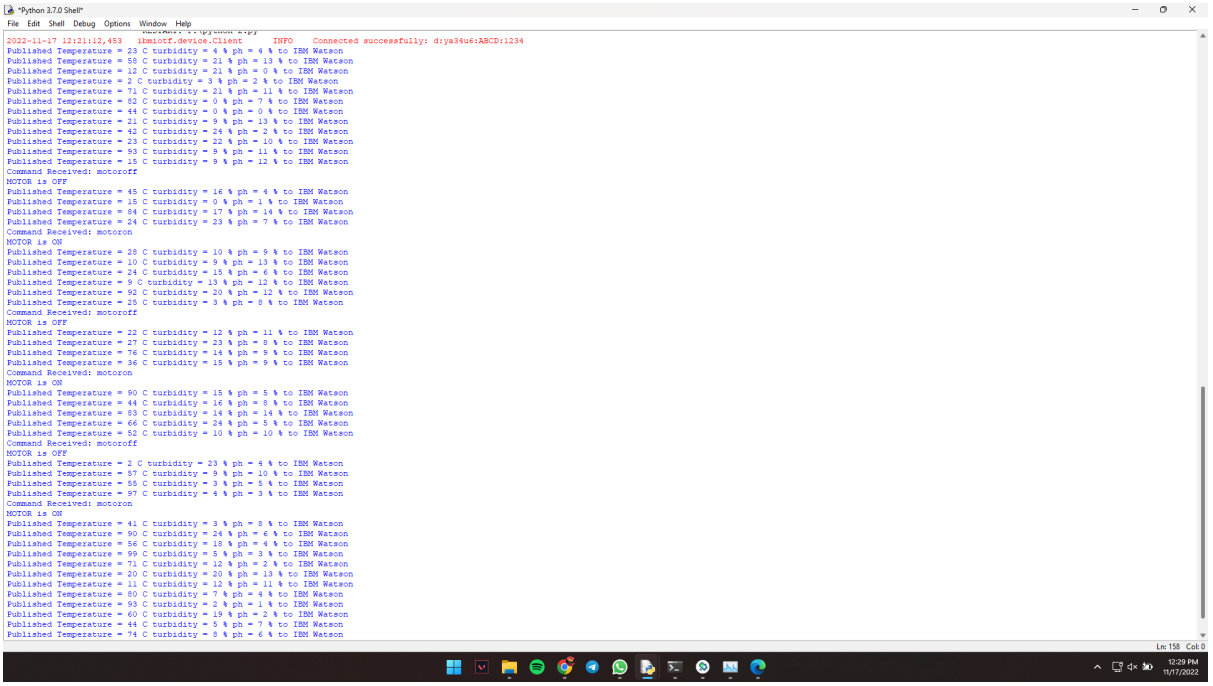
PYTHON SCRIPT TO MONITOR TEMPERATURE, PH, TURBIDITY IN RIVER WATER

PROGRAM:

```
import time
import sys
import ibmiotf.application
import ibmiotf.device
import random

#vishnu IBM
organization = "ya34u6"
deviceType = "ABCD"
deviceId = "1234"
authMethod = "token"
authToken = "12345678"

#Gpio

def mycommandCallback(cmd):
    print("Command Received: %s" %cmd.data['command'])
    status = cmd.data['command']
    if status=="motoron":
        print("MOTOR is ON")
    elif status=="motoroff":
        print("MOTOR is OFF")
    else:
        print("please send proper command")
try:
    deviceOptions =
{"org":organization,"type":deviceType,"id":deviceId,"auth-method":authMethod,"auth
-token":authToken}
    deviceCli = ibmiotf.device.Client(deviceOptions)

except Exception as e:
    print("Caught exception connecting device: %s" %str(e))
    sys.exit()

#CONNECT
deviceCli.connect()

while True:
    temperature=random.randint(0,100)
```

```
    turbidity=random.randint(0,25)
    ph=random.randint(0,14)

    data={'temperature':temperature,'turbidity':turbidity,'ph':ph}

    def myOnPublishCallback():
        print("Published Temperature = %s C"%temperature,"turbidity = %s %%"
%turbidity,"ph = %s %%" %ph, "to IBM Watson")

    success = deviceCli.publishEvent("IoTSensor","json",data,qos=0,
on_publish=myOnPublishCallback)
    if not success:
        print("Not connected to IoTF")
    time.sleep(10)

    deviceCli.commandCallback = mycommandCallback

#Disconnect

deviceCli.disconnect()
```

OUTPUT :



PUBLISHING DATA TO IBM CLOUD :