

TITLE : Implement a python script

Expt No. : 4b

to check the element is in the
list or not by using linear search
or Binary Search.

Date : 26/11/2022

Source Code :-

for linear search:-

$l = [10, 23, 4, 56, 7]$

Key = int(input("Enter number to search:"))

found = False

for i in range(len(l)):

if l[i] == key:

found = True

print("found at index", i)

break

if found == False:

print("not found")

Output

Enter number to search: 5

not found

Marks :

Staff :

Another method for linear Search

```
l = []
n = int(input("enter size of list:"))
for i in range(n):
    num = int(input("enter the elements:"))
    l.append(num)
key = int(input("enter number to search:"))
found = False
for i in range(len(l)):
    if l[i] == key:
        found = True
        print("found at index", i)
        break
if found == False:
    print("not found")
```

Output

```
enter size of list: 5
enter the elements: 1
enter the elements: 3
enter the elements: 4
enter the elements: 2
enter number to search: 4
found at index: 2
```

For binary search:

```
def binary_search(lst, key):
    low = 0
    high = len(lst) - 1
    while low <= high:
        mid = (high + low) // 2
        if (key == lst[mid]):
            return mid
        elif (key > lst[mid]):
            low = mid + 1
        else:
            high = mid - 1
lst = []
n = int(input("Enter size of list:"))
for i in range(n):
    num = int(input("Enter any number:"))
    lst.append(num)
key = int(input("Enter number to search:"))
index = binary_search(lst, key)
if (index < 0):
    print("{} was not found.".format(key))
else:
    print("{} was found at index {}.".format(key, index))
```

Output

Enter size of list: 4
Enter any number: 1
Enter any number: 2
Enter any number: 3

Enter number to search: 3
3 was found at index 2.

TITLE : Implement a python script

Expt No. : 4C

to arrange the elements in

Date : _____

sorted order using Bubble, Selection, Insertion
and Merge sorting techniques.

source code for bubble Sort:-

```
def bubble_sort(lst):
```

```
    n = len(lst)
```

```
    for i in range(n-1):
```

```
        for j in range(0, n-i-1):
```

```
            if lst[j] < lst[j+1]:
```

```
                lst[j], lst[j+1] = lst[j+1], lst[j]
```

```
lst = []
```

```
n = int(input("Enter size of list:"))
```

```
for i in range(n):
```

```
    num = int(input("Enter any number:"))
```

```
    lst.append(num)
```

```
print("List elements before sorting:\n")
```

```
for i in range(len(lst)):
```

```
    print(lst[i], end=' ')
```

```
bubble_sort(lst)
```

```
print("\nList elements after sorting:\n")
```

```
for i in range(len(lst)):
```

```
    print(lst[i], end=' ')
```

Marks :

Staff :

Output:- (bubble sort)

Enter size of list: 5

Enter any number: 12

Enter any number: 4

Enter any number: 56

Enter any number: 7

Enter any number: 8

Enter any number

List elements before sorting:

12, 4, 56 7 8

List elements after sorting:

56, 12 8 7 4

Output:- (Selection Sort)

Enter size of list: 5

Enter any number: 12

Enter any number: 3

Enter any number: 45

Enter any number: 6

Enter any number: 8

List elements before sorting:

12 3 45 6 8

List elements after sorting:

3 6 8 12 45

TITLE : Selection Sort

Expt No. : 4C

Date :

```
def selection_sort(lst):
    n = len(lst) - 1
    for i in range(0, n-1):
        smallest = i
        for j in range(i+1, len(lst)):
            if lst[j] < lst[smallest]:
                smallest = j
        lst[i], lst[smallest] = lst[smallest], lst[i]

lst = []
n = int(input("Enter size of list:"))
for i in range(n):
    num = int(input("Enter any number:"))
    lst.append(num)
print("List elements before sorting:")
for i in range(len(lst)):
    print(lst[i], end=' ')
selection_sort(lst)
print("List elements after sorting:")
for i in range(len(lst)):
    print(lst[i], end=' ')
```

Marks :

Staff :

