

# IBM

## ASSIGNMENT 3

**TEAM ID: PNT2022TMID05356**

TASK 1: Write a python code for blinking LED using python

Code:

```
from time import sleep
import RPi.GPIO as
GPIO
GPIO.setmode(GPIO.B
CM)
GPIO.setup(17,GPIO.O
UT)
GPIO.setup(27,GPIO.O
UT)
print ("lights on")
GPIO.output(17,GPIO.HI
GH)
GPIO.output(27,GPIO.HI
GH)
sleep(1)
print ("lights off")
GPIO.output(17,GPIO.L
OW)
GPIO.output(27,GPIO.L
OW)
sleep(1)
print ("lights on")
GPIO.output(17,GPIO.HI
GH)
GPIO.output(27,GPIO.HI
GH)
sleep(1)
print( "lights off")
```

```
GPIO.output(17,GPIO.L  
OW)
```

```
GPIO.output(27,GPIO.L  
OW)
```

```
GPIO.cleanup()
```

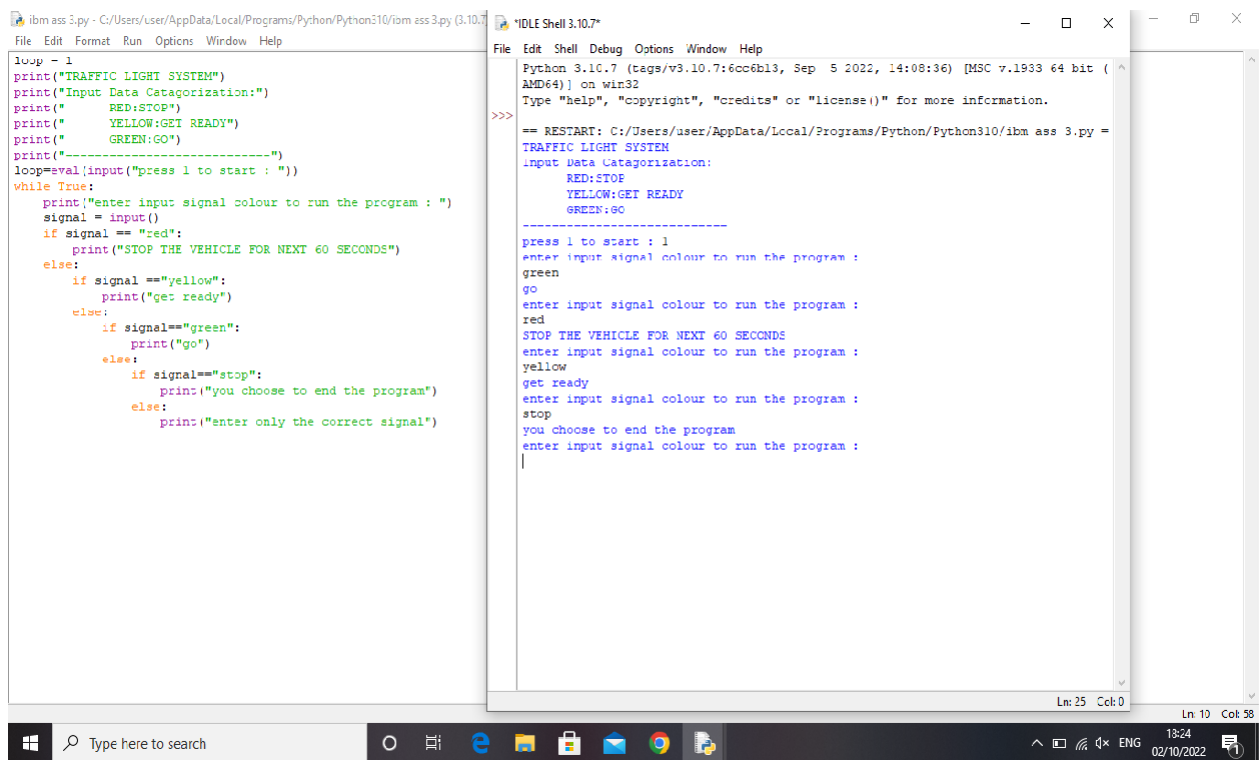
## Output:

```
pi@raspberrypi:~ $ cd gpio_python_code/
pi@raspberrypi:~/gpio_python_code $ touch 3_blink.py
pi@raspberrypi:~/gpio_python_code $ touch 3_blink_forever.py
pi@raspberrypi:~/gpio_python_code $ nano 3_blink.py
pi@raspberrypi:~/gpio_python_code $ nano 3_blink_forever.py
pi@raspberrypi:~/gpio_python_code $ sudo python 3_blink.py
3_blink.py:10: RuntimeWarning: This channel is already in use, continuing anyway
. Use GPIO.setwarnings(False) to disable warnings.
  GPIO.setup(17,GPIO.OUT)
3_blink.py:11: RuntimeWarning: This channel is already in use, continuing anyway
. Use GPIO.setwarnings(False) to disable warnings.
  GPIO.setup(27,GPIO.OUT)
lights on
lights off
lights on
lights off
pi@raspberrypi:~/gpio_python_code $
```

