smoke is detected, the alarm will sound and the sprinkler system will activate to put out the fire;
- if a high temperature is detected, the alarm will sound and the fire department will be called.
- If neither smoke nor high temperature are detected, the system remains in its normal state with the alarm off and the sprinkler system deactivated.

Answer:

Code Overview:

```
def getSmoke(valve):
    if(valve==1):
        return 1
    else:
        return 0
def getTemp(valve):
    if(valve>40):
        return 1
    else:
        return 0
def updateSmoke():
    valve=int(input("Enter smoke value: "))
    global stateSmoke
    stateSmoke=getSmoke(valve)
    print(stateSmoke)
def updateTemp():
    valve=int(input("Enter temp value: "))
    global stateTemp
    stateTemp=getTemp(valve)
    print(stateTemp)
    print("\t")
def sprinkleF(var):
    if(var==1):
        print("Sprinkling the water. ")
        sprinkleVar=var
        return sprinkleVar
def alarmF():
    print("Alarm is Triggered.")
    alarmVar=0
    return alarmVar
def callDept():
    print("Call to fire department is successfully made.")
def getStatus():
    print('\n')
    print("Status")
    if(stateSmoke==1):
        print("Smoke is on =","stateSmoke: ",stateSmoke)
        getAlarm=1
        print("alarmVar: ",getAlarm)
        getSprinkle=sprinkleF(1)
        print("sprinkleVar: ",getSprinkle)
    elif(stateTemp==1):
        print("High Temperature Detected =","stateTemp: ",stateTemp)
    elif(stateTemp==0 and stateSmoke==0):
        getAlarm=alarmF()
        print("There is no Smoke and Temperature ||","stateSmoke: ",stateSmoke,"stateTemp: ",stateTemp, "Alarm is off =","alarmvar: ",getAl
def resetV():
    stateTemp=0
    stateSmoke=0
    alarmVar=0
    sprinkleVar=0
    print("\nstateSmoke: ",stateSmoke)
    print("stateTemp: ",stateTemp)
    print("alarmVar: ",alarmVar)
    print("sprinkleVar: ",sprinkleVar)
def detectorD():
    while True:
        ch=str(input("Enter Y to continue and N to stop: "))
        ch=ch.upper()
        if(ch=='Y'):
            updateSmoke()
            updateTemp()
            if(stateSmoke==1 and stateTemp==0):
                alarmF()
                sprinkleF(1)
                ch1=str(input("Enter Y to check the status of Attributes and N to Skip: "))
                ch1=ch1.upper()
                if(ch1=='Y'):
                    getStatus()
            elif(stateSmoke==1 and stateTemp==1):
                alarmF()
                callDept()
                sprinkleF(1)
                ch2=str(input("Enter Y to check the status of Attributes and N to Skip: "))
                ch2=ch2.upper()
                if(ch2=='Y'):
                    getStatus()
            else:
                print("Everything is Safe")
                ch3=str(input("Enter Y to check the status of Attributes and N to Skip: "))
                ch3=ch3.upper()
                if(ch3=='Y'):
                    getStatus()
        else:
            break
    ch4=str(input("Enter Y if you want to Reset the Status of Alarm, Smoke Detector and Temperature: "))
    ch4=ch4.upper()
    if(ch4=='Y'):
        resetV()




stateSmoke=0
stateTemp=0
detectorD()
```

MainFunction() Overview:



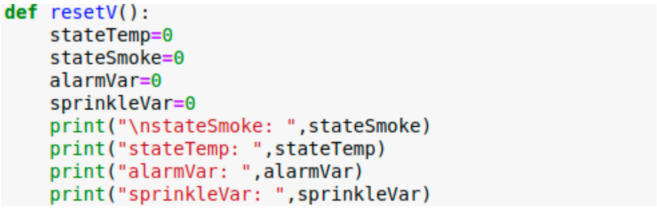
```
def detectorD():
    while True:
        ch=str(input("Enter Y to continue and N to stop: "))
        ch=ch.upper()
        if(ch=='Y'):
            updateSmoke()
            updateTemp()
            if(stateSmoke==1 and stateTemp==0):
                alarmF()
                sprinkleF(1)
                ch1=str(input("Enter Y to check the status of Attributes and N to Skip: "))
                ch1=ch1.upper()
                #print("Working")
                if(ch1=='Y'):
                    getStatus()
            elif(stateSmoke==1 and stateTemp==1):
                alarmF()
                callDept()
                sprinkleF(1)
                ch2=str(input("Enter Y to check the status of Attributes and N to Skip: "))
                ch2=ch2.upper()
                if(ch2=='Y'):
                    getStatus()
            else:
                print("Everything is Safe")
                ch3=str(input("Enter Y to check the status of Attributes and N to Skip: "))
                ch3=ch3.upper()
                if(ch3=='Y'):
                    getStatus()
        else:
            break
    ch4=str(input("Enter Y if you want to Reset the Status of Alarm, Smoke Detector and Temperature: "))
    ch4=ch4.upper()
    if(ch4=='Y'):
        resetV()
```

