```
**PROGRAM_1**
# A Program to read a name and print Hello <name>
name = input("Please Enter Your Name Here:\n")
print("Hello " + name)


# A Program to read two numbers and display sum, difference, product and division
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("************************************")


# A Program to calculate number of words and characters of a given string
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)
```

```

# A Program to get Area of a selected shape.
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("")


# A Program to print a name n times where name and n values has to be input from standard input
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)
print("")


# A Program to Handle Divide By Zero Exception
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")
print("")


# A Program to print current time with an interval of 10 seconds
import time
for i in range(10):
      seconds = time.time()
      local_time = time.ctime(seconds)
      print("Local time:", local_time)
      time.sleep(10)
print("")


# A Program to Read a file and print No of Words in each Line
file1 = open('myfile.txt', 'r')
Lines = file1.readlines()
i=0
for line in Lines:
      i += 1
      count = len(line.split())
      print("Line ", i, "No of Words ", count)
print("")
```

```
**PROGRAM_2**
# Program to Take 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=0)
gpio.setup(led2,gpio.OUT,initial=0)
gpio.setup(switch1,gpio.IN)
gpio.setup(switch2,gpio.IN)
def glow_led(event):
    if event == switch1 :
        gpio.output(led1, True)
        time.sleep(3)
        gpio.output(led1, False)
    elif event == switch2 :
        gpio.output(led2, True)
        time.sleep(3)
        gpio.output(led2, False)
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()
print("")
```

```
**PROGRAM_3**
#Program to switch ON and Switch off led for corresponding delay times provided
in file.
#OnTime and OffTime values can be read from ledintervals.txt file
import time
import RPi.GPIO as gpio
gpio.setwarnings(False)
gpio.setmode(gpio.BOARD)
led1 =  15
gpio.setup(led1,gpio.OUT,initial=0)
file1 = open('ledintervals.txt', 'r')
Lines = file1.readlines()
ON_TIME = int(Lines[0].split("=")[1])
OFF_TIME = int(Lines[1].split("=")[1])
try:
    while(True):
        gpio.output(led1,True)
        time.sleep(ON_TIME)
        gpio.output(led1,False)
        time.sleep(OFF_TIME)
except KeyboardInterrupt:
    gpio.cleanup()
print("")
```

```
**PROGRAM_4**
# Program to switch on relay at given time using cron.
# use crontab -e.
#*/5 * * * * python3 /home/pi/prog_4.py
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")
print("")
```

```
**PROGRAM_5**
# Program to capture a image form pi camera
#connect a Pi camera. and Enable camera from Raspberry pi configuaration
#All the captured images will be stored at /home/pi/images folder.

```

```
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/images/'+current_date+'.jpg')
camera.stop_preview()
print("Image captured")
```

```
**PROGRAM_6**
# Program to controlling a light source using web page
#install the following command in terminal
#sudo apt-get install python3-flask
# Create a folder "templates" and write your html pages in this folder.
import RPi.GPIO as GPIO
import time
import datetime
led = 13
GPIO.setmode(GPIO.BOARD)
GPIO.setwarnings(False)
GPIO.setup(led, GPIO.OUT,initial=0)
GPIO.setup(led,GPIO.OUT)
from flask import Flask, render_template
app = Flask(__name__)
@app.route('/')
def hello_world():
    return render_template('web.html')
@app.route('/redledon')
def redledon():
    GPIO.output(13, GPIO.LOW)
    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)
    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 = 4000, host='0.0.0.0')


#templates/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://localhost:4000/redledon">
   <input type="submit" value="Red LED On">
</form>
<form action="http://localhost:4000/redledoff">
   <input type="submit" value="Red LED Off">
</form>
</body>
</html>
```

```
**PROGRAM_7**
# Program to implement intruder system that sends an alarm to given gmail.
# Commands to install
#sudo apt-get install ssmtp
#sudo apt-get install mailutils
#Plaese use General email for sending emails as Google is having few security issues.
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 = "<**yourEmailAddress**>"
toaddr = "<toAddress>"
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)
gpio.setup(pir, gpio.IN)
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, "alsdatasave2017")
    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)
```

```
**PROGRAM_8**
# Program to Read Light Status from remote place.
#sudo apt-get install python-flask
import time
import RPi.GPIO as gpio
from flask import Flask, render_template
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('/')
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 = 4000, host='0.0.0.0')

#templates/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://localhost:4000">
   <input type="submit" value="Get Light Status">
</form>
</body>
</html>
```

```
**PROGRAM_9**
#server program to read gas values
#sudo pip3 install Adafruit_MCP3008
import socket
import Adafruit_MCP3008
import Adafruit_GPIO.SPI as SPI
import time
HOST = '127.0.0.1'
PORT = 4000
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 >300):
                    data = "Alert".encode('utf-8')
                    conn.sendall(data)
                time.sleep(3)
except KeyboardInterrupt:
    s.close()
    GPIO.cleanup()

#client program to alert through buzzer.
import socket
import RPi.GPIO as GPIO
import time
Buzzer = 36
HOST = '127.0.0.1'
PORT = 4000
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()
```