Output (Insertion Sort) :-

Enter size of list: 5

Enter any numbers: 12

Enter any number: 3

Enter any number: 45

Enter any number: 6

Enter any number: -4

Enter any number

List elements before sorting:

12 3 45 6 -4

List elements after sorting:

-4 3 6 12 45

TITLE : insertion sort

Expt No. : 4C

Date : _____

```
def insertion_sort(lst)
    for i in range(1, len(lst)):
        temp = lst[i]
        j = i - 1
        while (j) >= 0 and temp < lst[j]:
            lst[j + 1] = lst[j]
            j = j - 1
        lst[j + 1] = temp
    lst = []
n = int(input("Enter size of list: "))
for i in range(n):
    num = int(input("Enter any number: "))
    lst.append(num)
print("List elements before sorting: ", lst)
for i in range(len(lst)):
    print(lst[i], end=' ')
print("\nList elements after sorting: ", lst)
for i in range(len(lst)):
    print(lst[i], end=' ')
```

Marks :

Staff :



Scanned with OKEN Scanner

Output: (Merge Sort)

Enter size of list: 10

Enter any number: 10

Enter any number: 5

Enter any number: 45

Enter any number: 6

Enter any number: 78

Enter any number: 9

Enter any number: 11

splitting [10, 5, 45, 6, 78, 9, 11]

Splitting [10, 5, 45]

splitting [10]

Merging [10]

splitting [5, 45]

splitting [5]

Merging [5]

splitting [45]

Merging [45]

Merging [5, 45]

Merging [5, 10, 45]

splitting [6, 78, 9, 11]

splitting [6, 78]

splitting [6]

Merging [6]

splitting [78]

Merging [78]

Merging [6, 78]

splitting [9, 11]

splitting [9]

Merging [9]

splitting [11]

Merging [11]

Merging [9, 11]

Merging [6, 9, 11, 78]

Merging [5, 6, 9, 10, 11, 45, 78]

[5, 6, 9, 10, 11, 45, 78]

TITLE : Merge Sort

Expt No. : 4c

Date :

Source code :-

```
def mergesort(nlist):
    print("Splitting ", nlist)
    if len(nlist) > 1:
        mid = len(nlist) // 2
        lefthalf = nlist[:mid]
        righthalf = nlist[mid:]
```

```
        mergesort(lefthalf)
        mergesort(righthalf)
```

i=j=k=0

```
while i < len(lefthalf) and j < len(righthalf):
    if lefthalf[i] < righthalf[j]:
        nlist[k] = lefthalf[i]
        i = i + 1
```

else:

```
    nlist[k] = righthalf[j]
    j = j + 1
```

k = k + 1

```
while i < len(lefthalf):
```

```
nlist[k] = lefthalf[i]
```

i = i + 1

k = k + 1

```
while j < len(righthalf):
```

```
nlist[k] = righthalf[j]
```

j = j + 1

Marks :

Staff :

TITLE : Merge Sort

Expt No. : 4C

Date :

```
K = K + 1
print("Merging ", nlist)
nlist = []
n = int(input("Enter size of list: "))
for i in range(n):
    num = int(input("Enter any number: "))
    nlist.append(num)
mergeSort(nlist)
print(nlist)
```

Marks :

TITLE : Write python program using
NumPy and Pandas:

Expt No. : 63

Date :

2. Write a Pandas program to create and display a DataFrame from a specified dictionary data which has the index labels.

Source code:-

```
import pandas as pd
student_data = {'Name': ['Sachin', 'AKshay', 'Lakshay',
'Suhani', 'Saurav', 'Chetan'],
'USN': ["18BTR101", "18BTR102", "18BTR103",
"18BTR104", "18BTR105", "18BTR106"],
'Marks': [99, 93, 94, 95, 96, 98],
'Program': ['B.Tech', 'M.Tech', 'B.Tech', 'M.Tech',
'B.Tech', 'M.Tech']}
labels = ['a', 'b', 'c', 'd', 'e', 'f']
df = pd.DataFrame(student_data, index=labels)
print(df)
```

Marks :

Staff :

Output (62)

	Name	USN	MARKS	Program
a	Sachin	18BTR101	90	B-Tech
b	AKshara	18BTR102	93	M-Tech
c	Lara	18BTR103	94	B-Tech
d	Suhani	18BTR104	95	M-Tech
e	Saurav	18BTR105	96	B-Tech
f	Chetan	18BTR106	98	M-Tech

TITLE : To create a 4×2 integer array and prints its attributes.

