Program 1: Run some python programs on Pi like: Read your name and print Hello message with name Read two numbers and print their sum, difference, product and division. Word and character count of a given string Area of a given shape (rectangle, triangle and circle) reading shape and appropriate values from standard input Print a name ‘n’ time, where name and n are read from standard input, using for and while loops. Handle Divided by Zero Exception. Print current time for 10 times with an interval of 10 seconds. Read a file line by line and print the word count of each line.

1a.py:

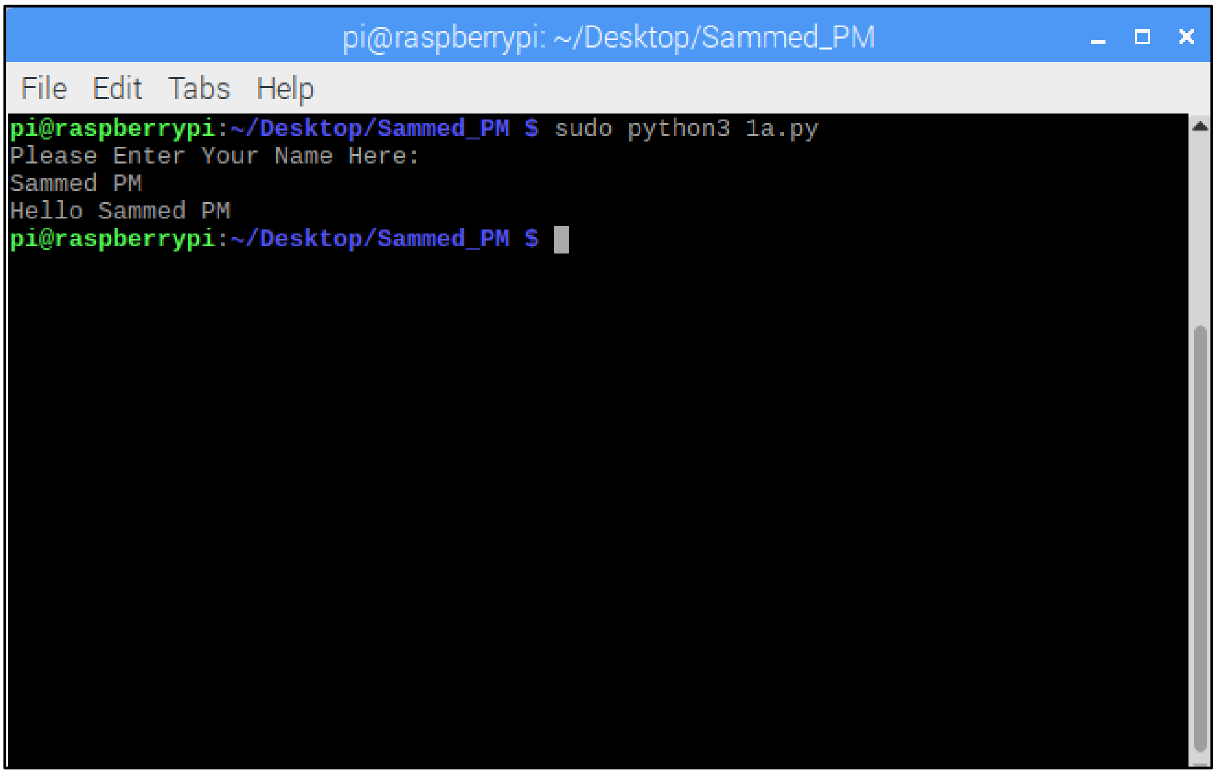
```
name = input("Please Enter Your Name Here:\n")
print("Hello " + name)
```

Procedure:

Run the program as: **sudo python3 1a.py**

It prompts for the name type your name.

Output:



```
pi@raspberrypi:~/Desktop/Sammed_PM $ sudo python3 1a.py
Please Enter Your Name Here:
Sammed PM
Hello Sammed PM
pi@raspberrypi:~/Desktop/Sammed_PM $ 
```

1b.py:

```
print("Enter First Number")

num1 = input()

print("Enter Second Number")

num2 = input()

print("************************************")

print("Sum of Given numbers is : "+str(int(num1)+int(num2)))

print("Difference of Given numbers is : "+str(int(num1)-int(num2)))

print("Product of Given numbers is : "+str(int(num1)*int(num2)))

print("Division of Given numbers is : "+str(int(num1)/int(num2)))

print("************************************")
```

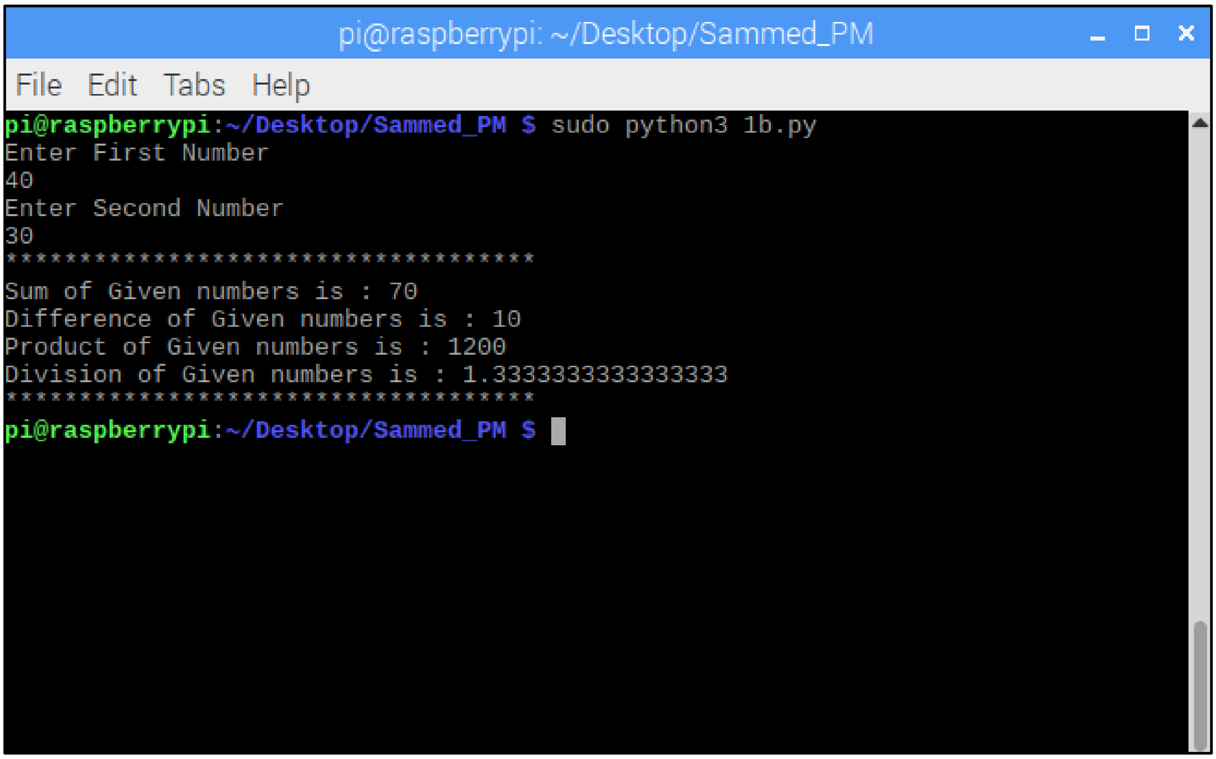
Procedure:

Read 2 numbers & print their Sum, Difference, Product & Division.

Run the program as: **sudo python3 1b.py**

It prompts to enter 2 numbers.

Output:



```
pi@raspberrypi:~/Desktop/Sammed_PM $ sudo python3 1b.py
Enter First Number
40
Enter Second Number
30
************************************
Sum of Given numbers is : 70
Difference of Given numbers is : 10
Product of Given numbers is : 1200
Division of Given numbers is : 1.3333333333333333
************************************
pi@raspberrypi:~/Desktop/Sammed_PM $
```

1c.py:

```
print("Enter a sentence ")
sentence = input()
words = sentence.split()
word_count = 0
character_count = 0
for word in words:
        word_count += 1
        character_count += len(word)
print("Total Numbers of Words in the sentence are : ",word_count)
print("Total Numbers of characters in the sentence excluding spaces are : ",character_count)
print("Total Numbers of characters in the sentence including spaces are :
",character_count+word_count-1)
```

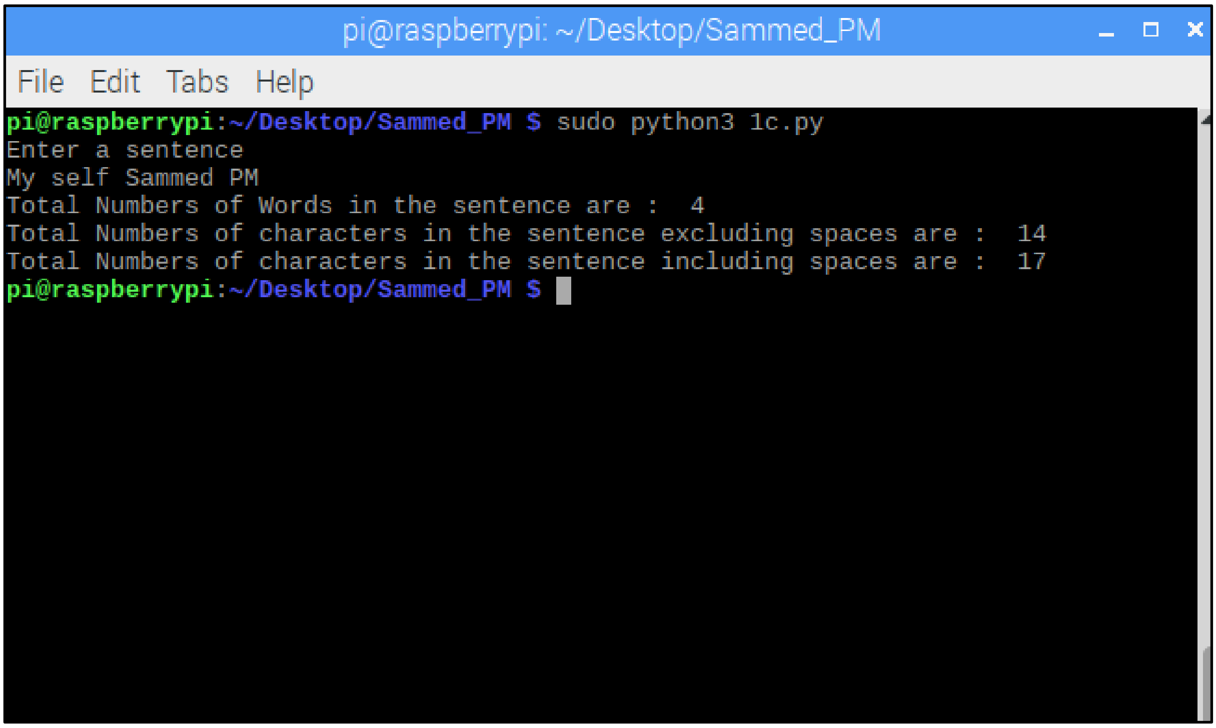
Procedure:

Word & Character count of a given string.

Run the program as: **sudo python3 1c.py**

It prompts to type a sentence, the typed sentence is “Sum of Given numbers is: 40”

Output:



```
pi@raspberrypi:~/Desktop/Sammed_PM $ sudo python3 1c.py
Enter a sentence
My self Sammed PM
Total Numbers of Words in the sentence are :  4
Total Numbers of characters in the sentence excluding spaces are :  14
Total Numbers of characters in the sentence including spaces are :  17
pi@raspberrypi:~/Desktop/Sammed_PM $ 
```

1d.py:

```
while True:
        print("****************************************")
        print("Select the Shape that you want to calculate Area")
        print("""                    1. Rectangle
                2. Triangle
                3. Circle
                4. Exit """)
        choice = input()
        if(choice == '1'):
                print("Enter the Width of the Rectangle in meters")
                width = int(input())
                print("Enter the height of the Rectangle in meters")
                height = int(input())
                print("The area of a Given Rectangle is ", width*height , " square meters ")
                continue
        elif(choice == '2'):
                print("Enter the Base value of the Triangle in meters")
                base = int(input())
                print("Enter the height of the Triangle in meters")
                height = int(input())
                print("The area of a Given Rectangle is ", 0.5*base*height , " square meters ")
                continue
        elif(choice == '3'):
                print("Enter the Radius of the Circle in meters")
                radius = int(input())
                print("The area of a Given Circle is ", 3.14*radius*radius , " square meters ")
                continue
        elif(choice == '4'):
                break
        else:
                print("Please enter a valid number from the menu")
                continue
```

print("Thank You")

Procedure:

Area of a given shape (Rectangle, Triangle & Circle) reading shape & appropriate value from standard I/p.

