1 \*\*PROGRAM\_1\*\*

1. # A Program to read a name and print Hello <name>
2. name = input("Please Enter Your Name Here:\n")
3. print("Hello " + name)

5

6

1. # A Program to read two numbers and display sum, difference, product and division
2. print("Enter First Number")
3. num1 = input()
4. print("Enter Second Number")
5. num2 = input()
6. print("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*")
7. print("Sum of Given numbers is : "+str(int(num1)+int(num2)))
8. print("Difference of Given numbers is : "+str(int(num1)-int(num2)))
9. print("Product of Given numbers is : "+str(int(num1)\*int(num2)))
10. print("Division of Given numbers is : "+str(int(num1)/int(num2)))
11. print("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*")

18

19

1. # A Program to calculate number of words and characters of a given string
2. print("Enter a sentence ")
3. sentence = input()
4. words = sentence.split()
5. word\_count = 0
6. character\_count = 0
7. for word in words:
8. word\_count += 1
9. character\_count += len(word)
10. print("Total Numbers of Words in the sentence are : ",word\_count)
11. print("Total Numbers of characters in the sentence excluding spaces are :
12. ",character\_count)
13. print("Total Numbers of characters in the sentence including spaces are :
14. ",character\_count+word\_count-1)

34

35

1. # A Program to get Area of a selected shape.
2. while True:

38 print("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*")

1. print("Select the Shape that you want to calculate Area")
2. print("""
3. 1. Rectangle
4. 2. Triangle
5. 3. Circle
6. 4. Exit """)
7. choice = input()
8. if(choice == '1'):
9. print("Enter the Width of the Rectangle in meters")
10. width = int(input())
11. print("Enter the height of the Rectangle in meters")
12. height = int(input())
13. print("The area of a Given Rectangle is ", width\*height , " square
14. meters ")
15. continue
16. elif(choice == '2'):
17. print("Enter the Base value of the Triangle in meters")
18. base = int(input())
19. print("Enter the height of the Triangle in meters")
20. height = int(input())
21. print("The area of a Given Rectangle is ", 0.5\*base\*height , " square
22. meters ")
23. continue
24. elif(choice == '3'):
25. print("Enter the Radius of the Circle in meters")
26. radius = int(input())
27. print("The area of a Given Circle is ", 3.14\*radius\*radius , " square
28. meters ")
29. continue
30. elif(choice == '4'):
31. break
32. else:

71

72

73 print("")

74

75

print("Please enter a valid number from the menu") continue

1. # A Program to print a name n times where name and n values has to be input from
2. standard input
3. print("Enter Your name : ")
4. name = input()
5. print("Enter How many times you want to print your name")
6. n = int(input())
7. for i in range(n):
8. print(name)
9. print("")

85

86

1. # A Program to Handle Divide By Zero Exception
2. print("Enter Numerator Value : ")
3. num1 = int(input())
4. print("Enter Denominator value : ")
5. num2 = int(input())
6. try:
7. result = num1/num2
8. print("The Division of Given Numbers is : ", result)
9. except ZeroDivisionError:
10. print("Divide By zero Error. The Denominator should not be Zero")
11. print("")

98

99

1. # A Program to print current time with an interval of 10 seconds
2. import time
3. for i in range(10):
4. seconds = time.time()

|  |  |  |
| --- | --- | --- |
| 104 | local\_time = time.ctime(seconds) |  |
| 105 | print("Local time:", local\_time) |
| 106 | time.sleep(10) |
| 107 | print("") |
| 108 |  |
| 109 |  |
| 110 | # A Program to Read a file and print No | of Words in each Line |
| 111 | file1 = open('myfile.txt', 'r') |  |
| 112 | Lines = file1.readlines() |  |
| 113 | i=0 |  |
| 114 | for line in Lines: |  |
| 115 | i += 1 |  |
| 116 | count = len(line.split()) |  |
| 117 | print("Line ", i, "No of Words ", | count) |
| 118 | print("") |  |