Expt No. : 6b

Date :

Source code:

```
import numpy as np
array1 = np.empty([4,2], dtype=np.int64)
print("----")
print("printing Array1(empty array) ")
print(array1)
print("----")
array2 = np.zeros([4,2], dtype=np.int64)
print("printing Array2(zero filled) ")
print("----")
print(array2)
print("----")
array3 = np.array([[1,2],[3,4],[5,6],[7,8]], dtype=
np.int64)
print("printing Array3")
print(array3)
print("----")
```

Marks :

Staff :

Output: (6b)

printing Array (empty array)
[[]]
[[4592 267800]]
[[8390634266]]
[[3144069482]]
[[1720310080]]

printing Array (zero filled)
[[0 0]]
[[0 0]]
[[0 0]]
[[0 0]]

printing Array

[[1, 2]]
[[3, 4]]
[[5, 6]]
[[7, 8]]

TITLE : For the given numpy array
return array of odd rows
and even columns.

Expt No. : 6C

Date :

Source code:

```
import numpy as np
array = np.array([[1, 2, 3, 4, 5],
                  [6, 7, 8, 9, 10],
                  [11, 12, 13, 14, 15],
                  [16, 17, 18, 19, 20],
                  [21, 22, 23, 24, 25],
                  [26, 27, 28, 29, 30]])

print("\n Given array is:\n")
print(array)

print("\n Odd rows of array is:\n")
for i in range(array.shape[0]):
    if i % 2 == 0:
        print(array[i, :])

print("\n Even rows of array is:\n")
for i in range(array.shape[0]):
    if i % 2 != 0:
        print(array[i, :])
```

Marks :

Staff :

Output: (6c)

Given array is:
[[1, 2, 3, 4, 5]
 [6, 7, 8, 9, 10]
 [11, 12, 13, 14, 15]
 [16, 17, 18, 19, 20]
 [21, 22, 23, 24, 25]
 [26, 27, 28, 29, 30]]

Odd rows of array is:

[1 2 3 4 5]
 [11 12 13 14 15]
 [21 22 23 24 25]

Even rows of array is:

[6 7 8 9 10]
 [16 17 18 19 20]
 [26 27 28 29 30]

Even columns of array is:

[2 7 12 17 22 27]
 [4 9 14 19 24 29]

Odd columns of array is:

[1 6 11 16 21 26]
 [3 8 13 18 23 28]
 [5 10 15 20 25 30]

TITLE : _____

Expt No. : 6C

Date : _____

print("In Even columns of array is: \n")
for i in range(arry.shape[1]):

if i % 2 == 1:

print(arry[:, i])

print("In odd columns of array is: \n")
for i in range(arry.shape[1]):

if i % 2 != 1:

print(arry[:, i])

Marks :

Output (6d):

$\begin{bmatrix} [1. 2. 3] \end{bmatrix}$

$\begin{bmatrix} [4. 5. 6] \end{bmatrix}$

$\begin{bmatrix} [7. 8. 9] \end{bmatrix}$

$\begin{bmatrix} [1. 2. 3] \end{bmatrix}$

$\begin{bmatrix} [4. 5. 6] \end{bmatrix}$

$\begin{bmatrix} [7. 8. 9] \end{bmatrix}$

Addition of two arrays is (using +)

$\begin{bmatrix} [2. 4. 6] \end{bmatrix}$

$\begin{bmatrix} [8. 10. 12] \end{bmatrix}$

$\begin{bmatrix} [14. 16. 18] \end{bmatrix}$

Addition of two arrays is (using add())

$\begin{bmatrix} [2. 4. 6] \end{bmatrix}$

$\begin{bmatrix} [8. 10. 12] \end{bmatrix}$

$\begin{bmatrix} [14. 16. 18] \end{bmatrix}$

Square root of each element of matrix

$\begin{bmatrix} [1.41421356 \quad 2. \quad 2.44948974] \end{bmatrix}$

$\begin{bmatrix} [2.82842712 \quad 3.16227766 \quad 3.46410162] \end{bmatrix}$

$\begin{bmatrix} [3.74165735 \quad 4. \quad 4.24264069] \end{bmatrix}$

TITLE : To add the two Numpy arrays

Expt No. : 6d

and modify a result array by calculating the square root of each element.