Run the program as: **sudo python3 3 1d.py**

It will prompt as below

Select the Shape that you want to calculate Area

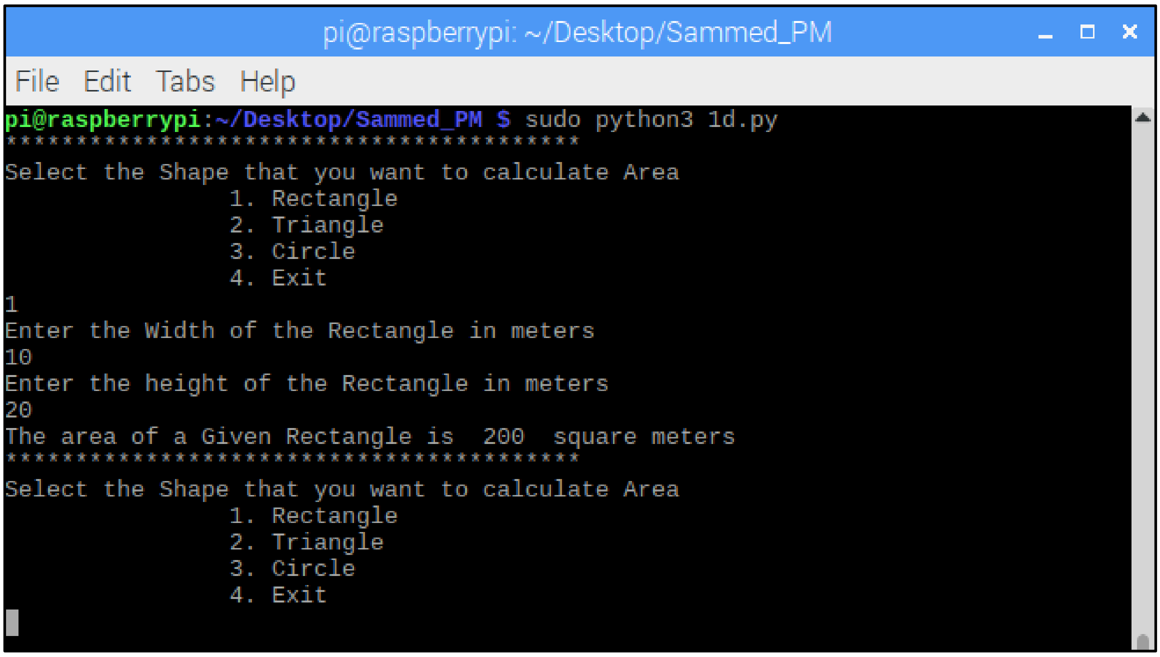
1. Rectangle

2. Triangle

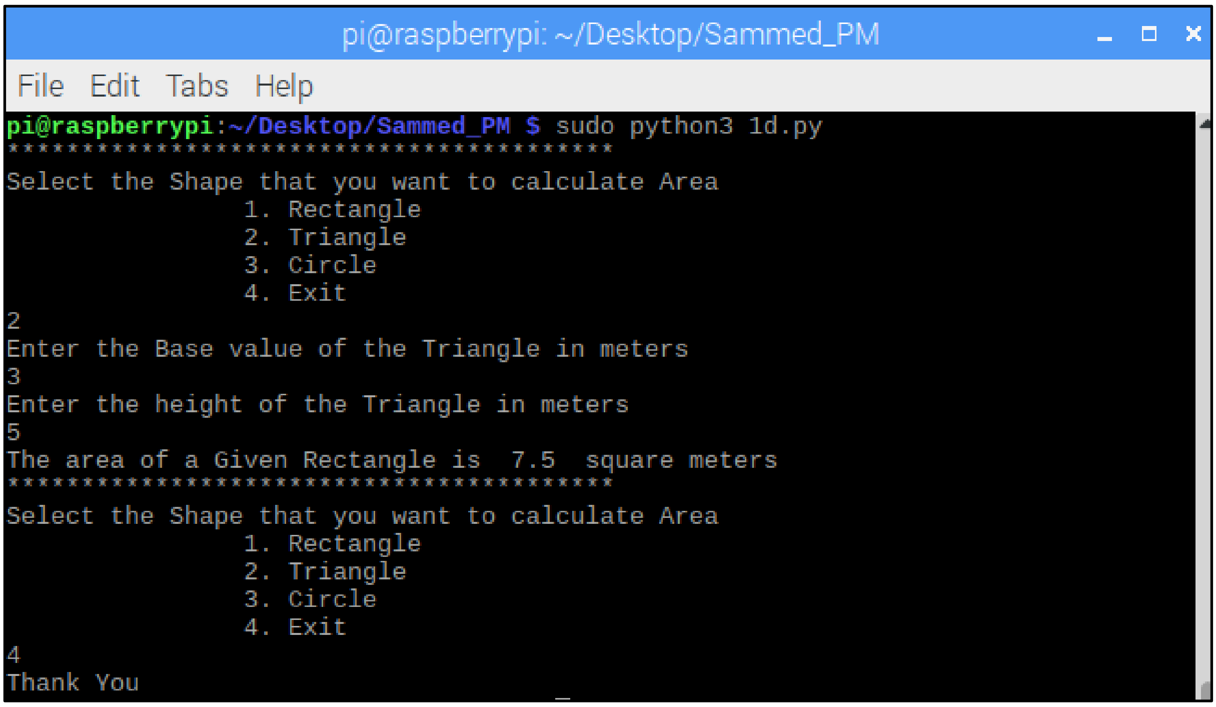
3. Circle

4. Exit

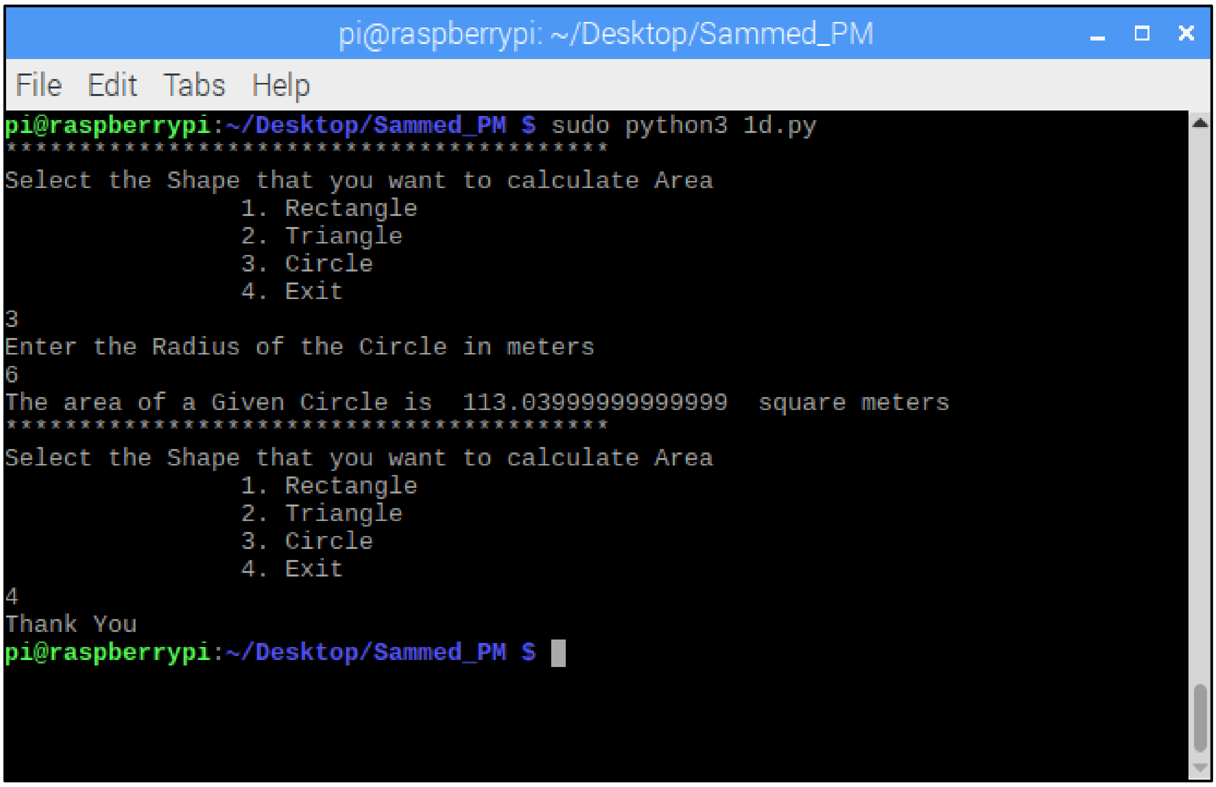
Output:



```
pi@raspberrypi:~/Desktop/Sammed_PM $ sudo python3 1d.py
*****************************************
Select the Shape that you want to calculate Area
                1. Rectangle
                2. Triangle
                3. Circle
                4. Exit
1
Enter the Width of the Rectangle in meters
10
Enter the height of the Rectangle in meters
20
The area of a Given Rectangle is  200  square meters
*****************************************
Select the Shape that you want to calculate Area
                1. Rectangle
                2. Triangle
                3. Circle
                4. Exit
```



```
pi@raspberrypi:~/Desktop/Sammed_PM $ sudo python3 1d.py
*****************************************
Select the Shape that you want to calculate Area
                1. Rectangle
                2. Triangle
                3. Circle
                4. Exit
2
Enter the Base value of the Triangle in meters
3
Enter the height of the Triangle in meters
5
The area of a Given Rectangle is  7.5  square meters
*****************************************
Select the Shape that you want to calculate Area
                1. Rectangle
                2. Triangle
                3. Circle
                4. Exit
4
Thank You
```



```
pi@raspberrypi:~/Desktop/Sammed_PM $ sudo python3 1d.py
*****************************************
Select the Shape that you want to calculate Area
                1. Rectangle
                2. Triangle
                3. Circle
                4. Exit
3
Enter the Radius of the Circle in meters
6
The area of a Given Circle is  113.03999999999999  square meters
*****************************************
Select the Shape that you want to calculate Area
                1. Rectangle
                2. Triangle
                3. Circle
                4. Exit
4
Thank You
pi@raspberrypi:~/Desktop/Sammed_PM $ 
```

1e.py:

```
print("Enter Your name : ")

name = input()

print("Enter How many times you want to print your name")

n = int(input())

for i in range(n):

    print(name)
```

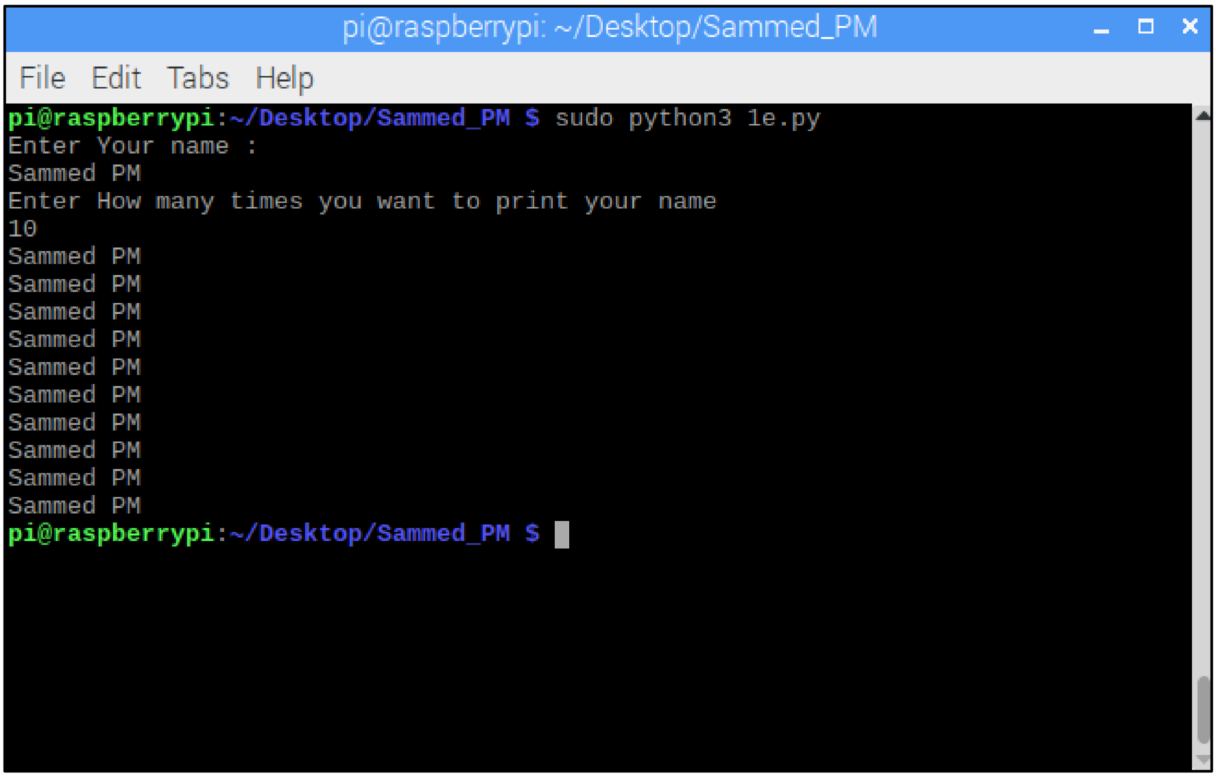
Procedure:

Print a Name “n” times where Name and “n” are read from standard I/p using FOR & WHILE loops.