ParameterReseting:

I’ve created a Function with a name “**resetV()”.** It will reset all the attributes including:

- Smoke State
- Temperature State
- Alarm State
- Sprinkler System State

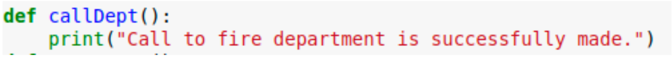
to zero.



```
def resetV():
    stateTemp=0
    stateSmoke=0
    alarmVar=0
    sprinkleVar=0
    print("\nstateSmoke: ",stateSmoke)
    print("stateTemp: ",stateTemp)
    print("alarmVar: ",alarmVar)
    print("sprinkleVar: ",sprinkleVar)
```

Fire Department Call Function:

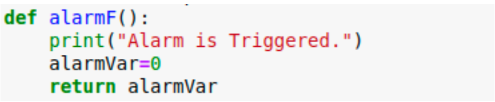
“callDept()” function will make a call to the fire Department.



```
def callDept():
    print("Call to fire department is successfully made.")
```

Triggering The Alarm Function:

“alarmF()” function will trigger the Alarm on call and will change its state to active state.
”***state=1”***

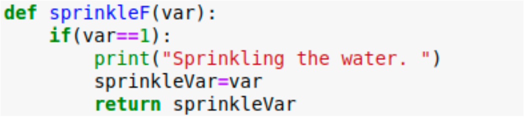


```
def alarmF():
    print("Alarm is Triggered.")
    alarmVar=0
    return alarmVar
```

Activating The Water Sprinkle System:

“sprinkleF()” function will activate the **Water Sprinkling System** and will change it’s state to active state.

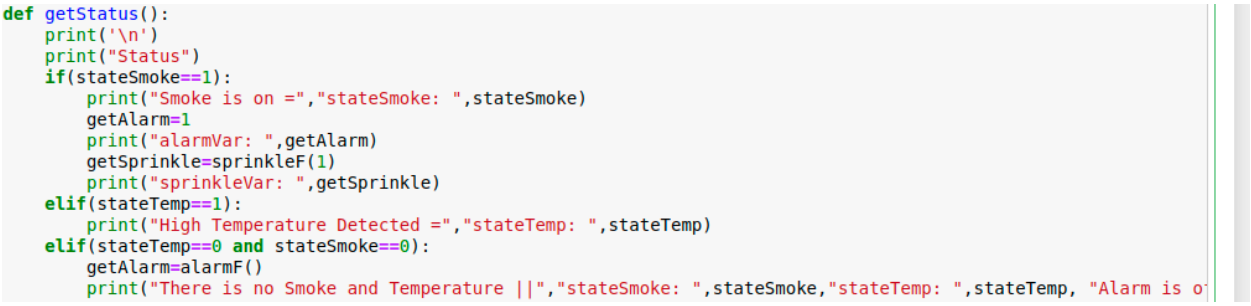
“state=1”



```
def sprinkleF(var):
    if(var==1):
        print("Sprinkling the water. ")
        sprinkleVar=var
        return sprinkleVar
```

Status Function:

“getStatus” function will show the status of the **Activate and Offline** systems.
At the end of main function, user will be asked if he/she wants to check the status of the Systems or not. On the call of yes, this function will be Called.



```
def getStatus():
    print('\n')
    print("Status")
    if(stateSmoke==1):
        print("Smoke is on =","stateSmoke: ",stateSmoke)
        getAlarm=1
        print("alarmVar: ",getAlarm)
        getSprinkle=sprinkleF(1)
        print("sprinkleVar: ",getSprinkle)
    elif(stateTemp==1):
        print("High Temperature Detected =","stateTemp: ",stateTemp)
    elif(stateTemp==0 and stateSmoke==0):
        getAlarm=alarmF()
        print("There is no Smoke and Temperature ||","stateSmoke: ",stateSmoke,"stateTemp: ",stateTemp, "Alarm is o
```