Date :

Source code

```
a=np.array([[1,2,3],[4,5,6],[7,8,9]])
```

```
b=np.array([[1,2,3],[4,5,6],[7,8,9]])
```

```
print(a)
```

```
print(b)
```

```
c=a+b
```

```
print("Addition of two arrays is (using +)\n")
```

```
print(c)
```

```
d=np.add(a,b)
```

```
print("\nAddition of two arrays is (using add())\n")
```

```
print(d)
```

```
s=np.sqrt(d)
```

```
print("\nSquare root of each element of matrix\n")
```

```
print(s)
```

Marks :

Staff :

TITLE : Write python programs
for following graphical objects.

Expt No. : 8 (a)
Date :

2. Draw the target symbols (8 set of concentric squares, alternating red and white) in a graphics window, that is, 200 pixels wide by 200 pixels high.
(Hint: Draw the target square first in red, followed by next small square in white, then draw the next smaller square in red).

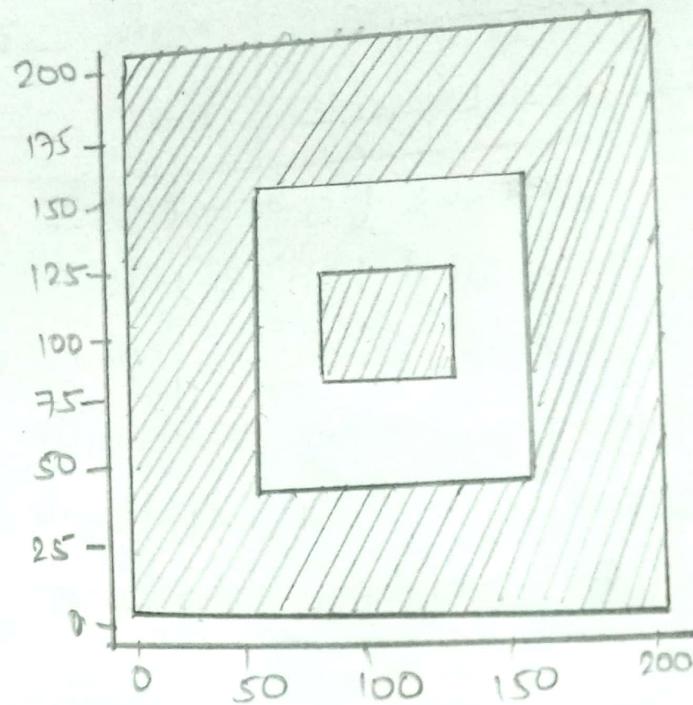
source code:-

```
import matplotlib.pyplot as plt
rect=plt.Rectangle((0,0), 200, 200, fc='r')
plt.gca().add_patch(rect)
rect=plt.Rectangle((150,150), 100, 100, fc='w')
plt.gca().add_patch(rect)
rect=plt.Rectangle((75,75), 50, 50, fc='r')
plt.gca().add_patch(rect)
plt.axis('scaled')
plt.show()
```