Run the program as: **sudo python3 1e.py**

It prompts you to type the name and how many times the name has to be displayed.

Output:



```
pi@raspberrypi:~/Desktop/Sammed_PM $ sudo python3 1e.py
Enter Your name :
Sammed PM
Enter How many times you want to print your name
10
Sammed PM
Sammed PM
Sammed PM
Sammed PM
Sammed PM
Sammed PM
Sammed PM
Sammed PM
Sammed PM
Sammed PM
pi@raspberrypi:~/Desktop/Sammed_PM $
```

1f.py:

```
print("Enter Numerator Value : ")
num1 = int(input())
print("Enter Denominator value : ")
num2 = int(input())
try:
        result = num1/num2
        print("The Division of Given Numbers is : ", result)
except ZeroDivisionError:
        print("Divide By zero Error. The Denominator should not be Zero")
```

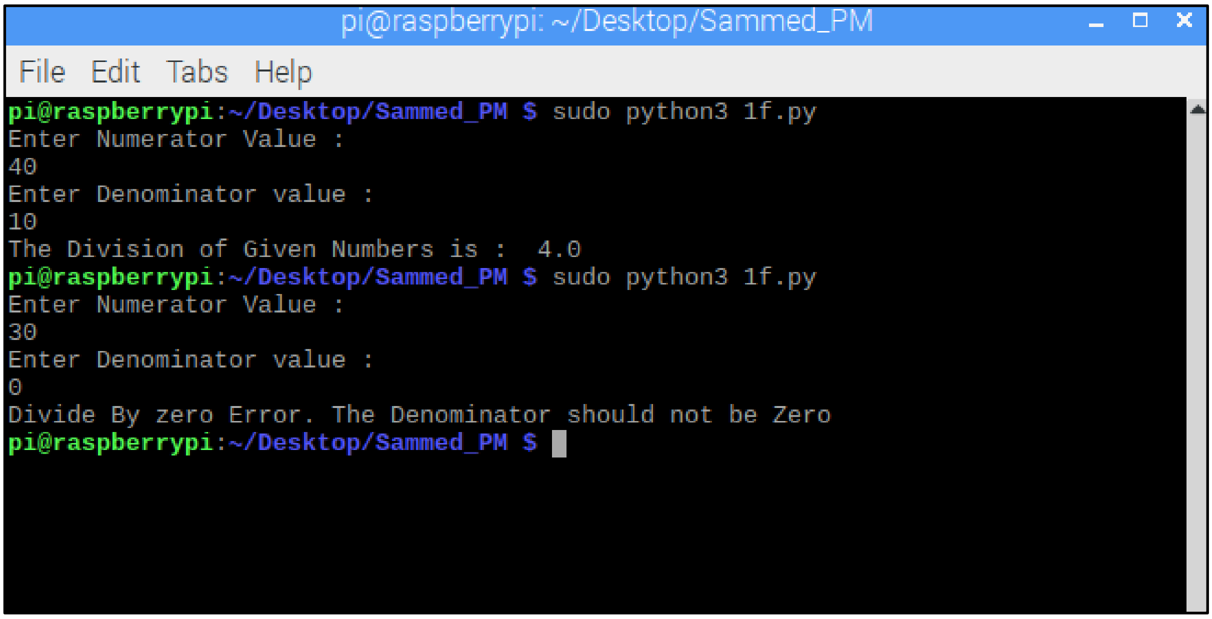
Procedure:

Handle divided by zero exception.

Run the program as: **sudo python3 1f.py**

It prompts you to enter the numerator and denominator value.

Output:



```
pi@raspberrypi:~/Desktop/Sammed_PM $ sudo python3 1f.py
Enter Numerator Value :
40
Enter Denominator value :
10
The Division of Given Numbers is :  4.0
pi@raspberrypi:~/Desktop/Sammed_PM $ sudo python3 1f.py
Enter Numerator Value :
30
Enter Denominator value :
0
Divide By zero Error. The Denominator should not be Zero
pi@raspberrypi:~/Desktop/Sammed_PM $
```

1g.py:

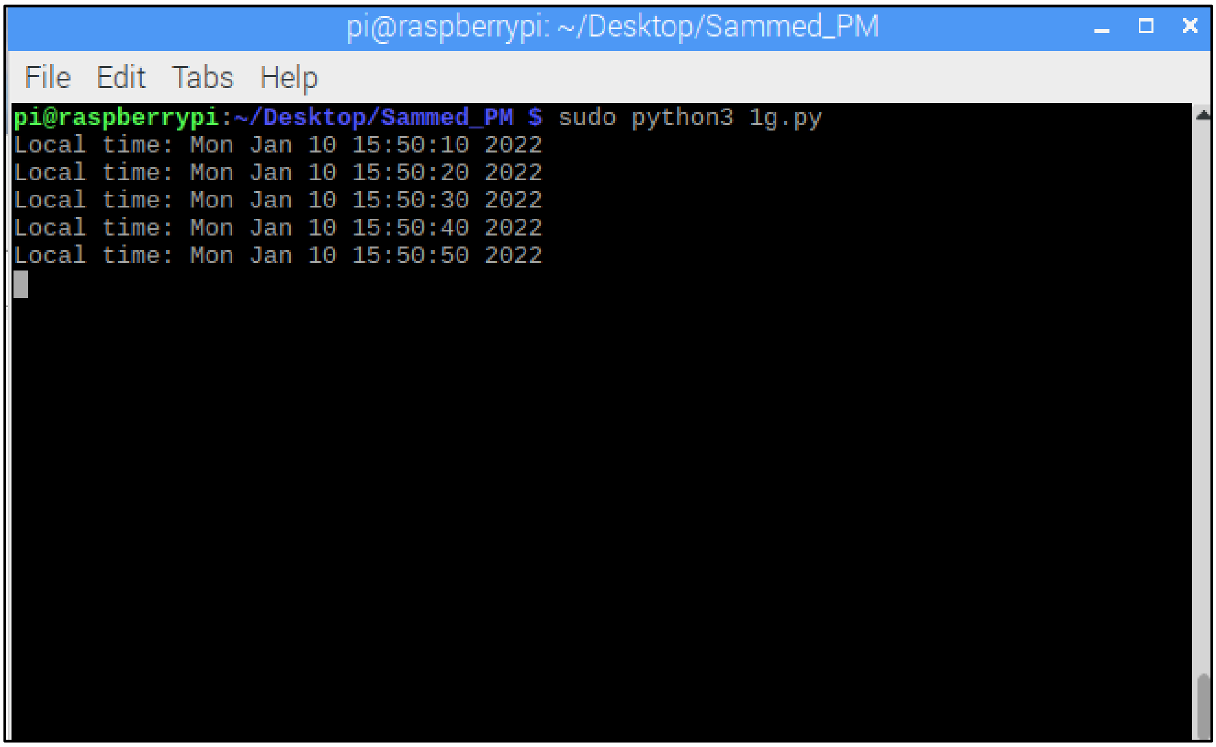
```
import time
for i in range(10):
        seconds = time.time() #it will give to you as a float values
        local_time = time.ctime(seconds) #it will gives to you as a current local time
        print("Local time:", local_time)
        time.sleep(10)
```

Procedure:

Print current time for 10 times with an interval of 10 seconds.

Run the program as: **sudo python3 1g.py**

Output:



```
pi@raspberrypi:~/Desktop/Sammed_PM $ sudo python3 1g.py
Local time: Mon Jan 10 15:50:10 2022
Local time: Mon Jan 10 15:50:20 2022
Local time: Mon Jan 10 15:50:30 2022
Local time: Mon Jan 10 15:50:40 2022
Local time: Mon Jan 10 15:50:50 2022
```

1h.py:

```
import time
for i in range(10):
        seconds = time.time() #it will give to you as a float values
        local_time = time.ctime(seconds) #it will gives to you as a current local time
        print("Local time:", local_time)
        time.sleep(10)
```

myfile.txt:

```
Hello World! Welcome to python programming
This is sample text file for testing the words
We are the students of MCA
These are my first python
```

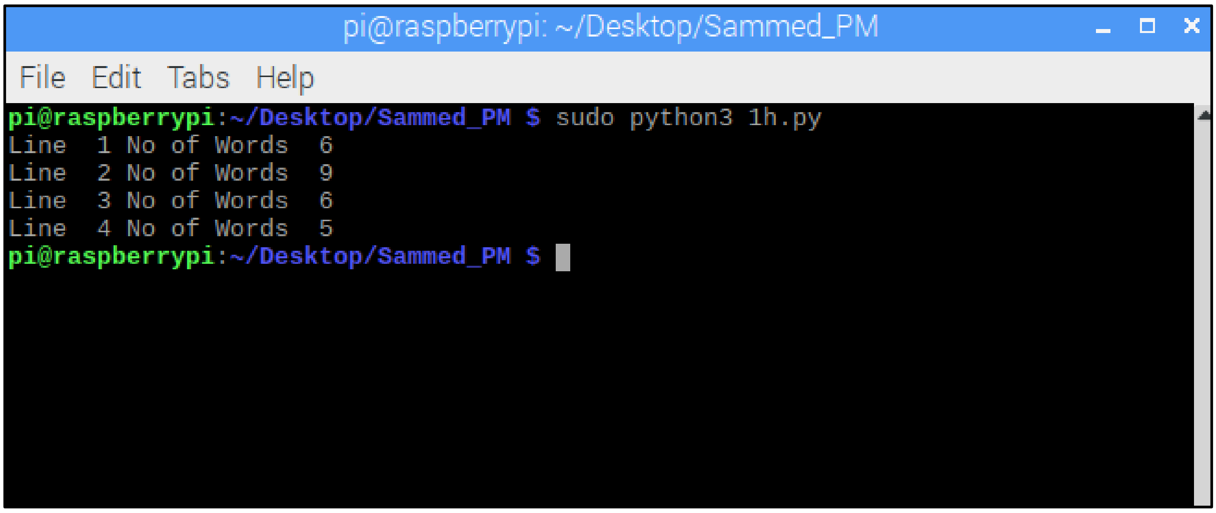
Procedure:

Read a file line by line and print the word count of each line.

Run the program as: **sudo python3 1h.py**

In the program we are using the file named as “Myfile.txt” to enter our text. When you run the program it invokes the Myfile.txt and gives us the result.

