

PRACTICAL - I

Program :-

```
a = open ("sample.txt", "w")
a.write ("This is Python" + "\n")
a.write ("Python Version 2.7 \n Python is untyped language")
a.close
```

- (A) Aim : To create a file

Algorithm:

- Step 1 :- Create a file object using the write method.
- Step 2 :- Using write access mode write contents in it.
- Step 3 :- Close the file when not needed.

- (B) Aim : Open the file in read mode and use the read, readline and readlines methods & store the O/P in the variable

(B) Algorithm :

Program :

(B) # using read method
 a = open ("sample.txt", "r")

```
step1 = a.read()
print ("The output of read method is", step1)
a.close.
```

Step 3 :- Close the file

SS0
using readline method

a = open ("sample.txt", "r")

str = a.readline()
print ("The output of readline method is ", str)

a.close()

Using readlines method

a = open ("sample.txt", "r")

str = a.readlines()

print ("The output of readlines method is ", str)

a.close()

Algorithm:

Step 1 : Open the file in read mode

Step 2 : Using the file object use a.closed , a.mode
a.softspace

Step 3 : Close the file

Program:

a = open ("Sample.txt", "r")

b = a.closed

print (b)

c = a.mode

print (c)

d = a.softspace

print (d)

Output:
True

0

Q80

Program
a = open ("sample.txt", "r")

Aim : To implement tell() and seek() method
a.read()
a.readline()
a.tell()
a.seek()

Algorithm:

Step 1 : Open the file using the file object in the read mode

Step 2 : Using the variable pos implement the tell method and print it.

Step 3 : Using str use seek method & print the value.

Ex: : using file operation\read\individual elements from a file

Output:

70

70

Program :

```
a = open("sample.txt", "r")
```

```
str1 = a.readline()
```

```
for line in str1:
```

```
    print(len(line))
```

Output :

50

Aim : Using conditional statement print the length of a line

Algorithm :

Step 1 : Open the file in the read mode.

Step 2 : Read the line using the readlines() and assign a variable to it.

Program :

```
a = open("sample.txt", "r")
```

```
str1 = a.read()
```

```
str = input("Enter a word ")
```

```
count = 0
```

for word in str1:

```
if (word == "the"):
```

```
    print(count + 1)
```

```
else
```

```
    print(count)
```

Output :

5

Aim : To find frequency of occurrence of given word

Step 1 : Open the file and read it using read() method

Step 2 : Initialise a counter

if (word == "the"):

print(count + 1)

else

print(count)

Step 3 : use the for conditional statement followed by if conditional statement to find the frequency

Step 4 : Increment the counter if the condition satisfied and print the value of counter

Ques : Inserting special character in content of file

With open ("sample.txt", "w") as f :

```
s = 1
c = g.read(2)
while len(c) > 0 :
    print(c, end = "\t")
    c = g.read(2)
```

Step 1 : Open a file using conditional statement

Step 2 : Store the read file method in some user defined variable.

Step 3 : Using the while statement check whether the length of variable is greater than zero.

Step 4 : Print the variable.

Output :

Henry, it's time to meet with the committee.

~~last~~
~~Dan~~
~~alice~~

```

mytuple = ("Neeraj", "Raj", "Pranay", "Sackin")
myiter = iter(mytuple)
print (next(myiter))
print (next(myiter))
print (next(myiter))
print (next(myiter))

```

Output:

Neeraj

Raj

Pranay

Sackin

```

mytuple = ("Neeraj", "Raj", "Pranay", "Sackin")
myiter = iter(mytuple)
print (next(myiter))
print (next(myiter))
print (next(myiter))
print (next(myiter))

```

Algorithm:

Step 1 : Form a tuple with certain elements inserted in it.

Step 2 : Use the for conditional statement to access each element of tuple

Step 3 : Print the elements of tuple

Program:

```

mytuple = ("Neeraj", "Raj", "Pranay", "Sackin")
for a in mytuple:
    print(a)

```

Output:

```

Neeraj
Raj
Pranay
Sackin

```

Aim : To use iter method with for loopAlgorithm:

Step 1 : Form a tuple with certain elements inserted in it.