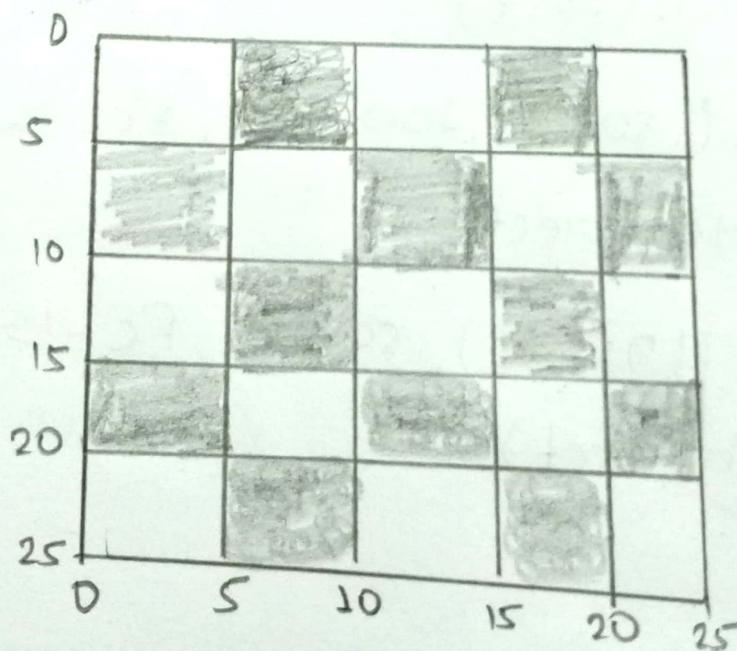
Marks :

Staff :

Output: (8a)



Output: 8b



TITLE : Create a 5×5 rectangle whose top left corner is at $(\text{row}*5, \text{col}*5)$.

Expt No. : 8b

Date :

If the sum of the rows and columns' numbers is even, set the fill color of the rectangle to white, otherwise set it to the black. Then draw the rectangle.

source code:

```
import matplotlib.pyplot as plt
plt.figure()
plt.axis([0, 25, 25, 0])
cursor=plt.gca()
for i in range(0,5):
    for j in range(0,5):
        if ((i+j)%2 == 0):
            cursor.add_patch(plt.Rectangle(i*5, j*5),
5,5,color='white'))
        else:
            cursor.add_patch(plt.Rectangle((i * 5, j * 5),
5,5,color='black'))
plt.show()
```

Marks :

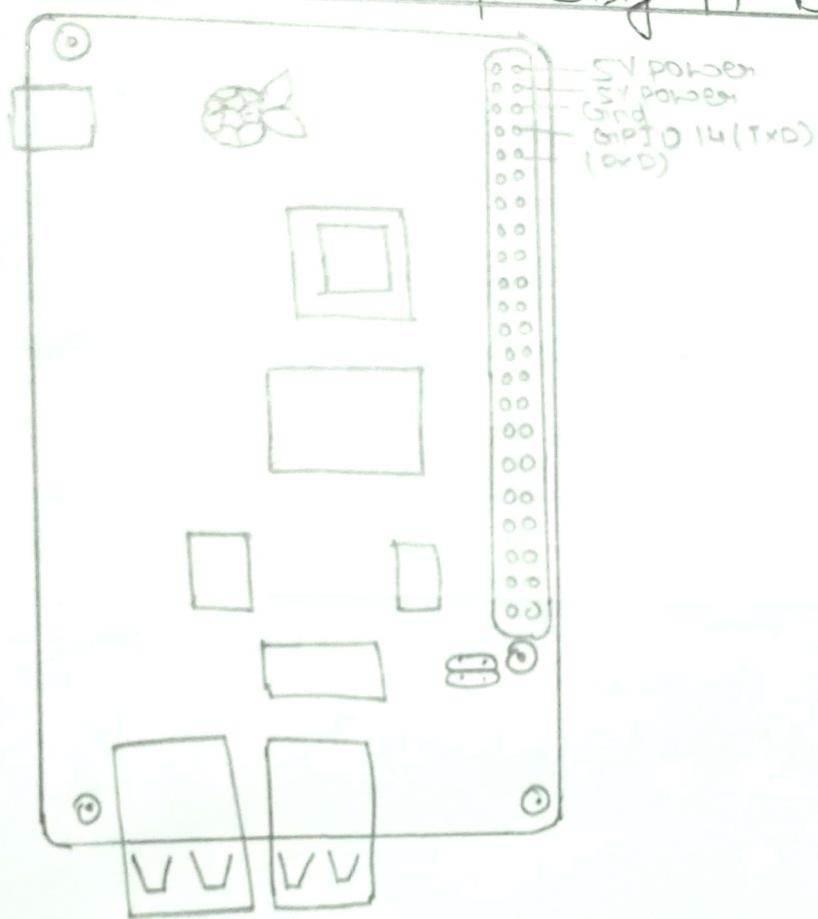
Staff :