Output:



```
pi@raspberrypi:~/Desktop/Sammed_PM $ sudo python3 1h.py
Line  1 No of Words  6
Line  2 No of Words  9
Line  3 No of Words  6
Line  4 No of Words  5
pi@raspberrypi:~/Desktop/Sammed_PM $
```

Program 2: Get input from two switches and switch on corresponding LEDs.

```
import time
import RPi.GPIO as gpio
gpio.setwarnings(False)
gpio.setmode(gpio.BOARD)
led1 = 15
led2 = 13
switch1 = 37
switch2 = 35
gpio.setup(led1,gpio.OUT,initial=1)
gpio.setup(led2,gpio.OUT,initial=1)
gpio.setup(switch1,gpio.IN)
gpio.setup(switch2,gpio.IN)
def glow_led(event):
    if event == switch1 :
        gpio.output(led1, False)
        time.sleep(3)
        gpio.output(led1, True)
    elif event == switch2 :
        gpio.output(led2, False)
        time.sleep(3)
        gpio.output(led2, True)
gpio.add_event_detect(switch1, gpio.RISING , callback = glow_led, bouncetime = 1)
gpio.add_event_detect(switch2, gpio.RISING , callback = glow_led, bouncetime = 1)
try:
    while(True):
       time.sleep(1)
except KeyboardInterrupt:
    gpio.cleanup()
```

Procedure:

Get I/p from 2 switches and switch ON corresponding LED’s.

Run the program as: **sudo python3 2.py**

Output:

Press the SW1 or SW2 on the IoT SENSOR/IFC-02, the corresponding led will be on for 3 sec and will turned off. Only one switch will be taken as the I/p. SW1 = LED2, SW2 = LED3. This program will be in loop, press **Ctrl + C** to quit.

Program 3: Flash an LED at a given on time and off time cycle, where the two times are taken from a file.

```
import time
import RPi.GPIO as gpio
gpio.setwarnings(False)
gpio.setmode(gpio.BOARD)
led1 =  10
gpio.setup(led1,gpio.OUT,initial=1)
file1 = open('ledintervals.txt', 'r')
Lines = file1.readlines()
ON_TIME = int(Lines[0].split("=")[1])
OFF_TIME = int(Lines[1].split("=")[1])
try:
fea    while(True):
        gpio.output(led1,False)
        time.sleep(ON_TIME)
        gpio.output(led1,True)
        time.sleep(OFF_TIME)
except KeyboardInterrupt:
    #cleanup GPIO settings before exiting
    gpio.cleanup()
```

ledintervals.txt:

```
ON_TIME=5
OFF_TIME=10
```

Procedure:

10 Flash an LED at a given ON time & OFF time cycle where the two times are taken from a file.

Run the program as: **sudo python3 3.py**

When you run this program, it invokes the file ledintervals.txt where in this file you can enter the LED’s ON time and OFF time.

Output:

The led LED1 will be turned ON for 10 sec and turned OFF for 5 sec.

Program 4: Switch on a relay at a given time using cron, where the relay’s contact terminals are connected to a load.

```
import time
import RPi.GPIO as gpio
gpio.setwarnings(False)
gpio.setmode(gpio.BOARD)
relay1 = 38
gpio.setup(relay1,gpio.OUT,initial=0)
try:
        gpio.output(relay1, True)
        print("Relay is Switched On. Please Press ctrl+c to exit")
        time.sleep(15)
        print("Relay is Switched Off.")
        gpio.output(relay1, False)
except KeyboardInterrupt:
        gpio.cleanup()
        print("Program Exited")
```

Procedure 01:

Switch on a relay at a given time using CRON, where the relay’s contact terminals are connected to a load.

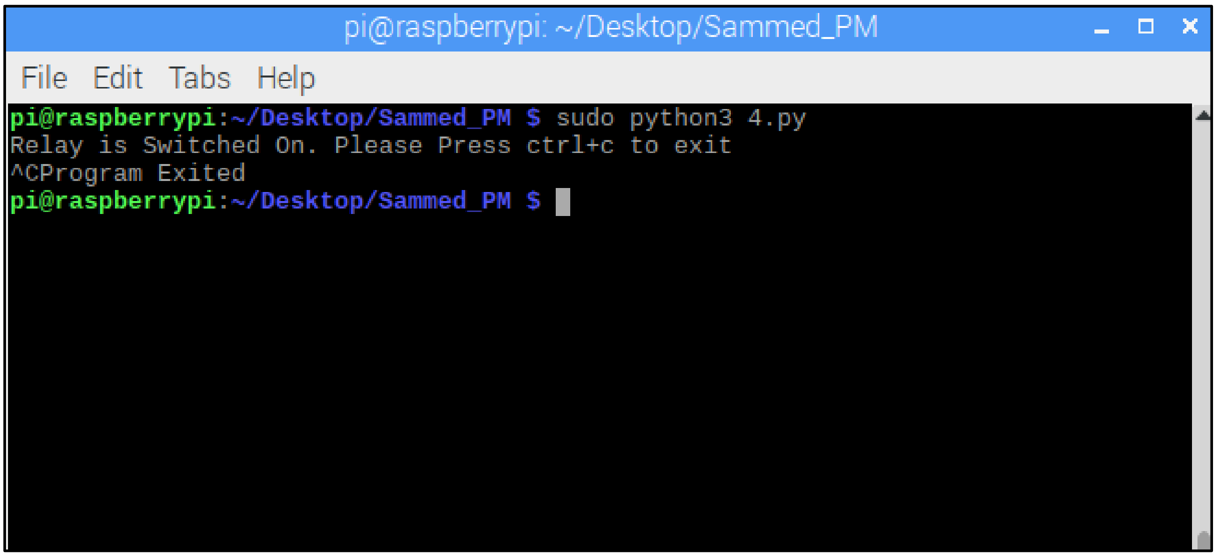
Run the program as: **sudo python3 4.py**

Output 01:

“Relay is switched On. Please Press ctrl+c to exit” will be displayed.

If the program is aborted by pressing ctrl+c before the time period “Program exited” will be displayed. Otherwise after the specified time delay in seconds relay gets switched off.

“Relay is Switched Off.” will be displayed and the program will be terminated.



```
pi@raspberrypi:~/Desktop/Sammed_PM $ sudo python3 4.py
Relay is Switched On. Please Press ctrl+c to exit
^CProgram Exited
pi@raspberrypi:~/Desktop/Sammed_PM $ 
```

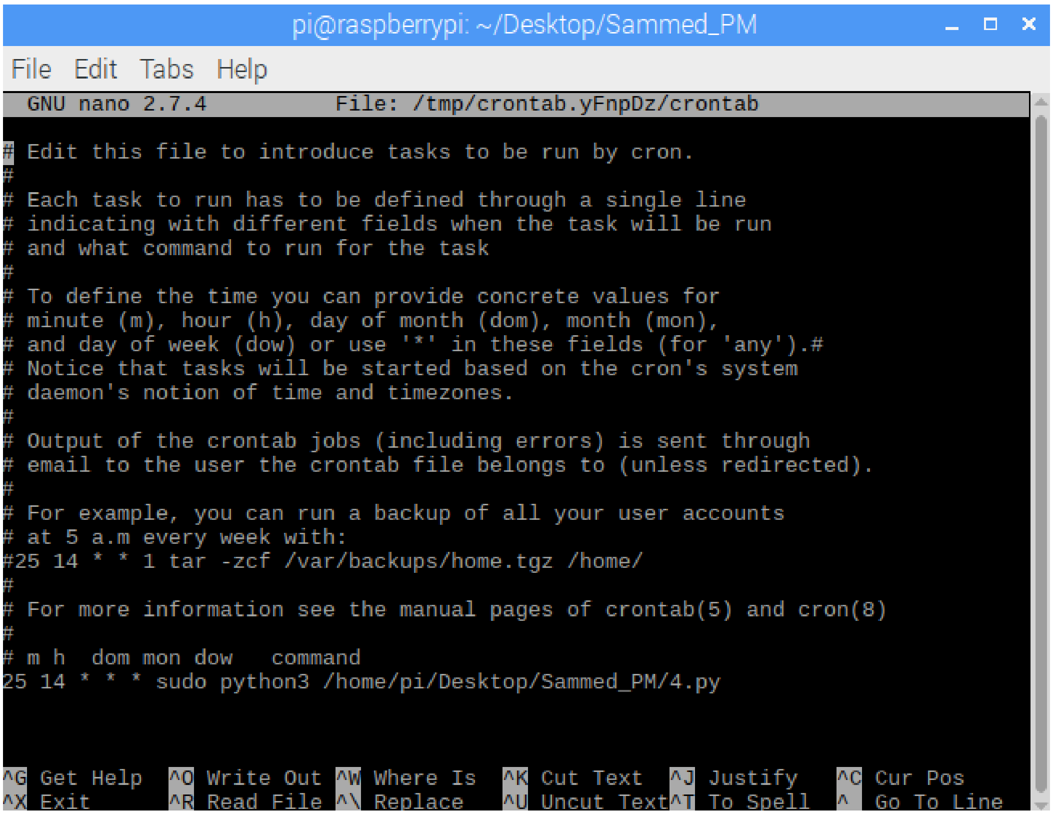
<u>Procedure 02:</u>

To run the same program using cron scheduler, type **crontab -e** in Terminal window. Select the option 2 (nano editor) from the menu if it is first time and a file will be opened. After the last line type the following: m h dom mon dow command. This is the format

For Example we type:

40 10 * * * sudo python3 /home/pi/Desktop/MCASyllabus/4.py

in the above line 40 is the min 10 is the hr * * * is date, month, day of the week, type the command as sudo python3 followed by the file name which has to be executed along with the full path.



```
  GNU nano 2.7.4             File: /tmp/crontab.yFnpDz/crontab

# Edit this file to introduce tasks to be run by cron.
#
# Each task to run has to be defined through a single line
# indicating with different fields when the task will be run
# and what command to run for the task
#
# To define the time you can provide concrete values for
# minute (m), hour (h), day of month (dom), month (mon),
# and day of week (dow) or use '*' in these fields (for 'any').#
# Notice that tasks will be started based on the cron's system
# daemon's notion of time and timezones.
#
# Output of the crontab jobs (including errors) is sent through
# email to the user the crontab file belongs to (unless redirected).
#
# For example, you can run a backup of all your user accounts
# at 5 a.m every week with:
#25 14 * * 1 tar -zcf /var/backups/home.tgz /home/
#
# For more information see the manual pages of crontab(5) and cron(8)
#
# m h  dom mon dow   command
25 14 * * * sudo python3 /home/pi/Desktop/Sammed_PM/4.py



^G Get Help  ^O Write Out ^W Where Is  ^K Cut Text  ^J Justify   ^C Cur Pos
^X Exit      ^R Read File ^\ Replace   ^U Uncut Text^T To Spell  ^  Go To Line
```

Output 02:

With the above configuration setup in cron scheduler, the program will execute at 14:25 every day.

Program 5: Access an image through a Pi web cam.

```
from picamera import PiCamera

from time import sleep

import datetime

camera = PiCamera()

camera.start_preview()

current_date = datetime.datetime.now().strftime('%d-%m-%Y %H:%M:%S')

sleep(3)

camera.capture('/home/pi/Desktop/Sammed_PM/images/'+current_date+'.jpg')

camera.stop_preview()

print("Image captured")

#sleep(2)
```

Procedure:

Connect a Pi camera using a ribbon cable to Raspberry Pi in camera slot, which is on pi board.

NOTE: - Pi camera should be connected when trainer is powered off.

Run the program as: **sudo python3 5.py**

Output:

Program 6: Control a light source using web page.

```
import RPi.GPIO as GPIO

import time

import datetime

led = 13 #P15

GPIO.setmode(GPIO.BOARD)

GPIO.setwarnings(False)

GPIO.setup(led, GPIO.OUT,initial=1) #Initialising GPIO pins as output

GPIO.setup(led,GPIO.OUT)

from flask import Flask, render_template #Importing Flask

app = Flask(__name__)

@app.route('/') #Simple Hello World Route

def hello_world():

    return render_template('web.html')

@app.route('/redledon') #Route for Turning RedLed On

def redledon():

    GPIO.output(13, GPIO.LOW) #Turning Pin 31 --> Red Led HIGH

    now = datetime.datetime.now()

    timeString = now.strftime("%Y-%m-%d %H:%M")

    templateData = {

      'status' : 'ON',

      'time': timeString

      }

    return render_template('web.html', **templateData)

@app.route('/redledoff') #Route for Turning RedLed Off

def redledoff():

    GPIO.output(13, GPIO.HIGH)  #Turning Pin 31 --> Red Led LOW

    now = datetime.datetime.now()
    timeString = now.strftime("%Y-%m-%d %H:%M")
    templateData = {
      'status' : 'OFF',
      'time': timeString
      }
    return render_template('web.html', **templateData)
if __name__ == "__main__":
    app.run(debug = True, port = 4030, host='192.168.118.156')
```

web.html:

```
<html>
<body>
            <title>Raspberry PI Remote Control</title>
            <h1>Raspberry PI Remote Control</h1>
            <h2>Light Status : {{status}}, Last Modified : {{time}}</h2>
            <form action="http://192.168.118.156:4030/redledon">
                         <input type="submit" value="Red LED On">
            </form>
            <form action="http://192.168.118.156:4030/redledoff">
                        <input type="submit" value="Red LED Off">
            </form>
</body>
</html>
```

<u>Procedure:</u>

Install the package python3-flask. Enter the command “sudo apt-get install python3-flask”.
Find out the ip address of the system by entering the command “ifconfig” note down the inet address of wlan0. (Ex: inet 192.168.1.7 netmask 255.255.255.0 broadcast 192.168.1.255)
Open the program 6.py using Thonny editor. Go to the last line of the program and modify the inet address which has been noted and the port no. (Ex: 3000 or 4000 or 5000), save the program.
Go to the folder “/home/pi/Desktop/MCASyllabus/templates” right click on “web.html”, file.
Open the file using geany editor.
Edit the line numbers 6 and 9 with the noted inet address and the port and save the file.