Step 2 : Use the for conditional statement to access each element of tuple

Step 3 : Print the elements of tuple

PROGRAM :

```

class odd :
    def __iter__(self):
        self.num = 1
        return self

    def next(self):
        num = self.num
        self.num += 2
        return num

    myobj = odd()

    myiter = iter(myobj)

    x = int(input("Enter range number: "))

    for i in range(myiter):
        if x < i:
            print(i)

```

def __iter__(self):
def next(self):
return self

self.num = 1

self.num += 2

return num

myobj = odd()

myiter = iter(myobj)

x = int(input("Enter range number: "))

for i in range(myiter):

if x < i:

print(i)

Output:

Enter a number: 6

/

1

3

5

7

9

11

13

15

17

19

21

23

25

27

29

31

33

35

37

39

41

43

45

47

49

51

53

55

57

59

61

63

65

67

69

71

73

75

77

79

81

83

85

87

89

91

93

95

97

99

101

103

105

107

109

111

113

115

117

119

121

123

125

127

129

131

133

135

137

139

141

143

145

147

149

151

153

155

157

159

161

163

165

167

169

171

173

175

177

179

181

183

185

187

189

191

193

195

197

199

201

203

205

207

209

211

213

215

217

219

221

223

225

227

229

231

233

235

237

239

241

243

245

247

249

251

253

255

257

259

261

263

265

267

269

271

273

275

277

279

281

283

285

287

289

291

293

295

297

299

301

303

305

307

309

311

313

315

317

319

321

323

325

327

329

331

333

335

337

339

341

343

345

347

349

351

353

355

357

359

361

363

365

367

369

371

373

375

377

379

381

383

385

387

389

391

393

395

397

399

401

403

405

407

409

411

413

415

417

419

421

423

425

427

429

431

433

435

437

439

441

443

445

447

449

451

453

455

457

459

461

463

465

467

469

471

473

475

477

479

481

483

485

487

489

491

493

495

497

499

501

503

505

507

509

Ques : To print first 20 numbers using iter method.

Algorithm :

Step 1 : Define a class which will contain various functions.

Step 2 : Define iter method with an argument and return the value of argument.

Step 3 : Define next method which increments the value of argument by 1 & prints it.

Step 4 : Create a object which inherits the property of class and take the user input.

Step 5 : use for loop to print the value of the variable

Output:

```
class MyClass :
    def __init__(self):
        self.a = 1
    def __iter__(self):
        return self
    def __next__(self):
        if self.a <= 20:
            x = self.a
            self.a += 1
            return x
        else:
            raise StopIteration
```

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20

Program :

```
num = [0, 1, 5, 7, 9, 11, 13, 15, 20, 9]
num = list(map(lambda x: x % 5, num))
```

Algorithm :

```
print (num)
def even (x)
    if x % 2 == 0 :
        return "even"
    else:
        return "odd"
list (map (even, num))
```

Aim : To find if the number is odd or even from given list using map method.

Step 1 : Declare a list num variable and declare some elements

Step 2 : Define a function even which consist various conditional statement.

Step 3 : Further use if conditional statement to check the modulo number of each element of list and return the message accordingly.

Output :

```
[0, 1, 5, 7, 9, 11, 13, 15, 20, 9]
```

```
[even, even, odd, odd, odd, odd, odd, odd, even, odd]
```

Aim : To find square and cube of number simultaneously using map method.

Algorithm :

Step1 : Define a function which returns the square of given number.

Step2 : Define another function which returns cube of given number.

Step3 : Store the output simultaneously in a list

Step4 : Use the for loop followed by the map function and print the value of result

Output:

```
[0, 0]
[1, 1]
[4, 8]
[9, 27]
[16, 64]
```

Program

```
def square (xc):
    return (xc**2)

def cube (xc):
    return (xc**3)
```

```
func1 = [square, cube]

for i in range (4):
    value = list (map (lambda x : x(x), func1))
    print (list (value))
```

Program :

```
list1 = [1, 2, 3, 4, 5]
```

```
empty = []
```

```
for i in list1 :
```

```
    empty.append(i**2)
```

```
print(empty)
```

Algorithm :

Step 1 : Define a list which contains certain values.

Step 2 : Define an empty list .