TITLE : Raspberry Pi

Expt No. : 9

Date :

a) Introduction to Raspberry Pi board



One of the powerful features of the Raspberry Pi is the row of GPIO (general-purpose input output) pins and the GPIO pinout is an interactive reference to these GPIO pins.

b) Setting up Python on Raspberry Pi board.

The easiest introduction to Python is through Thonny. The Python 3 development environment. You can open Thonny from the desktop or applications menu.

Marks :

Staff :

c) Python files in Thonny

= To create a python file in Thonny, click files New and you'll be given a window. This is an empty file, not a python prompt.

We write a python file in this window, save it then run it and we'll see the output in the other windows.

Then save this file (file > save or $ctrl+s$) and run (run > Run Module or hit F5) and we'll see the output in our original python window.

d) Printing Hello World from file.

⇒ Hello world in Python:

```
print("Hello World")
```

TITLE : Digital Input and Output

Expt No. : 10

Date :

a) Interface Tactile button to Raspberry Pi board using Python.

source code :-

```
import RPi.GPIO as GPIO
```

```
import time
```

```
button = 16
```

```
led = 18
```

```
def setup():
```

```
    GPIO.setmode(GPIO.BOARD)
```

```
    GPIO.setup(button, GPIO.IN, pull_up-  
down=GPIO.PUD_UP)
```

```
    GPIO.setup(led, GPIO.OUT)
```

```
def loop():
```

```
    while True:
```

```
        button_state = GPIO.input(button)
```

```
        if button_state == False:
```

```
            GPIO.output(led, True)
```

```
            print('Button Pressed...')
```

```
        while GPIO.input(button) == False:
```

```
            time.sleep(0.2)
```

```
    else:
```

```
        GPIO.output(led, False)
```

```
def endprogram():
```

```
    GPIO.output(led, False)
```

```
    GPIO.cleanup()
```

Marks :

Staff :

```
if __name__ == '__main__':
    setup()
    try:
        loop()
    except KeyboardInterrupt:
        print('Keyboard interrupt detected')
        endprogram()
```

Working:

The working of Raspberry Pi Push Button interface is very easy to understand. When the Python script is run, the Raspberry Pi initializes the button pin as input with internal pull-up and LED pin as output.

Now, it waits for a change in state of the input pin, which happens only when the button is pressed. If the button is pushed, Raspberry Pi will detect a low on the corresponding pin and activates the LED.

TITLE : Blink LED on and OFF

Expt No. : 10(b)

Date : _____

= for blinking LED on and off following commands in Terminal session:-

sudo apt-get update

sudo apt-get install python3-pip

sudo pip3 install gpiozero

Run the following code on the Thonny Python IDE in Raspberry Pi:

Source code :-

```
from gpiozero import LED #import LED function
from time import sleep #import sleep function
led = LED(27) # assign gpio27 to a variable.
while True: #use a 'while' loop
    led.on() #switch on the led
    sleep(2) #wait for 2 seconds
    led.off() # switch off the led
    sleep(2) #wait for 2 seconds
```

- Small pin of led connected to ground and long pin of led GPIO 27 (pin 13) ^{Pin 6}

Marks :

Staff :

10c) Sound buzzes

⇒ Run the following commands in Terminal

sudo apt-get update

Sudo apt-get install python3-pip

Sudo pip3 install gpiozero

Sudo pip3 install gpiod

Run the following code on the Thonny Python IDE in Raspberry Pi:

Source code:

```
from gpiozero import BUZZER #import buzzer function
from time import sleep #import sleep function
zed = BUZZER(27) #assign GPIO27 to a variable
while True: #use a 'while' loop
    zed.on() #switch on the buzzer
    sleep(2) #wait for 2 seconds
    zed.off() #switch off the buzzer
    sleep(2) #wait for 2 seconds.
```

⇒ Connection same as led interfacing.