Run the program as: sudo python3 6.py
When the program is executed the below message appears on the screen.

```
* Running on http://192.168.118.156:4030/ (Press CTRL+C to quit)
* Restarting with stat
* Debugger is active!
* Debugger pin code: 216-997-631
```

Open the browser and in the URL type as displayed in the first line “http://192.168.118.156:4030” and press enter. Below message will be displayed on the screen.

Raspberry PI Remote Control

RED LED ON

RED LED OFF

Now click on the RED LED ON box. Observe the LED status of LED3, it should be ON and the below message will be displayed on the screen

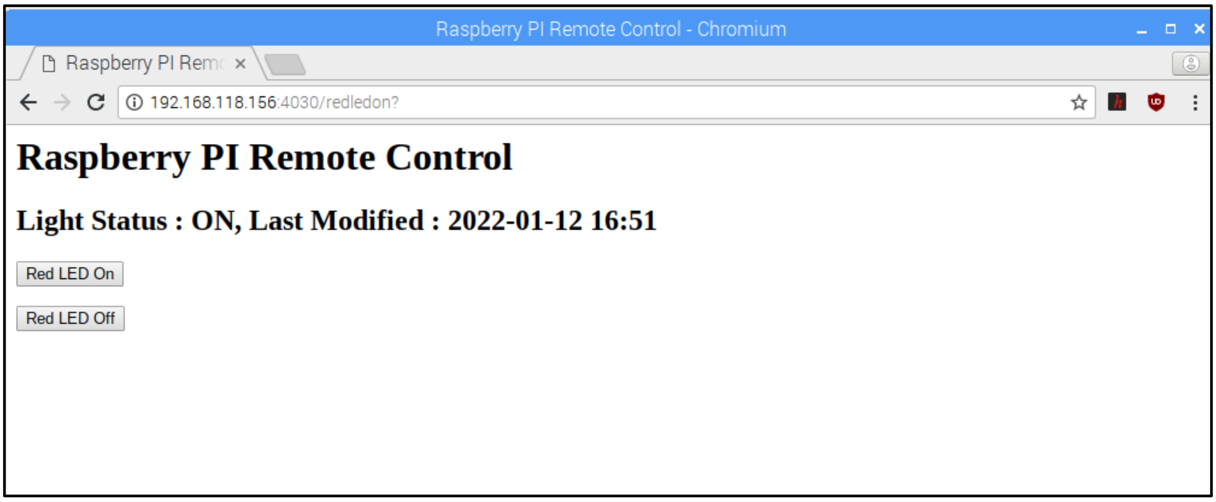
1.**Light Status: ON, Last Modified: 2021-05-04 16:56**

Now click on the RED LED OFF box. Observe the LED status of LED3, it should be OFF and the below message will be displayed on the screen

2.Light Status: OFF, Last Modified: 2021-05-04 16:59

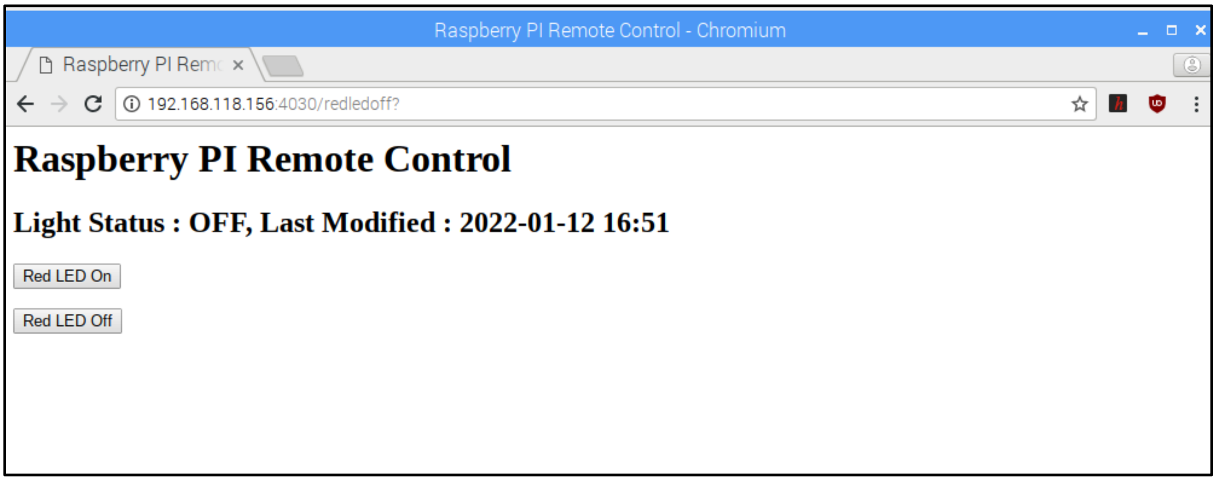
This will be in a loop. To abort the program press CTRL+C in the terminal window.

Output:


Raspberry PI Remote Control - Chromium
Raspberry PI Remc
192.168.118.156:4030/redledon?
Red LED On
Red LED Off

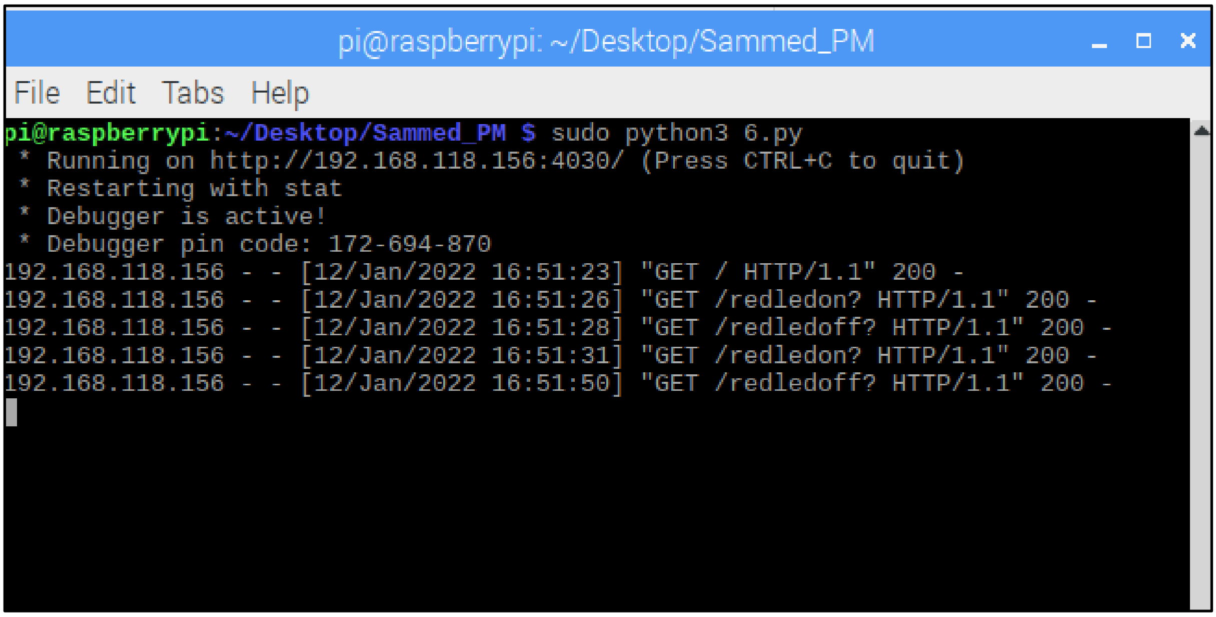
Raspberry PI Remote Control

Light Status : ON, Last Modified : 2022-01-12 16:51


Raspberry PI Remote Control - Chromium
Raspberry PI Remc
192.168.118.156:4030/redledoff?
Red LED On
Red LED Off

Raspberry PI Remote Control

Light Status : OFF, Last Modified : 2022-01-12 16:51



```
pi@raspberrypi:~/Desktop/Sammed_PM $ sudo python3 6.py
 * Running on http://192.168.118.156:4030/ (Press CTRL+C to quit)
 * Restarting with stat
 * Debugger is active!
 * Debugger pin code: 172-694-870
192.168.118.156 - - [12/Jan/2022 16:51:23] "GET / HTTP/1.1" 200 -
192.168.118.156 - - [12/Jan/2022 16:51:26] "GET /redledon? HTTP/1.1" 200 -
192.168.118.156 - - [12/Jan/2022 16:51:28] "GET /redledoff? HTTP/1.1" 200 -
192.168.118.156 - - [12/Jan/2022 16:51:31] "GET /redledon? HTTP/1.1" 200 -
192.168.118.156 - - [12/Jan/2022 16:51:50] "GET /redledoff? HTTP/1.1" 200 -
```

Program 7: Implement an intruder system that sends an alert to the given email.

```
import RPi.GPIO as gpio

import picamera

import time

import smtplib

from email.mime.multipart import MIMEMultipart

from email.mime.text import MIMEText

from email.mime.base import MIMEBase

from email import encoders

from email.mime.image import MIMEImage

fromaddr = "sammedpm40@gmail.com"    # change the email address accordingly

toaddr = "sammedpm48@gmail.com"

mail = MIMEMultipart()

mail['From'] = fromaddr

mail['To'] = toaddr

mail['Subject'] = "Attachment"

body = "Please find the attachment"

led=15

pir=12

HIGH=1

LOW=0

gpio.setwarnings(False)

gpio.setmode(gpio.BOARD)

gpio.setup(led, gpio.OUT)           # initialize GPIO Pin as outputs

gpio.setup(pir, gpio.IN)           # initialize GPIO Pin as input

data=""
```

```
def sendMail(data):
    mail.attach(MIMEText(body, 'plain'))
    print(data)
    dat='%s.jpg'%data
    print(data)
    attachment = open(dat, 'rb')
    image=MIMEImage(attachment.read())
    attachment.close()
    mail.attach(image)
    server = smtplib.SMTP('smtp.gmail.com', 587)
    server.starttls()
    server.login(fromaddr, "3574120.") #Password of Gmail Account
    text = mail.as_string()
    server.sendmail(fromaddr, toaddr, text)
    server.quit()
def capture_image():
    data= time.strftime("Image was captured on %H:%M:%S|%d_%b_%Y")
    camera.start_preview()
    time.sleep(5)
    print(data)
    camera.capture('%s.jpg'%data)
    camera.stop_preview()
    time.sleep(1)
    sendMail(data)
gpio.output(led , 0)
camera = picamera.PiCamera()
camera.rotation=180
camera.awb_mode= 'auto'
camera.brightness=55
```

```
while 1:
    if gpio.input(pir)==1:
        gpio.output(led, HIGH)
        capture_image()
        while(gpio.input(pir)==1):
            time.sleep(1)
    else:
        gpio.output(led, LOW)
        time.sleep(0.01)
```

Procedure:

Install the packages ssmtp and mailutils. Enter the command “sudo apt-get install ssmtp” and “sudo apt-get install mailutils” Edit the file by typing “sudo nano /etc/ssmtp/ssmtp.conf “. Edit the following lines in terminal window.
root=postmaster here replace the postmaster with your gmail id at line no. 6.
mailhub=mail, here replace the mail with smtp.gmail.com:587 at line no. 10.
Add these following just below the 16th line
AuthUser=<your gmail id>
AuthPass=<your gmail password>
In this file the last line will be “#FromLineOverride=YES”, remove # symbol (uncomment) in this line, then add the below parameters (next 2 lines).
UseSTARTTLS=YES
UseTLS=YES
“UseTLS=YES” will be the last line of the file. Save and quit the file (ctrl+x then press “Y”).
Go to the folder “/home/pi/Desktop/MCASyllabus/7.py” edit the code in line “from addr= put your mail id” and “to addr=put your mail id”(20,21--mail_id) and server. Login (from addr, "put your password") (51—password)