Step 3 : Use for loop followed by append method to append the result into the empty list .

Step 4 : Print the value of list .

Dhruv

Revised
Program:

```
try:  
    fileobj = open("abc.txt", "w")  
    fileobj.write("Python is an indented language")  
except IOError:  
    print("It is an Environmental Error!")
```

Algorithm:

Step 1 : Open a previous file using any mode of file operation under try block.

Step 2 : Define the except block to print the message for the error.

Output:
Operation is successful

Step 3 : Define the else block to print the message according if no error is found

Program:

```
try :
    x = int(input("Enter a number:"))
except ValueError:
    print ("This is value Error !")
except IOError:
    print ("This is environmental error !")
else:
    print ("Operation Successful !")
```

Output :

```
Enter a number: 20
Operation successful!
```

```
Enter a number: 2+3
This is value Error !
```

Algorithm :

Step 1 : Accept the user input of integer datatype in the try block.

Step 2 : Define the except block with value error as a keyword and display appropriate message.

Step 3 : Define else block if none of the error is encountered and display message accordingly.

Dyman

```
import re
sequence = input("Enter name and roll no. of students without space")
pattern = r'\d+'
output = re.findall(pattern, sequence)
```

Output :

Print (output)

Algorithm :

Ques :- To segregate alphabetical values from numerical value in the given string.

Step 1 :- Import the regular expression library
 Step 2 :- Initialise a variable which takes the user input that contains name & roll no. of student without space.

[17]

Enter name and roll no. of students without space :

Neeraj 1741 Sachin 1744 Aditya 1769

[1741, '1744', '1769']

Step 3 :- Define the pattern as per the requirement of program

Step 4 :- Use the findall method and print the output as per the input of pattern

Program:

```

import re
sentence = input("Enter a sentence: ")
pattern = r'\b\w{2}\b|\w{2}\b'
output = re.findall(pattern, sentence)
print(output)
    
```

Algorithm :

Step1 : Import the regular expression library

Step2 : Take a sentence from the user and assign it to variable

Step3 : Define a pattern that finds the consecutive two characters at word boundaries

Step4 : Use the.findall method and print the output.

Program :

```

import re
x = int(input("Enter a sentence : "))
y = input("Enter a word to be found from the string:")
output = re.findall(y, x)
if(output):
    print ("Match found!")
else:
    print ("Match not found!")

```

x = int(input("Enter a sentence : "))
y = input("Enter a word to be found from the string:")

Algorithm:

Step 1 : Import the regular expression library.

Step 2 : Define a variable which take a sentence from the user.

Step 3 : Define another variable which takes the word to be find.

Output :

(i) Enter a sentence : My name is Neeraj

Enter a word to be found from the string: name

Match found!

DYKJW

(ii) Enter a sentence : My name is Neeraj
Enter a word to be found from the string : Sachin
Match not found!

Aim : To find a word starting from vowel

Algorithm:

Step 1 : Import the regular expression library

Step 2 : Take a few lines as input from the user.

Step 3 : Define a pattern which finds the vowels in the string

Step 4 : Use the.findall method to find the word starting with vowel

Step 5 : Print the output

Output:-

Enter few lines: My name is Neeraj. I am Indian

[is, 'am', 'Indian']

Program :

```

import re
string = input("Enter few names: ")
pattern = r '\s'
replace = ''
output = re.sub(pattern, st.replace, string)
print(output)
    
```

Aim : To remove whitespace in a given string
Algorithm:

Step 1 : Import regular expression library

Step 2 : Take user string input from user which contains whitespace.

Step 3 : Define the pattern as per the requirement.

Step 4 : Assign a variable with empty string

Step 5 : Use the sub method to substitute the whitespace with string and print the output.

Enter few names: Neeraj Raj Pranay
Neeraj Raj Pranay

Program :

042

```

import re
list1 = ['8767522106', '9821971119', '7029623410', '87825904781']
pattern = r'\b[8-9]+\d{9}\b'

for i in list1:
    if re.match(pattern, i) and len(i) == 10:
        print("Number satisfies the condition!")
    else:
        print("Number does not satisfy the condition.")