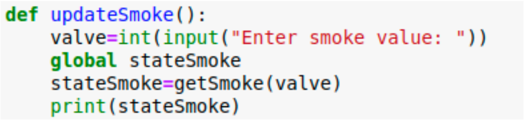
Smoke Status Function:

“updateSmoke()” function will activate the system on ***Smoke Detection.***

***”*stateSmoke”** variable will be set to **1** on smoke detection.

This function works by calling a second function **getSmoke().**

After getting the state from the getSmoke() function, our system will get activated or it will stay ideal if the conditions are normal.



```
def updateSmoke():
    valve=int(input("Enter smoke value: "))
    global stateSmoke
    stateSmoke=getSmoke(valve)
    print(stateSmoke)
```

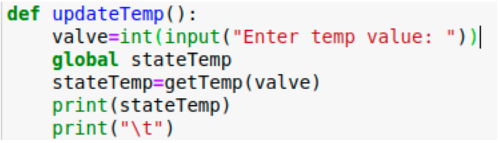
Temperature Status Function:

“**updateTemp()”** function will activate the system on **High Temperature Detection.**

”stateTemp” variable will be set to **1** on high temperature detection.

This fumction works by calling a second function **“getTemp()”.**

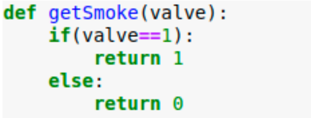
After getting the state from the getTemp() function, our system will get activated or it will stay ideal if the temperatures are normal.



```
def updateTemp():
    valve=int(input("Enter temp value: "))
    global stateTemp
    stateTemp=getTemp(valve)
    print(stateTemp)
    print("\t")
```

Smoke Detected:

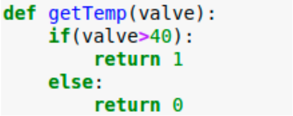
getSmoke() function will **return 1** to **updateSmoke()** function if smoke is detected and **0** if smoke is not detected.



```
def getSmoke(valve):
    if(valve==1):
        return 1
    else:
        return 0
```

Temperature Detected:

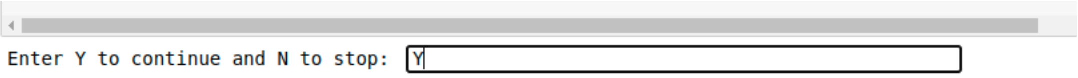
getTemp() function will **return 1** to **updateTemp()** function if temperature is above 40 Degree Celcius and **0** if Temperature is below 40 Degree.



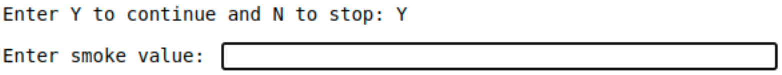
```
def getTemp(valve):
    if(valve>40):
        return 1
    else:
        return 0
```

Working:

Initial Popup whether you want to start system or not.



```
Enter Y to continue and N to stop: Y
```



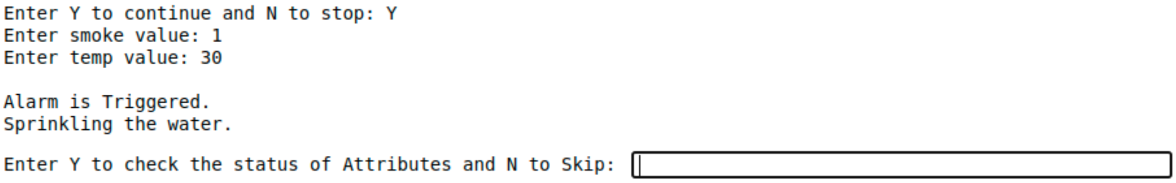
```
Enter Y to continue and N to stop: Y

Enter smoke value:
```

If smoke is detected, **press 1** if there is no smoke, **Enter 0**.