Short JP2 in IoT Sensor IFC-02 Board

Run the program as: **sudo python3 7.py**

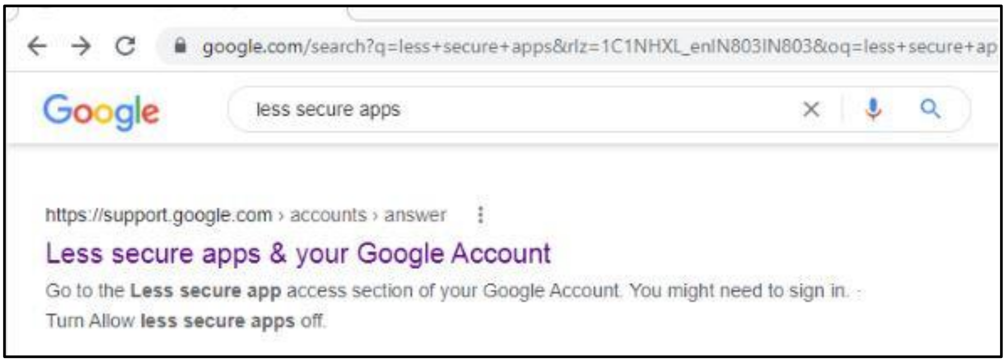
After the program is executed, when an object is placed in front of the PIR sensor, the image will be captured and the following message will be displayed in the console. At the same time the image is sent as an attachment to the recipient mail id. This program will be in loop, press **Ctrl + C** to quit.

```
Image was captured on 09:59:05|07_May_2021
Image was captured on 09:59:05|07_May_2021
Image was captured on 09:59:05|07_May_2021
```

Note: If error comes like authentication error then we need to ENABLE the 'less secure app'.

Procedure to ENABLE the less secure apps:

Step 1: Click on it.



google.com/search?q=less+secure+apps&rlz=1C1NHXL_enIN803IN803&oq=less+secure+ap

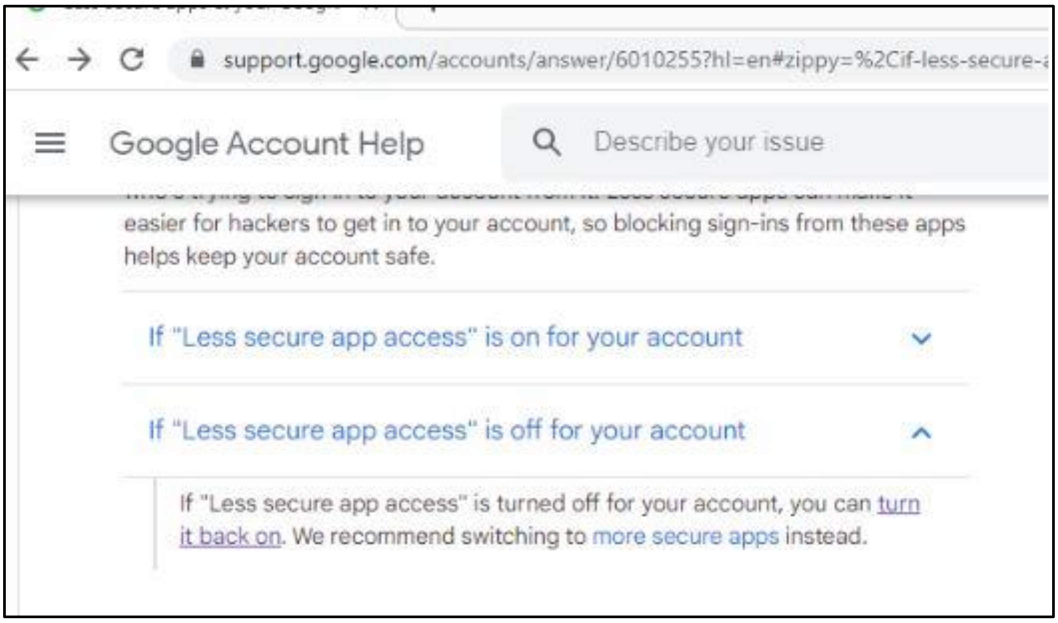
less secure apps

https://support.google.com › accounts › answer

Less secure apps & your Google Account

Go to the **Less secure app** access section of your Google Account. You might need to sign in. · Turn Allow **less secure apps** off.

Step 2: Click on 'turn it back on’.



support.google.com/accounts/answer/6010255?hl=en#zippy=%2Cif-less-secure-

Google Account Help

Describe your issue

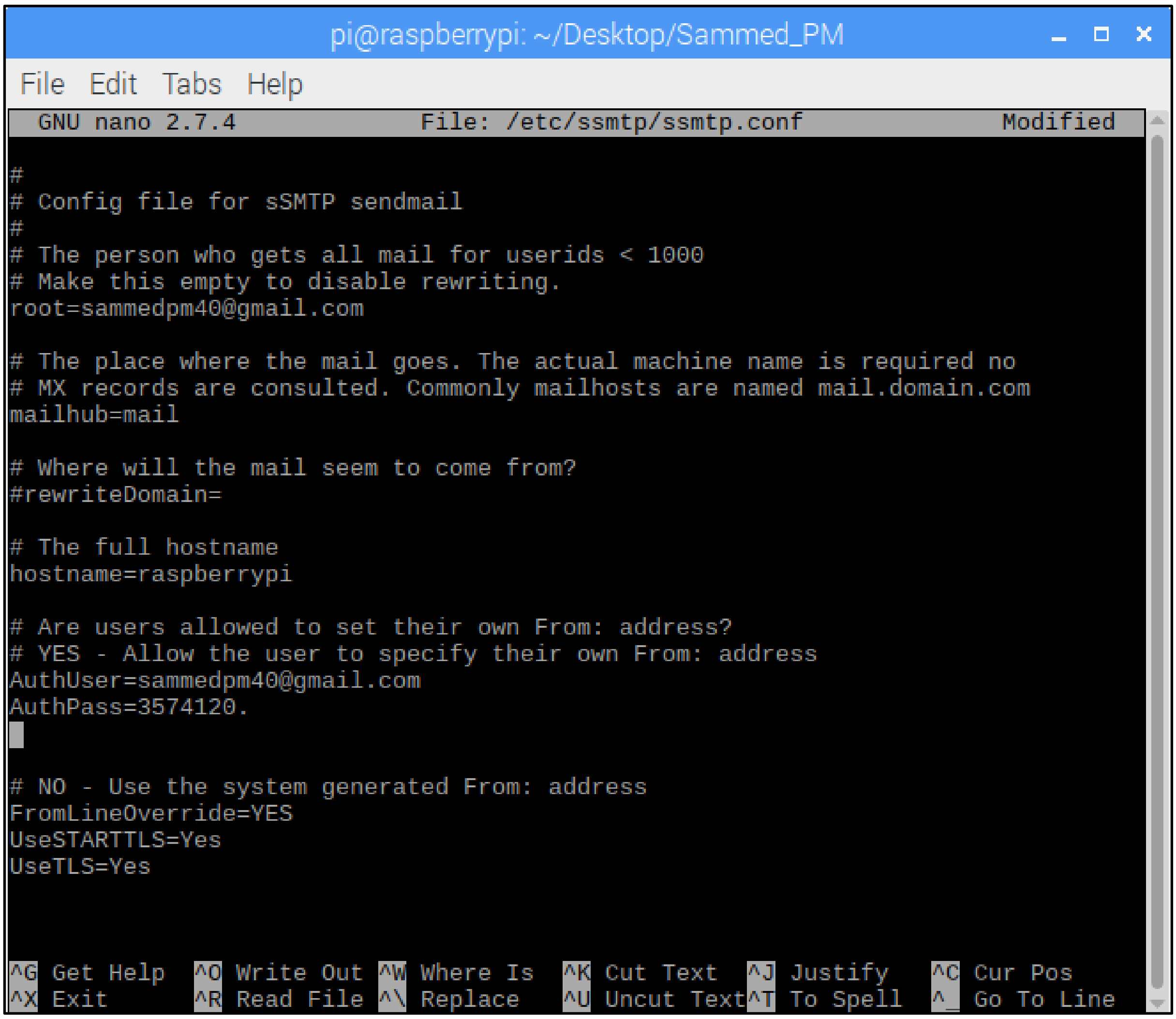
easier for hackers to get in to your account, so blocking sign-ins from these apps helps keep your account safe.

If "Less secure app access" is on for your account

If "Less secure app access" is off for your account

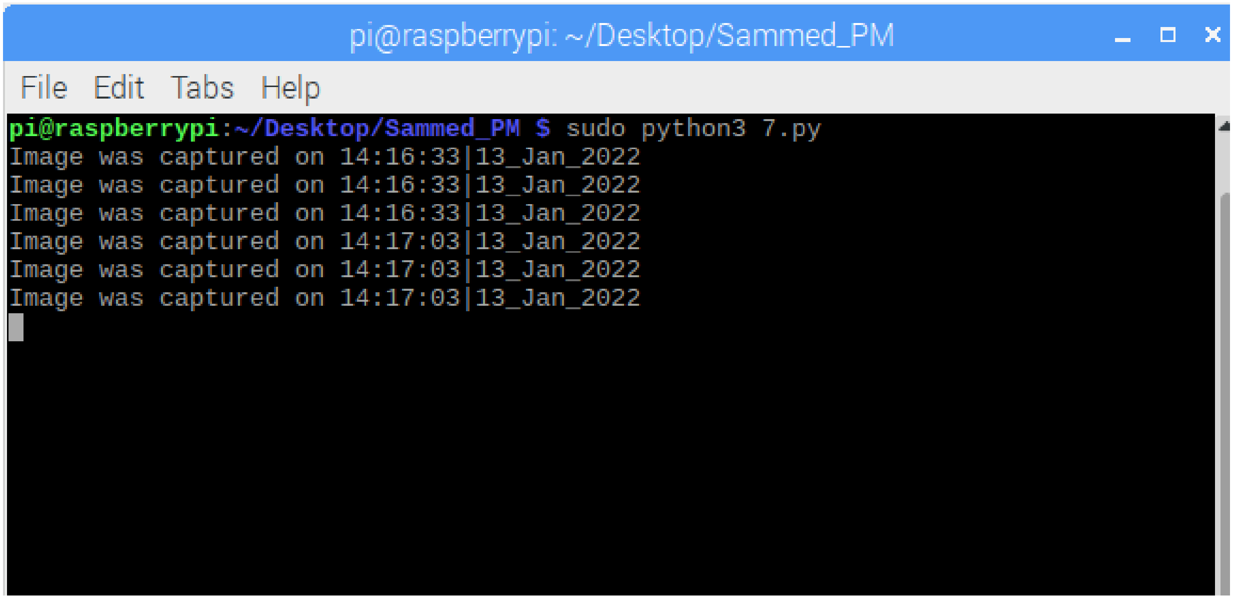
If "Less secure app access" is turned off for your account, you can turn it back on. We recommend switching to more secure apps instead.

Step 3: After step 2 there is login window will open, login with your given mail id and password. Then you can allow/Enable Less secure apps.