```

Algorithm:

- Step 1 : Import regular expression library
- Step 2 : Define a list variable which contains the input numbers

- Step 3 : Define the pattern required to extract numbers which satisfy the condition

- Step 4 : Use the for loop followed by the if conditional statement and print the message accordingly

Number satisfies the condition.
Number does not satisfy the condition.

Output :

Number satisfies the condition.
Number does not satisfy the condition.

Number does not satisfy the condition

```
import re
email = ['xyz@bsc-edu', 'abc@gmail.com', 'JK@hotmail.com']
pattern = re['@']
```

for i in email :

```
output = re.split(pattern,i)
```

print(output)

Output :

```
['xyz', 'bsc-edu']
```

```
['abc', 'gmail.com']
```

```
['JK', 'hotmail.com']
```

Output :

```
['xyz', 'bsc-edu']
```

```
['abc', 'gmail.com']
```

```
['JK', 'hotmail.com']
```

Algorithm :

Step 1 : Import the regular expression library

Step 2 : Assign a variable which contains the email address in list data types

Step 3 : Define a pattern required to split the two parts

Step 4 : Use the for loop followed by `re.split` method to segregate the hostnames & domain name & print the output.

Dr
Sik

PRACTICAL - 5

044

PROGRAM :

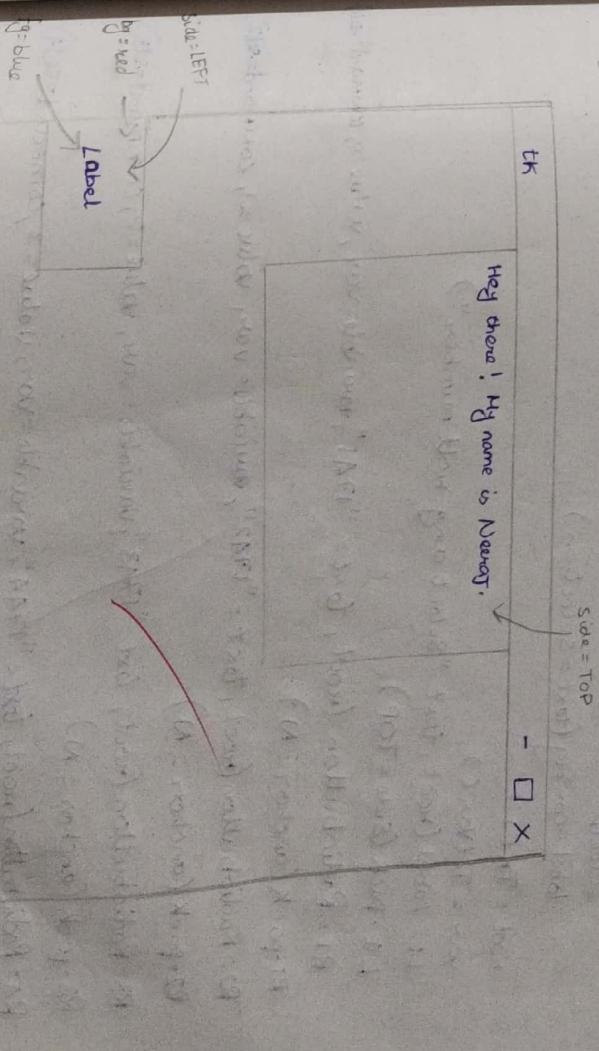
```

from tkinter import *
root = Tk()
root = Text(root)
T1.insert(END, "Hey There! My name is Neeraj..")
T1.pack(side=TOP, padx=20, pady=30, ipadx=40, ipady=50)
L1 = Label(root, text="Label", bg="red", fg="blue")
L1.pack(side=LEFT, pady=10, ipadx=20, ipady=30)
root.mainloop()

```

Step 1 : use the `tkinter` library for importing the features of `text` widget.

Step 2 : Create a variable from a text variable & position it onto the parent window.

Output :

Step 3 : use the `pack()` along with the object created from `text` method & use the parameters.

```
i) side = TOP, padx = 20, pady = 30
```

Step 4 : use the main loop method for triggering corresponding event.