Write a python code for traffic light system using python(should be communicatable with raspberrypi)

```
loop =  
    1  
  
print("TRAFFIC LIGHT SYSTEM")  
  
print("Input Data  
Catagorization:") print("  
        RED:STOP")  
  
print("    YELLOW:GET  
READY") print("  
        GREEN:GO")  
  
print(" .....-")  
  
loop=eval(input("press 1 to start :  
")) while True:  
  
    print("enter input signal colour to run the program  
: ") signal = input()  
  
    if signal == "red":  
  
        print("STOP THE VEHICLE FOR NEXT 60 SECONDS")  
  
    else:  
  
        if signal=="yellow":  
  
            print("get  
ready") else:  
  
                if signal=="green":  
  
                    print("go")  
  
                else:  
  
                    if signal=="stop":  
  
                        print("you choose to end the  
program") else:  
  
                            print("enter only the correct signal")
```

## output:



The screenshot shows an IDE with two windows. The left window displays a Python script for a traffic light system. The right window shows the output of the script in a shell, including the program's title, version information, and a series of prompts and responses for starting and running the program.

```
ibm ass 3.py - C:/Users/user/AppData/Local/Programs/Python/Python310/ibm ass 3.py (3.10.7)
File Edit Format Run Options Window Help

loop = 1
print("TRAFFIC LIGHT SYSTEM")
print("Input Data Categorization:")
print("    RED:STOP")
print("    YELLOW:GET READY")
print("    GREEN:GO")
print("-----")
loop=eval(input("press 1 to start : "))
while True:
    print("enter input signal colour to run the program : ")
    signal = input()
    if signal == "red":
        print("STOP THE VEHICLE FOR NEXT 60 SECONDS")
    else:
        if signal == "yellow":
            print("get ready")
        else:
            if signal == "green":
                print("go")
            else:
                if signal == "stop":
                    print("you choose to end the program")
                else:
                    print("enter only the correct signal")

IDLE Shell 3.10.7*
Python 3.10.7 (tags/v3.10.7:6cc6b13, Sep 5 2022, 14:08:36) [MSC v.1933 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
== RESTART: C:/Users/user/AppData/Local/Programs/Python/Python310/ibm ass 3.py ==
TRAFFIC LIGHT SYSTEM
Input Data Categorization:
RED:STOP
YELLOW:GET READY
GREEN:GO
-----
press 1 to start : 1
enter input signal colour to run the program :
green
go
enter input signal colour to run the program :
red
STOP THE VEHICLE FOR NEXT 60 SECONDS
enter input signal colour to run the program :
yellow
get ready
enter input signal colour to run the program :
stop
you choose to end the program
enter input signal colour to run the program :
|
```

## Using RPi library:

import RPi.GPIO as GPIO

import time

import signal

import sys

GPIO.setmode(GPIO.B

CM) GPIO.setup(9,

GPIO.OUT)

GPIO.setup(10,

GPIO.OUT)

GPIO.setup(11,

GPIO.OUT)

def allLightsOff(signal,

frame): GPIO.output(9,

False) GPIO.output(10,

False) GPIO.output(11,

False) GPIO.cleanup()

sys.exit(0)

signal.signal(signal.SIGINT,

allLightsOff)

```
while True:
    GPIO.output(9,
    True) time.sleep(3)
    GPIO.output(10,
    True) time.sleep(1)
    GPIO.output(9,
    False)
    GPIO.output(10,
    False)
    GPIO.output(11,
    True) time.sleep(5)
    GPIO.output(11,
    False)
    GPIO.output(10,
    True) time.sleep(2)
    GPIO.output(10,
    False)
```