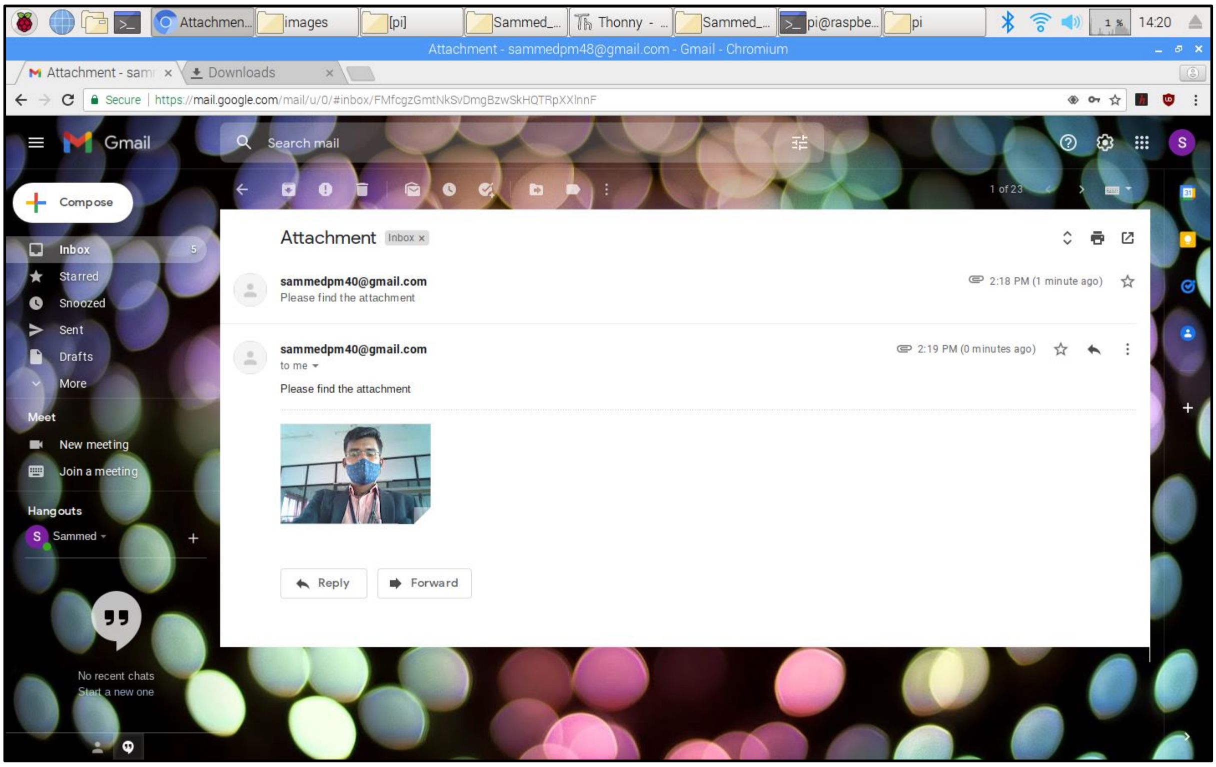


```
  GNU nano 2.7.4              File: /etc/ssmtp/ssmtp.conf               Modified

#
# Config file for sSMTP sendmail
#
# The person who gets all mail for userids < 1000
# Make this empty to disable rewriting.
root=sammedpm40@gmail.com

# The place where the mail goes. The actual machine name is required no
# MX records are consulted. Commonly mailhosts are named mail.domain.com
mailhub=mail

# Where will the mail seem to come from?
#rewriteDomain=

# The full hostname
hostname=raspberrypi

# Are users allowed to set their own From: address?
# YES - Allow the user to specify their own From: address
AuthUser=sammedpm40@gmail.com
AuthPass=3574120.


# NO - Use the system generated From: address
FromLineOverride=YES
UseSTARTTLS=Yes
UseTLS=Yes



^G Get Help  ^O Write Out ^W Where Is  ^K Cut Text  ^J Justify   ^C Cur Pos
^X Exit      ^R Read File ^\ Replace   ^U Uncut Text^T To Spell  ^_ Go To Line
```

Output:



```
pi@raspberrypi:~/Desktop/Sammed_PM $ sudo python3 7.py
Image was captured on 14:16:33|13_Jan_2022
Image was captured on 14:16:33|13_Jan_2022
Image was captured on 14:16:33|13_Jan_2022
Image was captured on 14:17:03|13_Jan_2022
Image was captured on 14:17:03|13_Jan_2022
Image was captured on 14:17:03|13_Jan_2022
```



Attachmen... images [pi] Sammed_... Thonny - ... Sammed_... pi@raspbe... pi
14:20
Attachment - sammedpm48@gmail.com - Gmail - Chromium
Attachment - samr ×
Downloads ×
Secure | https://mail.google.com/mail/u/0/#inbox/FMfcgzGmtNkSvDmgBzwSkHQTRpXXlnnF
Gmail
Search mail
Compose
Inbox 5
Starred
Snoozed
Sent
Drafts
More
Meet
New meeting
Join a meeting
Hangouts
Sammed
No recent chats
Start a new one
1 of 23

Attachment Inbox ×

sammedpm40@gmail.com
Please find the attachment
2:18 PM (1 minute ago)

sammedpm40@gmail.com
to me
2:19 PM (0 minutes ago)

Please find the attachment

Program 8: Get the status of a bulb at a remote place (on the LAN) through web.

```
import time

import RPi.GPIO as gpio

from flask import Flask, render_template   #Importing Flask

import datetime

app = Flask(__name__)

gpio.setwarnings(False)

gpio.setmode(gpio.BOARD)

led1 = 13

switch1 = 35

gpio.setup(led1,gpio.OUT,initial=1)

gpio.setup(switch1,gpio.IN)

light_status = "OFF"

def glow_led(event):

   print("Entered Here")

   global light_status

   if event == switch1 and light_status == "OFF":

      gpio.output(led1, False)

      light_status = "ON"

   elif event == switch1 and light_status == "ON":

      gpio.output(led1, True)

      light_status = "OFF"

@app.route('/') #Route for Turning RedLed On

def ledstatus():

   now = datetime.datetime.now()

   timeString = now.strftime("%H:%M %d-%m-%Y")

   templateData = {
        'status' : light_status,
        'time': timeString
        }
    return render_template('lightstatus.html', **templateData)
gpio.add_event_detect(switch1, gpio.RISING , callback = glow_led, bouncetime = 100)
app.run(debug = True, port = 4030, host='192.168.118.156')
```

lightstatus.html:

```
<html>
<body>
        <title>Raspberry PI Remote Light Status</title>
        <h1>Raspberry PI Remote Control</h1>
        <h2>Light Status : {{status}}, Last Seen : {{time}}</h2>
        <form action="http://192.168.118.156:4030">
                <input type="submit" value="Get Light Status">
        </form>
</body>
</html>
```

Procedure:

LAN connection: Inter connect the 2 systems (IoT Sensor IFC-02 Board+ Raspberry pi board) through a Hub with the help of LAN cables. Install the package python3-flask. Enter the command **“sudo apt-get install python3-flask”.**

Find out the ip address of the Host system by entering the command **“ifconfig**” note down the inet address. **(Ex: inet 192.168.118.156')**

Open the program 8.py using Thonny editor. Go to the last line of the program and modify the inet address which has been noted and the port no. (ex: port No. 3000 or 4000 or 5000), save the program.

Go to the folder “/home/pi/Desktop/SammedPM/templates” right click on lightstatus.html”, file. Open the file using geany editor. Edit the line number 6 with the port no. and noted inet address and save the file and quit.

Run the program as: **sudo python3 8.py**

When the program is executed the below message appears on the console.

```
* Running on http://192.168.118.156:4030/ (Press CTRL+C to quit)
* Restarting with stat
* Debugger is active!
* Debugger pin code: 216-997-631
```

When the switch SW2 is pressed on the IoT Sensor IFC-02 Board and the LED3 led will be toggled and below message will be displayed on the console.

```
Entered Here
Entered Here
```

This program is now running on a host system (IoT Sensor IFC-02 Board).

On the guest system (only Raspberry pi board) open the browser and type in the URL http://192.168.118.156:4030 (Host system IP address and port no.), and press enter. The following message will be displayed on the screen.

Raspberry PI Remote Control

1Light Status: OFF, Last Seen : 11:52 07-05-2021

Get Light Status

Assume the switch SW2 is pressed on the Host system (IoT Sensor IFC-02 Board) the LED3 led will be toggled. If you press on the button “Get Light Status” on the Guest system(Raspberry Pi Board)the status of the light will change as below

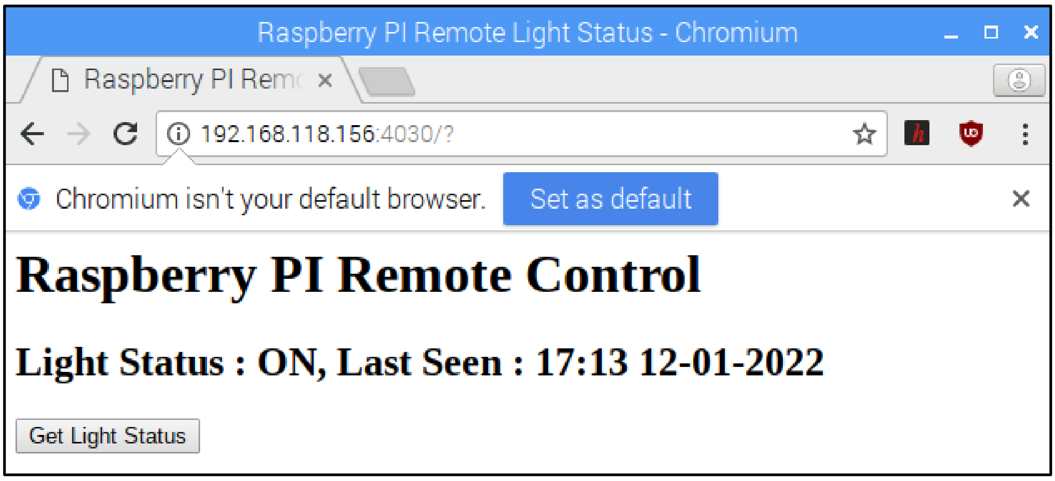
Raspberry PI Remote Control

1Light Status: ON, Last Seen : 12:02 07-05-2021

Get Light Status

This program running in the Host will be in loop, press **Ctrl + C** to quit.

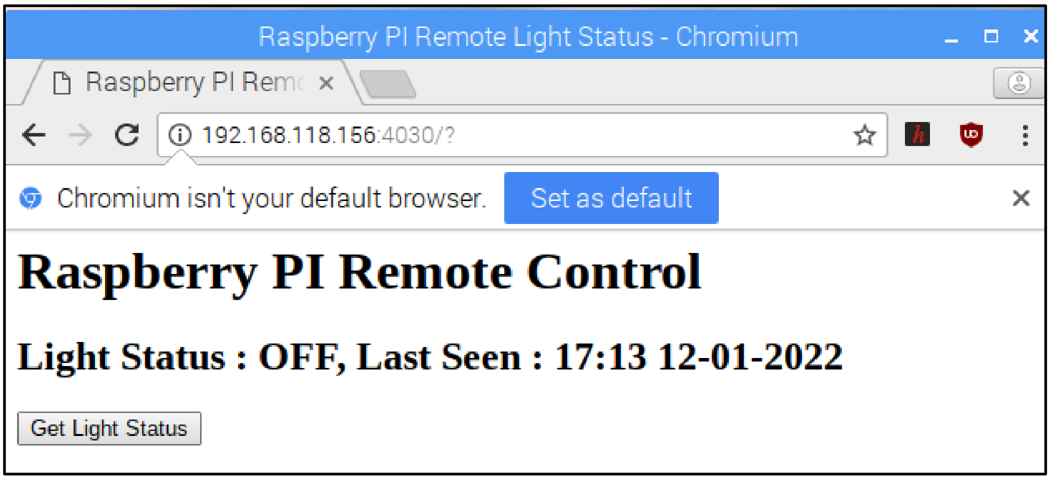
Output:



Raspberry PI Remote Light Status - Chromium
Raspberry PI Remo
192.168.118.156:4030/?
Chromium isn't your default browser.
Set as default
Get Light Status