Step 5 : Now repeat above step with a `label` method which takes the following argument.

i) Name of parent window

ii) The `text` attribute which defines the string.

iii) The background colour (`bg`)

iv) The foreground colour (`fg`)

Now use `pack()` with relevant attributes.

- (a) Label
- (b) Text
- (c) polynome

def sel1 () :

selection = "Wesley"

label = config(text=selection)

def sel2 () :

selection = "Raj"

label = config(text=selection)

def sel3 () :

selection = "Kenny"

label = config(text=selection)

def sel4 () :

selection = "Sackin"

label = config(text=selection)

root = Tk()

var = IntVar()

L1 = Label(root, text="Select any root number")

L1.pack(side=TOP)

R1 = Radiobutton(root, text="17A1", variable=var, value=0, command=sel1)

R1.pack(anchor=N)

R2 = Radiobutton(root, text="17A2", variable=var, value=1, command=sel2)

R2.pack(anchor=N)

R3 = Radiobutton(root, text="17A3", variable=var, value=2, command=sel3)

R3.pack(anchor=N)

R4 = Radiobutton(root, text="17A4", variable=var, value=3, command=sel4)

R4.pack(anchor=N)

label = Label(root)

label.pack(side=BOTTOM)

root.mainloop()

Algorithm:

Step 1 : Use the tkinter method to import the relevant method.

Step 2 : Define a function which tells user about given selection made from multiple option available.

Step 3 : Use the config method along with label method & call the variable as an argument within method.

Step 4 : Now define the parent window & define option using config variable.

Step 5 : Now create object of Radiobutton which will take following arguments

- Positioning on Parent Window
- Text Variable
- Define variable argument

iv) Corresponding value and trigger the given function

Step 6 : Now call the pack method for corresponding Radio object so created and specify argument as an anchor attribute.

Step 7 : Now define a label object and place it onto parent window using pack method & finally use mainloop method.

aim : To make & use of Radiobutton widget for selection of one of the multiple option .

OUTPUT:

046

CK

- □ X

Select any roll number:

1741

1742

1743

1744

Pranay

P

X E

✓
✓

```
from tkinter import *
root = Tk()
root.geometry('450x400')
l = Label(root, text="B Batch Roll Numbers", bg="black", fg="white")
l.pack()

scroll = Scrollbar(root)
```

```
scroll.pack(side=RIGHT, fill=Y)
```

```
mylist = Listbox(root, yscrollcommand=scroll.set, bg="light blue")
```

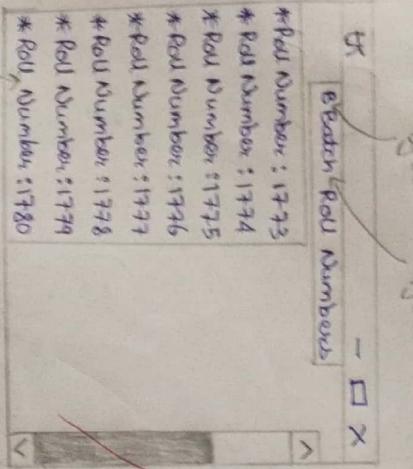
```
for num in range(41, 81):
```

```
    mylist.insert(END, "*Roll Number: 17", str(num))
```

```
scroll.config(command=mylist.yview)
```

```
root.mainloop()
```

Output :



Algorithm :

Step 1 : Import tkinter library to use Scroll Bar widget.

Step 2 : Create an object corresponding to scroll parent window & create an object from scrollbar & place it on the parent window & created.

Step 3 : Create an object of label method to provide a head and place it on parent window.

Step 4 : Use pack method along with object of scrollbar with use argument side + fill.

Step 5 : Create an object of listbox method and place it on parent window with attribute yscrollcommand

Step 6 : Use for loop to insert values in the object of list box by using insert method.

Step 7 : Use config method along with scroll bar object.

Step 8 : Finally call the mainloop method.

DY

Aim : To make use of Scroll Bar widget of the GUI application.