1 \*\*PROGRAM\_2\*\*

* 1. # Program to Take input from two switches and switch on corresponding LEDs.
  2. import time
  3. import RPi.GPIO as gpio
  4. gpio.setwarnings(False)
  5. gpio.setmode(gpio.BOARD)
  6. led1 = 15
  7. led2 = 13
  8. switch1 = 37
  9. switch2 = 35
  10. gpio.setup(led1,gpio.OUT,initial=0)
  11. gpio.setup(led2,gpio.OUT,initial=0)
  12. gpio.setup(switch1,gpio.IN)
  13. gpio.setup(switch2,gpio.IN)
  14. def glow\_led(event):
  15. if event == switch1 :
  16. gpio.output(led1, True)
  17. time.sleep(3)
  18. gpio.output(led1, False)
  19. elif event == switch2 :
  20. gpio.output(led2, True)
  21. time.sleep(3)
  22. gpio.output(led2, False)
  23. gpio.add\_event\_detect(switch1, gpio.RISING , callback = glow\_led, bouncetime = 1)
  24. gpio.add\_event\_detect(switch2, gpio.RISING , callback = glow\_led, bouncetime = 1)
  25. try:
  26. while(True):
  27. time.sleep(1)
  28. except KeyboardInterrupt:
  29. gpio.cleanup()
  30. print("")

1 \*\*PROGRAM\_3\*\*

1. #Program to switch ON and Switch off led for corresponding delay times provided
2. in file.
3. #OnTime and OffTime values can be read from ledintervals.txt file
4. import time
5. import RPi.GPIO as gpio
6. gpio.setwarnings(False)
7. gpio.setmode(gpio.BOARD)
8. led1 = 15
9. gpio.setup(led1,gpio.OUT,initial=0)
10. file1 = open('ledintervals.txt', 'r')
11. Lines = file1.readlines()
12. ON\_TIME = int(Lines[0].split("=")[1])
13. OFF\_TIME = int(Lines[1].split("=")[1])
14. try:
15. while(True):
16. gpio.output(led1,True)
17. time.sleep(ON\_TIME)
18. gpio.output(led1,False)
19. time.sleep(OFF\_TIME)
20. except KeyboardInterrupt:
21. gpio.cleanup()
22. print("")

1 \*\*PROGRAM\_4\*\*

1. # Program to switch on relay at given time using cron.
2. # use crontab -e.
3. #\*/5 \* \* \* \* python3 /home/pi/prog\_4.py
4. import time
5. import RPi.GPIO as gpio
6. gpio.setwarnings(False)
7. gpio.setmode(gpio.BOARD)
8. relay1 = 38
9. gpio.setup(relay1,gpio.OUT,initial=0)
10. try:
11. gpio.output(relay1, True)
12. print("Relay is Switched On. Please Press ctrl+c to exit")
13. time.sleep(15)
14. print("Relay is Switched Off.")
15. gpio.output(relay1, False)
16. except KeyboardInterrupt:
17. gpio.cleanup()
18. print("Program Exited")
19. print("")

1 \*\*PROGRAM\_5\*\*

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

5

1. from picamera import PiCamera
2. from time import sleep
3. import datetime
4. camera = PiCamera()
5. camera.start\_preview()
6. current\_date = datetime.datetime.now().strftime('%d-%m-%Y %H:%M:%S')
7. sleep(3)
8. camera.capture('/home/pi/Desktop/images/'+current\_date+'.jpg')
9. camera.stop\_preview()
10. print("Image captured")