Let suppose the Temperature is 30 Degree:

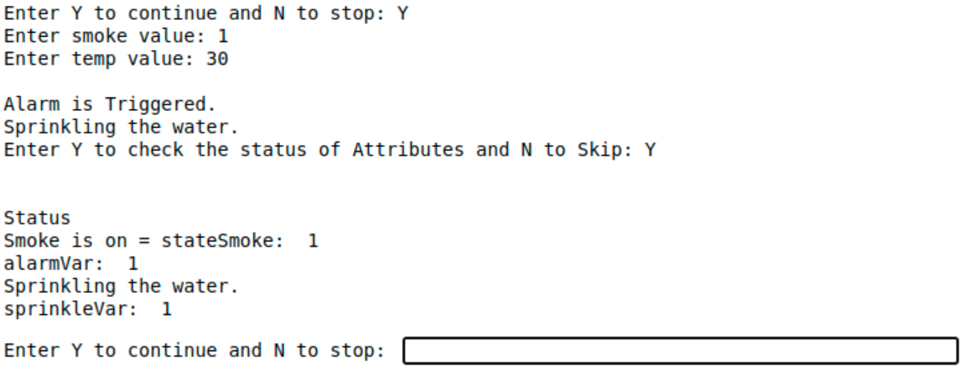
As the **Smoke** is detected but the temperature is **low,** so

- Alarm will be triggered
- Sprinkling System will turned on



```
Enter Y to continue and N to stop: Y
Enter smoke value: 1
Enter temp value: 30

Alarm is Triggered.
Sprinkling the water.

Enter Y to check the status of Attributes and N to Skip:
```

Press Y if you want to check the status of activated systems.

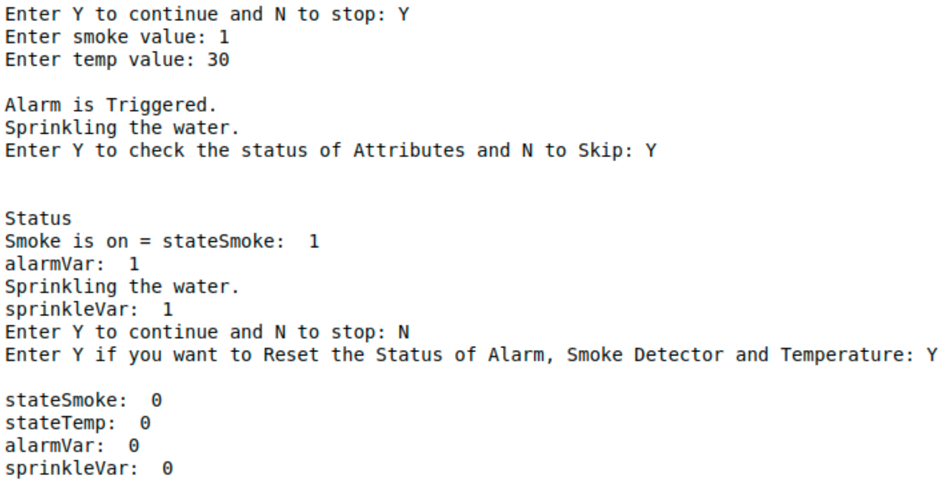


```
Enter Y to continue and N to stop: Y
Enter smoke value: 1
Enter temp value: 30

Alarm is Triggered.
Sprinkling the water.
Enter Y to check the status of Attributes and N to Skip: Y


Status
Smoke is on = stateSmoke:  1
alarmVar:  1
Sprinkling the water.
sprinkleVar:  1

Enter Y to continue and N to stop:
```

Press Y if you want continue testing with further values.

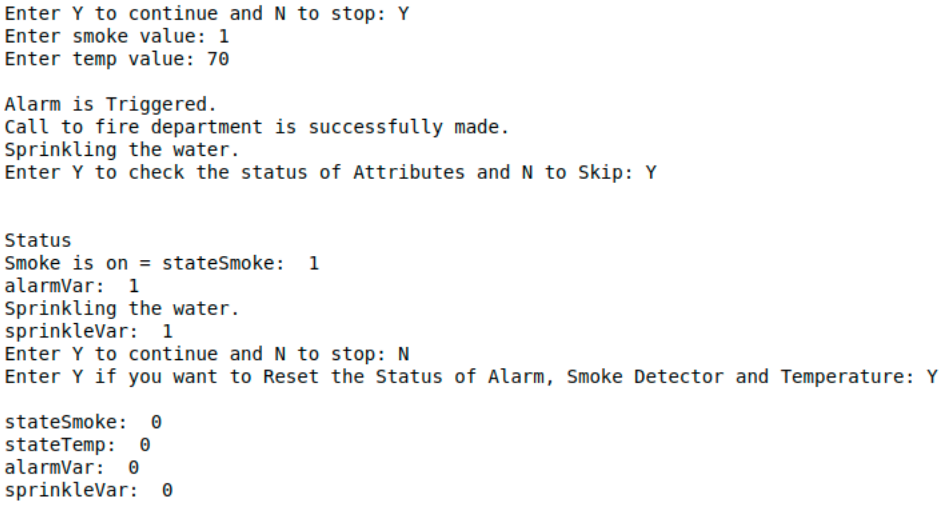
Setting the system back to normal:

If you want to set all the Fire, Smoke, Alarm Systems to normal **press Y** at the End.