Program : To make use of message method of the GUI application

Algorithm :

Step 1 Import relevant method from tkinter library

Step 2 Define a function and use messagebox along with different methods available which contains one or more arguments

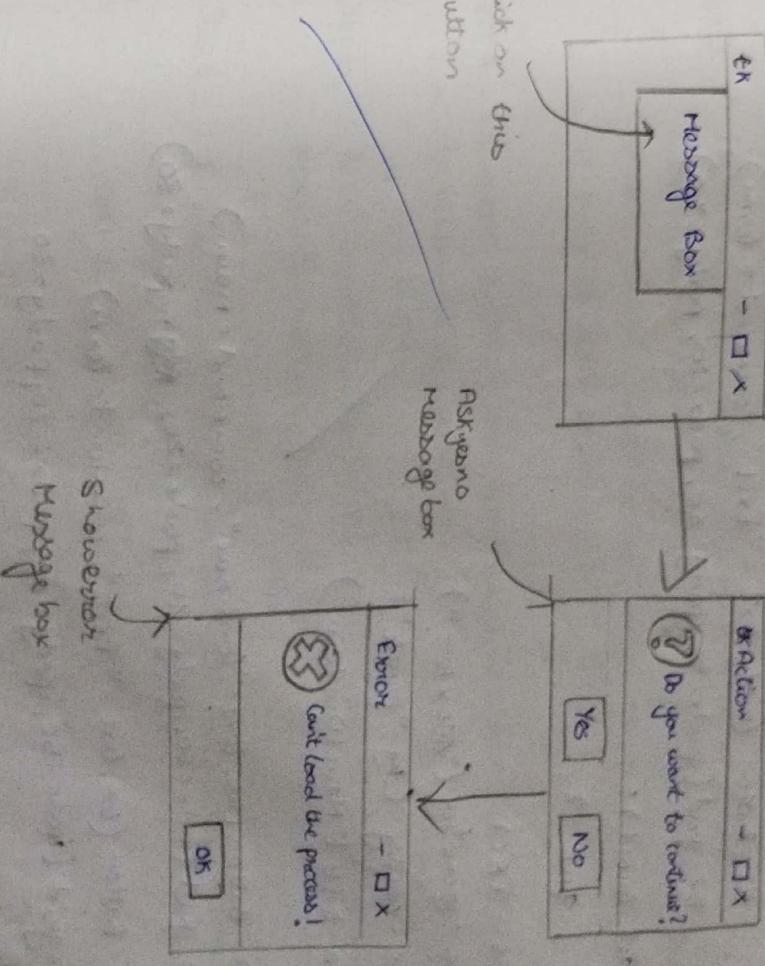
Step 3 Create an object from button method and place it onto the parent window with text and command attributes specified.

Step 4 Use pack method and finally use the mainloop method

Program :
 Python imports *
 mess window imports message
 mess window imports messagebox
 def myfunc():
 msg=messagebox("Action", "Do you want to continue?")
 msg=messagebox("Showcase", "Error", "Can't load the process!")

```
root = Tk()
root.config(bg="gray")
root.title("Message Box")
root.geometry("300x200")
root.resizable(0,0)
root.iconbitmap("tkinter icon.ico")
root.mainloop()
```

Output :



```

def
from tkinter import *
def main():
    root = Tk()
    root.geometry("450x300")
    root.config(bg="light green")
    root.title("Window 1")

B1 = Button(root, text="Next", command=main1)
B1.grid(ipadx=50, ipady=40, padx=20, pady=30)

B2 = Button(root, text="Exit", command=term)
B2.grid(ipadx=50, ipady=40, padx=20, pady=30)

def term():
    quit()

tos = Tk()
tos.geometry("450x300")
tos.config(bg="purple")
tos.title("Main Window")

```

Aim : Program to traverse various windows using the button widget.

Algorithm :

Step 1 : Import the relevant method from tkinter library

Step 2 : Define a function and create a object of given window by using the three methods namely config, title, minsize.

Step 3 : Define a button object which will be placed on the current window to traverse and define another button which will be used to exit from the window and place it onto current window.

Step A : Define another function which will use the quit method to terminate the program.

Step B : Now create an object of main window and use various attributes methods like config, title, geometry etc.