Raspberry PI Remote Control

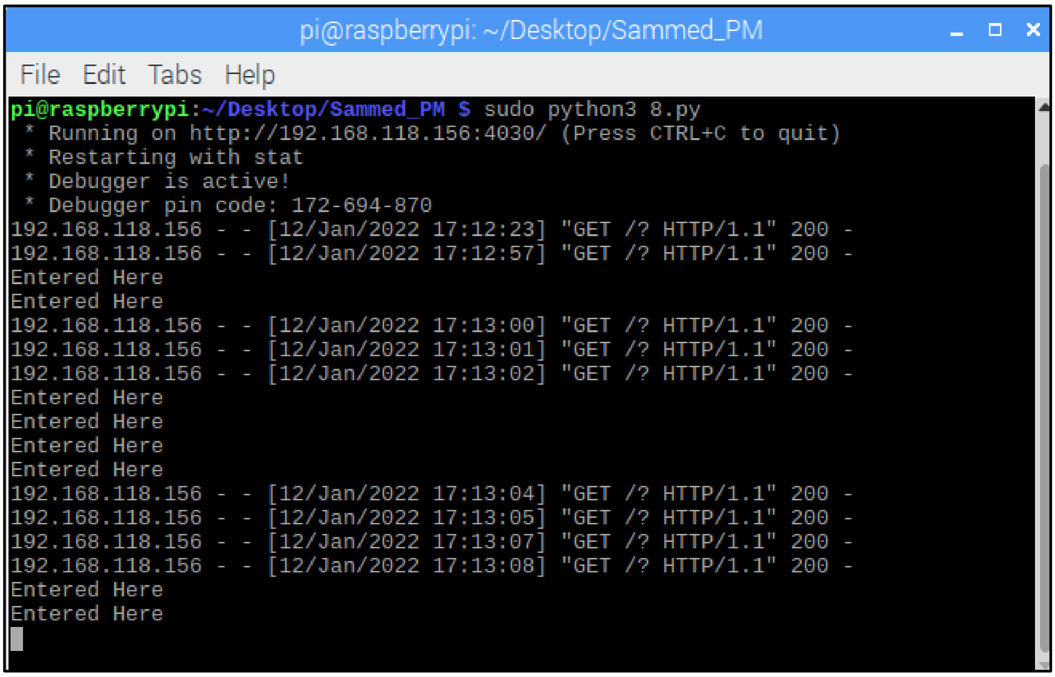
Light Status : ON, Last Seen : 17:13 12-01-2022



Raspberry PI Remote Light Status - Chromium
Raspberry PI Remo
192.168.118.156:4030/?
Chromium isn't your default browser.
Set as default
Get Light Status

Raspberry PI Remote Control

Light Status : OFF, Last Seen : 17:13 12-01-2022



```
pi@raspberrypi:~/Desktop/Sammed_PM $ sudo python3 8.py
 * Running on http://192.168.118.156:4030/ (Press CTRL+C to quit)
 * Restarting with stat
 * Debugger is active!
 * Debugger pin code: 172-694-870
192.168.118.156 - - [12/Jan/2022 17:12:23] "GET /? HTTP/1.1" 200 -
192.168.118.156 - - [12/Jan/2022 17:12:57] "GET /? HTTP/1.1" 200 -
Entered Here
Entered Here
192.168.118.156 - - [12/Jan/2022 17:13:00] "GET /? HTTP/1.1" 200 -
192.168.118.156 - - [12/Jan/2022 17:13:01] "GET /? HTTP/1.1" 200 -
192.168.118.156 - - [12/Jan/2022 17:13:02] "GET /? HTTP/1.1" 200 -
Entered Here
Entered Here
Entered Here
Entered Here
192.168.118.156 - - [12/Jan/2022 17:13:04] "GET /? HTTP/1.1" 200 -
192.168.118.156 - - [12/Jan/2022 17:13:05] "GET /? HTTP/1.1" 200 -
192.168.118.156 - - [12/Jan/2022 17:13:07] "GET /? HTTP/1.1" 200 -
192.168.118.156 - - [12/Jan/2022 17:13:08] "GET /? HTTP/1.1" 200 -
Entered Here
Entered Here
```

Program 9: Get an alarm from a remote area (through LAN) if smoke is detected. The student should have hands on experience in using various sensors like temperature, humidity, smoke, light, etc. and should be able to use control web camera, network, and relays connected to the Pi.

9a.py:

```
#server program to read gas values
import socket
import Adafruit_MCP3008
import Adafruit_GPIO.SPI as SPI
import time
HOST = '192.168.52.57'
PORT = 3040
SPI_DEVICE = 0
SPI_PORT = 0
mcp = Adafruit_MCP3008.MCP3008(spi=SPI.SpiDev(SPI_PORT, SPI_DEVICE))
try:
    with socket.socket(socket.AF_INET, socket.SOCK_STREAM) as s:
        s.bind((HOST, PORT))
        s.listen()
        conn, addr = s.accept()
        with conn:
            print('Connected by', addr)
            while True:
                value = mcp.read_adc(0)
                print("Gas Value ", value , "units")
                if(value >100):
                    data = "Alert".encode('utf-8')
                    conn.sendall(data)
                time.sleep(3)
```

```
except KeyboardInterrupt:
    s.close()
    GPIO.cleanup()
```

9b.py:

```
#client program to alert through buzzer.
import socket
import RPi.GPIO as GPIO
import time
Buzzer = 36
HOST = '192.168.52.57'
PORT = 3040
GPIO.setmode(GPIO.BOARD)
GPIO.setup(36, GPIO.OUT)
GPIO.setwarnings(False)
try:
    with socket.socket(socket.AF_INET, socket.SOCK_STREAM) as s:
        s.connect((HOST,PORT))
        while True:
            data = s.recv(1024).decode('utf-8')
            print(data)
            if(str(data) == 'Alert'):
                print("ALert! Gas Leakage detected")
                GPIO.output(36, True)
                time.sleep(3)
                GPIO.output(36, False)
                time.sleep(3)
except KeyboardInterrupt:
    s.close()
    GPIO.cleanup()
```

Procedure:

LAN connection: Inter connect the 2 systems (IoT Sensor IFC-02 Board + Raspberry Pi Board) through a Hub with the help of LAN cables.

Install the package Adafruit_MCP3008 in both the systems Host(IoT Sensor IFC-02 Board) and as well as guest(Raspberry Pi Project Board).

Host system (IoT Sensor IFC-02 Board) setup

Enter the command **"sudo pip3 install Adafruit_MCP3008".**

Find out the ip address of the Host system by entering the command **"ifconfig"** note down the inet address. (Ex: inet 192.168.52.57)

Open the program **9a.py** using Thonny editor. Go to the line no. 12 and 13 of the program and modify the inet address which has been noted and the port no. (ex: port No. 3000 or 4000 or5000), save the program.

Guest system (Raspberry Pi Board) setup:

Enter the command **"sudo pip3 install Adafruit_MCP3008".**

Open the program **9b.py** using Thonny editor. Go to the line no. 11 and 12 of the program andmodify the inet address and port no. (ex: port No. 3000 or 4000 or 5000) which should same as the Host system, save the program.

short JP1 in **IoT Sensor IFC-02 Board**

Run the program in the Host as: **sudo python3 9a.py**

Now the console will be blank without any display.

Run the program in the Guest as : **sudo python3 9b.py**

Once the program is executed in Guest system the below message will be displayed on the console on the Host system continuously.

```
Connected by ('169.254.92.230', 46714)
Gas Value 59 units
Gas Value 59 units
```

Note: Take precaution while using the gas lighter.

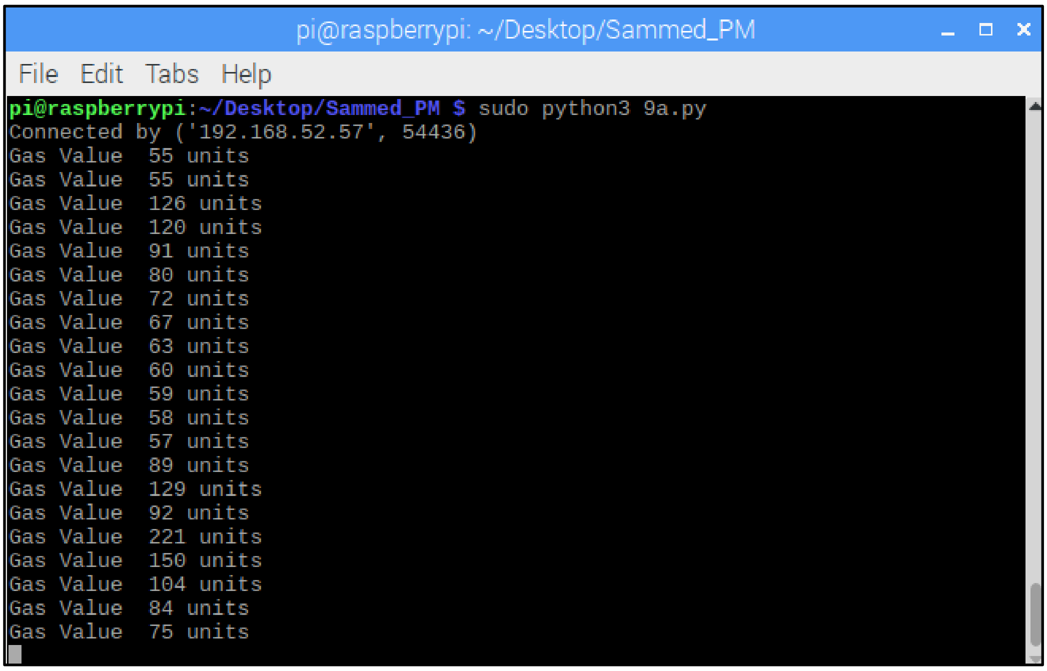
Use any lighter which contains gas, bring the lighter near the sensor (assume that the gas in the lighter is leaking).

When the gas value reaches 300 units and above, the following message on the console of Guest will be displayed, along with the buzzer will be ON and OFF. The buzzer will stop (the ON and OFF process) when the gas value reaches below 300 units.

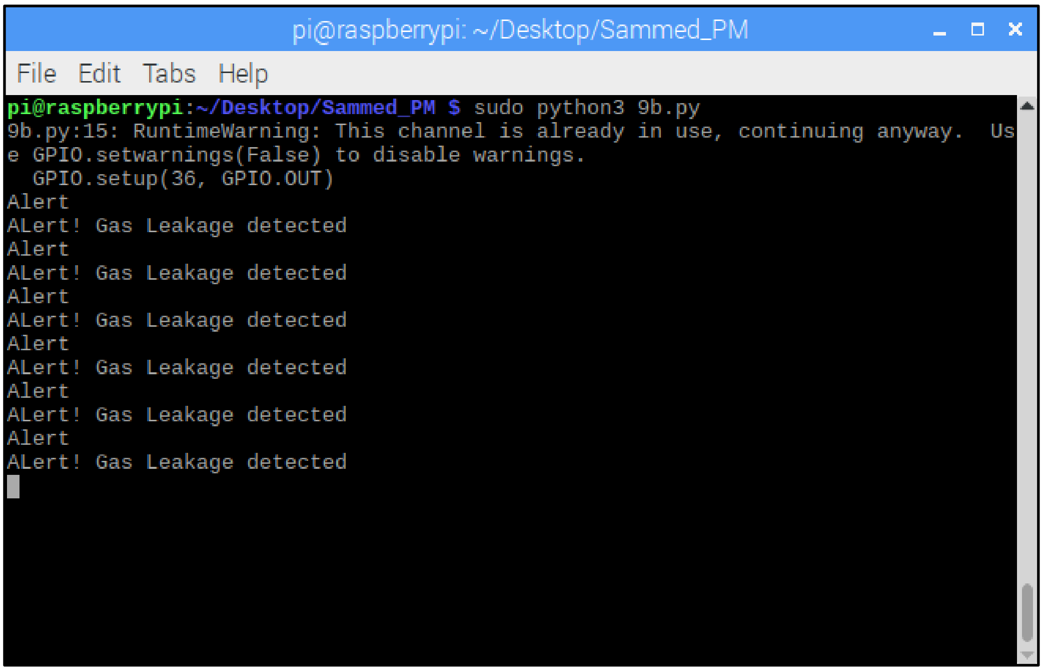
Alert

Alert! Gas Leakage detected

Output:



```
pi@raspberrypi:~/Desktop/Sammed_PM $ sudo python3 9a.py
Connected by ('192.168.52.57', 54436)
Gas Value  55 units
Gas Value  55 units
Gas Value  126 units
Gas Value  120 units
Gas Value  91 units
Gas Value  80 units
Gas Value  72 units
Gas Value  67 units
Gas Value  63 units
Gas Value  60 units
Gas Value  59 units
Gas Value  58 units
Gas Value  57 units
Gas Value  89 units
Gas Value  129 units
Gas Value  92 units
Gas Value  221 units
Gas Value  150 units
Gas Value  104 units
Gas Value  84 units
Gas Value  75 units
```



```
pi@raspberrypi:~/Desktop/Sammed_PM $ sudo python3 9b.py
9b.py:15: RuntimeWarning: This channel is already in use, continuing anyway.  Us
e GPIO.setwarnings(False) to disable warnings.
  GPIO.setup(36, GPIO.OUT)
Alert
ALert! Gas Leakage detected
Alert
ALert! Gas Leakage detected
Alert
ALert! Gas Leakage detected
Alert
ALert! Gas Leakage detected
Alert
ALert! Gas Leakage detected
Alert
ALert! Gas Leakage detected
```