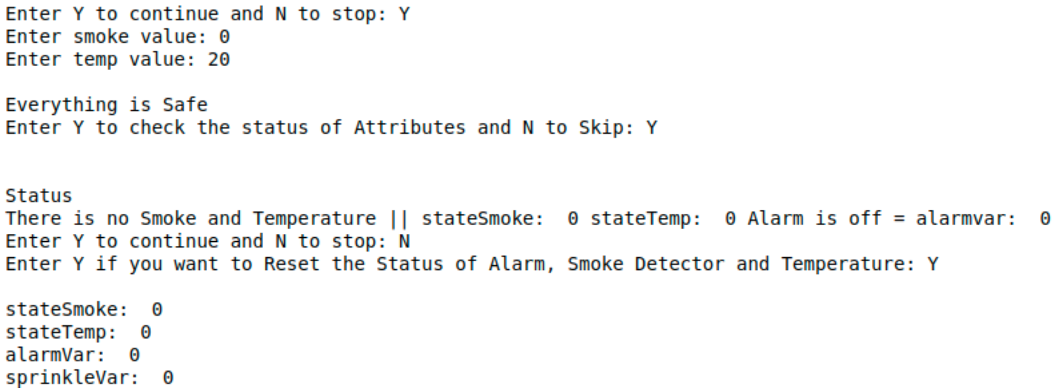
```
Enter Y to continue and N to stop: Y
Enter smoke value: 1
Enter temp value: 30

Alarm is Triggered.
Sprinkling the water.
Enter Y to check the status of Attributes and N to Skip: Y


Status
Smoke is on = stateSmoke:  1
alarmVar:  1
Sprinkling the water.
sprinkleVar:  1
Enter Y to continue and N to stop: N
Enter Y if you want to Reset the Status of Alarm, Smoke Detector and Temperature: Y

stateSmoke:  0
stateTemp:  0
alarmVar:  0
sprinkleVar:  0
```

For High Temperature:



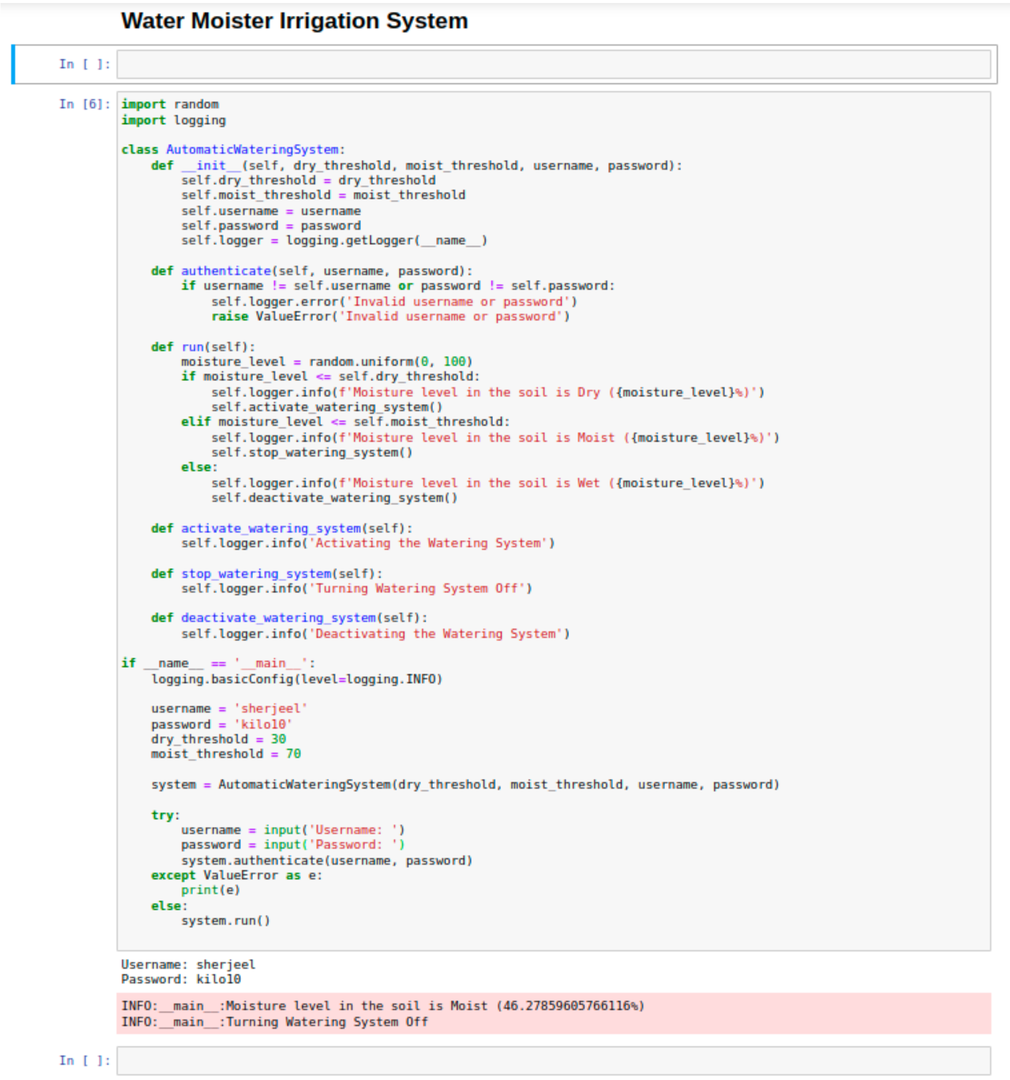
```
Enter Y to continue and N to stop: Y
Enter smoke value: 1
Enter temp value: 70

Alarm is Triggered.
Call to fire department is successfully made.
Sprinkling the water.
Enter Y to check the status of Attributes and N to Skip: Y


Status
Smoke is on = stateSmoke:  1
alarmVar:  1
Sprinkling the water.
sprinkleVar:  1
Enter Y to continue and N to stop: N
Enter Y if you want to Reset the Status of Alarm, Smoke Detector and Temperature: Y

stateSmoke:  0
stateTemp:  0
alarmVar:  0
sprinkleVar:  0
```