1 \*\*PROGRAM\_6\*\*

1. # Program to controlling a light source using web page
2. #install the following command in terminal
3. #sudo apt-get install python3-flask
4. # Create a folder "templates" and write your html pages in this folder.
5. import RPi.GPIO as GPIO
6. import time
7. import datetime
8. led = 13
9. GPIO.setmode(GPIO.BOARD)
10. GPIO.setwarnings(False)
11. GPIO.setup(led, GPIO.OUT,initial=0)
12. GPIO.setup(led,GPIO.OUT)
13. from flask import Flask, render\_template
14. app = Flask( name )
15. @app.route('/')
16. def hello\_world():
17. return render\_template('web.html')
18. @app.route('/redledon')
19. def redledon():
20. GPIO.output(13, GPIO.LOW)
21. now = datetime.datetime.now()
22. timeString = now.strftime("%Y-%m-%d %H:%M")
23. templateData = {
24. 'status' : 'ON',
25. 'time': timeString
26. }
27. return render\_template('web.html', \*\*templateData)
28. @app.route('/redledoff') #Route for Turning RedLed Off
29. def redledoff():
30. GPIO.output(13, GPIO.HIGH)
31. now = datetime.datetime.now()
32. timeString = now.strftime("%Y-%m-%d %H:%M")
33. templateData = {
34. 'status' : 'OFF',
35. 'time': timeString
36. }
37. return render\_template('web.html', \*\*templateData)
38. if

40

41

42

name == " main ":

app.run(debug = True, port = 4000, host='0.0.0.0')

1. #templates/web.html
2. <html>
3. <body>
4. <title>Raspberry PI Remote Control</title>
5. <h1>Raspberry PI Remote Control</h1>
6. <h2>Light Status : {{status}}, Last Modified : {{time}}</h2>
7. <form action="http://localhost:4000/redledon">
8. <input type="submit" value="Red LED On">
9. </form>
10. <form action="http://localhost:4000/redledoff">
11. <input type="submit" value="Red LED Off">
12. </form>
13. </body>
14. </html>

1 \*\*PROGRAM\_7\*\*

1. # Program to implement intruder system that sends an alarm to given gmail.
2. # Commands to install
3. #sudo apt-get install ssmtp
4. #sudo apt-get install mailutils
5. #Plaese use General email for sending emails as Google is having few security
6. issues.
7. import RPi.GPIO as gpio
8. import picamera
9. import time
10. import smtplib
11. from email.mime.multipart import MIMEMultipart
12. from email.mime.text import MIMEText
13. from email.mime.base import MIMEBase
14. from email import encoders
15. from email.mime.image import MIMEImage
16. fromaddr = "<\*\*yourEmailAddress\*\*>"
17. toaddr = "<toAddress>"
18. mail = MIMEMultipart()
19. mail['From'] = fromaddr
20. mail['To'] = toaddr
21. mail['Subject'] = "Attachment"
22. body = "Please find the attachment
23. led=15
24. pir=12
25. HIGH=1
26. LOW=0
27. gpio.setwarnings(False)
28. gpio.setmode(gpio.BOARD)
29. gpio.setup(led, gpio.OUT)
30. gpio.setup(pir, gpio.IN)
31. data=""
32. def sendMail(data):
33. mail.attach(MIMEText(body, 'plain'))
34. print(data)
35. dat='%s.jpg'%data
36. print(data)
37. attachment = open(dat, 'rb')
38. image=MIMEImage(attachment.read())
39. attachment.close()
40. mail.attach(image)
41. server = smtplib.SMTP('smtp.gmail.com', 587)
42. server.starttls()
43. server.login(fromaddr, "alsdatasave2017")
44. text = mail.as\_string()
45. server.sendmail(fromaddr, toaddr, text)
46. server.quit()
47. def capture\_image():
48. data= time.strftime("Image was captured on %H:%M:%S|%d\_%b\_%Y")
49. camera.start\_preview()
50. time.sleep(5)
51. print(data)
52. camera.capture('%s.jpg'%data)
53. camera.stop\_preview()
54. time.sleep(1)
55. sendMail(data)
56. gpio.output(led , 0)
57. camera = picamera.PiCamera()
58. camera.rotation=180
59. camera.awb\_mode= 'auto'
60. camera.brightness=55
61. while 1:
62. if gpio.input(pir)==1:
63. gpio.output(led, HIGH)
64. capture\_image()
65. while(gpio.input(pir)==1):
66. time.sleep(1)
67. else:
68. gpio.output(led, LOW)
69. time.sleep(0.01)