TITLE : connecting PIR Motion sensor to Raspberry Pi board

Expt No. : 10d

Date : _____

Procedure:-

Components Required:

1. Raspberry Pi
2. PIR Sensor
3. 5V Buzzer
4. connecting wires
5. Mini Breadboard
6. Power supply

source code:-

```
import RPi.GPIO as GPIO
import time
sensor = 16
buzzer = 18
GPIO.setmode(GPIO.BCM)
GPIO.setup(sensor, GPIO.IN)
GPIO.setup(buzzer, GPIO.OUT)
GPIO.output(buzzer, False)
print "Initializing PIR Sensor...."
time.sleep(12)
print "PIR Ready..."
print "
```

Marks :

Staff :

try:

while True:

if GPIO.input(sensor):
GPIO.output(buzzer, True)

print "Motion detected"

while GPIO.input(sensor):
time.sleep(0.2)

else:

GPIO.output(buzzer, False)

except KeyboardInterrupt:

GPIO.cleanup()

TITLE :

Image Recognition

Expt No. : 12

Date :

- a. Interfacing Camera Module
- b. Capture Images
- c. Capture Videos.

Procedure:

The python picamera module is currently not, by default, compatible with the latest version of Raspberry Pi OS (called Bullseye).

To use the picamera module, we will need to enable legacy support for the camera. Open a terminal window and type the following command:

```
sudo raspi-config
```

Use the cursor keys to scroll down to Interface options and press the 'Enter' key.

Make sure 'Legacy camera enable / disable legacy camera support' is selected and press the 'Enter' key.

Use the cursor keys to select <Yes> and press the 'Enter' key.

Marks :

Staff :

Press 'Enter' again to continue confirm.

Use the cursor keys to select <Finish>

Press 'Enter' to reboot.

- Go to the main menu and open the Raspberry Pi configuration tool. Select the Interfaces tab and ensure that the camera is enabled. Reboot our Raspberry Pi.

- Open a terminal window by clicking the black monitor icon in the taskbar.

- Type in the following command to take a still picture and save it to the desktop:

```
raspistill -o Desktop/image.jpg
```

add -h and -w to change the height and width of the image:

```
raspistill -o Desktop/image-small.jpg -w 640 -h 480
```

Now record a video with the camera module by using the following raspivid command:

```
raspivid -o Desktop/video.h264
```

In order to play the video file, double-click the video.h264 file icon on the Desktop to open it in VLC Media player.

TITLE : capture image and
video

Expt No. : 12 b, c

Date :

How to control the camera module with Python code:-

Open a Python 3 editor, such as Thonny Python IDE:

Open a new file and save it as captureimage.py

Enter the following code for capturing images:

```
from picamera import PiCamera
```

```
from time import sleep
```

```
camera = PiCamera()
```

```
camera.start_preview()
```

```
sleep(5)
```

```
camera.capture('home/pi/Desktop/images.jpg')
```

```
camera.stop_preview()
```

Open a new file and save it as capturevideos.py

Enter the following code for capturing videos:

```
from picamera import PiCamera
```

```
from time import sleep
```

```
camera = PiCamera()
```

```
camera.start_preview()
```

Marks :

Staff :

```
camera.start_recording('~/Desktop/video.h264')
sleep(5)
camera.stop_recording()
camera.stop_preview()
```

Qd) Transmission of Data | Analysis Example:-

Step-1: Connect your PIR Sensor to GPIO 4

Step-2: Connect a Raspberry Pi Camera Module.

Source code

```
from gpiozero import MotionSensor
from picamera import picamera
from datetime import datetime
pir = MotionSensor(4)
camera = picamera()
filename = "intruder.h264"
```

While True:

```
filename = "01.01.19 - 01.01.01.h264"
pir.wait_for_motion()           (now)
print("Motion detected!")
camera.start_recording(filename)
pir.wait_for_no_motion()
camera.stop_recording()
```