For Normal Circumstances:



```
Enter Y to continue and N to stop: Y
Enter smoke value: 0
Enter temp value: 20

Everything is Safe
Enter Y to check the status of Attributes and N to Skip: Y


Status
There is no Smoke and Temperature || stateSmoke:  0 stateTemp:  0 Alarm is off = alarmvar:  0
Enter Y to continue and N to stop: N
Enter Y if you want to Reset the Status of Alarm, Smoke Detector and Temperature: Y

stateSmoke:  0
stateTemp:  0
alarmVar:  0
sprinkleVar:  0
```

Question 2:



Water Moister Irrigation System

```
import random
import logging

class AutomaticWateringSystem:
    def __init__(self, dry_threshold, moist_threshold, username, password):
        self.dry_threshold = dry_threshold
        self.moist_threshold = moist_threshold
        self.username = username
        self.password = password
        self.logger = logging.getLogger(__name__)

    def authenticate(self, username, password):
        if username != self.username or password != self.password:
            self.logger.error('Invalid username or password')
            raise ValueError('Invalid username or password')

    def run(self):
        moisture_level = random.uniform(0, 100)
        if moisture_level <= self.dry_threshold:
            self.logger.info(f'Moisture level in the soil is Dry ({moisture_level}%)')
            self.activate_watering_system()
        elif moisture_level <= self.moist_threshold:
            self.logger.info(f'Moisture level in the soil is Moist ({moisture_level}%)')
            self.stop_watering_system()
        else:
            self.logger.info(f'Moisture level in the soil is Wet ({moisture_level}%)')
            self.deactivate_watering_system()

    def activate_watering_system(self):
        self.logger.info('Activating the Watering System')

    def stop_watering_system(self):
        self.logger.info('Turning Watering System Off')

    def deactivate_watering_system(self):
        self.logger.info('Deactivating the Watering System')

if __name__ == '__main__':
    logging.basicConfig(level=logging.INFO)

    username = 'sherjeel'
    password = 'kilol0'
    dry_threshold = 30
    moist_threshold = 70

    system = AutomaticWateringSystem(dry_threshold, moist_threshold, username, password)

    try:
        username = input('Username: ')
        password = input('Password: ')
        system.authenticate(username, password)
    except ValueError as e:
        print(e)
    else:
        system.run()
```

```
Username: sherjeel
Password: kilol0
```

```
INFO:__main__:Moisture level in the soil is Moist (46.27859605766116%)
INFO:__main__:Turning Watering System Off
```

FIN!!