1 \*\*PROGRAM\_8\*\*

* 1. # Program to Read Light Status from remote place.
  2. #sudo apt-get install python-flask
  3. import time
  4. import RPi.GPIO as gpio
  5. from flask import Flask, render\_template
  6. import datetime
  7. app = Flask( name )
  8. gpio.setwarnings(False)
  9. gpio.setmode(gpio.BOARD)
  10. led1 = 13
  11. switch1 = 35
  12. gpio.setup(led1,gpio.OUT,initial=1)
  13. gpio.setup(switch1,gpio.IN)
  14. light\_status = "OFF"
  15. def glow\_led(event):
  16. print("Entered Here")
  17. global light\_status
  18. if event == switch1 and light\_status == "OFF":
  19. gpio.output(led1, False)
  20. light\_status = "ON"

22

1. elif event == switch1 and light\_status == "ON":
2. gpio.output(led1, True)
3. light\_status = "OFF"
4. @app.route('/')
5. def ledstatus():
6. now = datetime.datetime.now()
7. timeString = now.strftime("%H:%M %d-%m-%Y")
8. templateData = {
9. 'status' : light\_status,
10. 'time': timeString
11. }
12. return render\_template('lightstatus.html', \*\*templateData)
13. gpio.add\_event\_detect(switch1, gpio.RISING , callback = glow\_led, bouncetime =
14. 100)
15. app.run(debug = True, port = 4000, host='0.0.0.0')

38

1. #templates/lightstatus.html
2. <html>
3. <body>
4. <title>Raspberry PI Remote Light Status</title>
5. <h1>Raspberry PI Remote Control</h1>
6. <h2>Light Status : {{status}}, Last Seen : {{time}}</h2>
7. <form action="http://localhost:4000">
8. <input type="submit" value="Get Light Status">
9. </form>
10. </body>
11. </html>

1 \*\*PROGRAM\_9\*\*

1. #server program to read gas values
2. #sudo pip3 install Adafruit\_MCP3008
3. import socket
4. import Adafruit\_MCP3008
5. import Adafruit\_GPIO.SPI as SPI
6. import time
7. HOST = '127.0.0.1'
8. PORT = 4000
9. SPI\_DEVICE = 0
10. SPI\_PORT = 0
11. mcp = Adafruit\_MCP3008.MCP3008(spi=SPI.SpiDev(SPI\_PORT, SPI\_DEVICE))
12. try:
13. with socket.socket(socket.AF\_INET, socket.SOCK\_STREAM) as s:
14. s.bind((HOST, PORT))
15. s.listen()
16. conn, addr = s.accept()
17. with conn:
18. print('Connected by', addr)
19. while True:
20. value = mcp.read\_adc(0)
21. print("Gas Value ", value , "units")
22. if(value >300):
23. data = "Alert".encode('utf-8')
24. conn.sendall(data)
25. time.sleep(3)
26. except KeyboardInterrupt:
27. s.close()
28. GPIO.cleanup()

30

1. #client program to alert through buzzer.
2. import socket
3. import RPi.GPIO as GPIO
4. import time
5. Buzzer = 36
6. HOST = '127.0.0.1'
7. PORT = 4000
8. GPIO.setmode(GPIO.BOARD)
9. GPIO.setup(36, GPIO.OUT)
10. GPIO.setwarnings(False)
11. try:
12. with socket.socket(socket.AF\_INET, socket.SOCK\_STREAM) as s:
13. s.connect((HOST,PORT))
14. while True:
15. data = s.recv(1024).decode('utf-8')
16. print(data)
17. if(str(data) == 'Alert'):
18. print("ALert! Gas Leakage detected")
19. GPIO.output(36, True)
20. time.sleep(3)
21. GPIO.output(36, False)
22. time.sleep(3)
23. except KeyboardInterrupt:
24. s.close()
25. GPIO.cleanup()