B3 = Button(tos, text="Continue", command=main)

B3.grid(ipadx=50, ipady=40, padx=20, ~~pady~~=pady=30)

B2 = Button(tos, text="Exit", command=term)

B2.grid(ipadx=50, ipady=40, padx=20, pady=30)

def main():
 bg = "red"
 fg = "white"

top = tk.Tk()
 top.title("Kontrol")

top.resizable(False, False)

top.geometry("300x200")

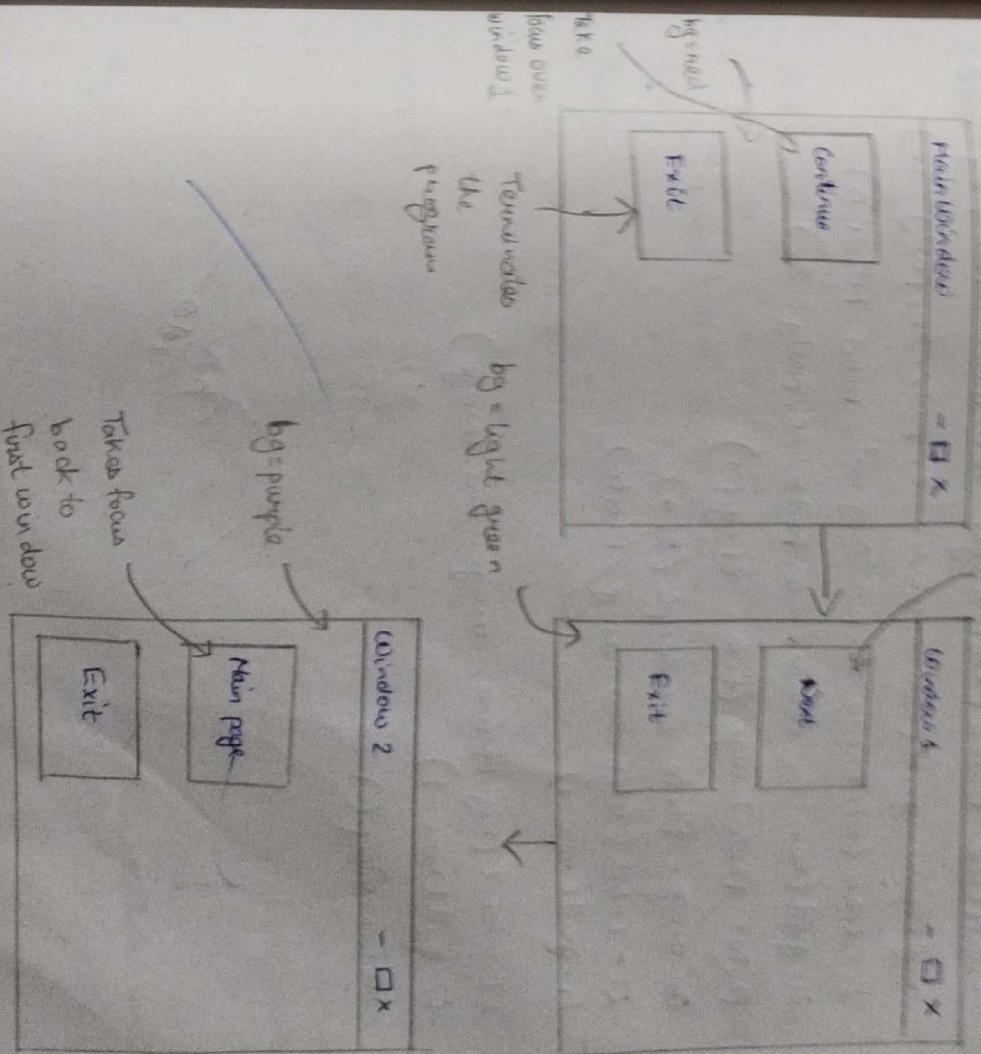
- Step 9 : Define another function with controls
variables bottom placed on third window & define
list feedback respectively and use the ~~end~~ global net
along with the text ~~bottom~~

- Step 8 : Finally call the ~~mainloop~~ method of module

Defining:

Main window

Window 2



```

from tkinter import *
root = Tk()
root.title("Image")
root.geometry("500 x 500")
root.config(bg="red")

leftframe = Frame(root, height=150, width=150, bg="gray")
leftframe.grid(row=0, column=0, padx=20, pady=30)

rightframe = Frame(root, height=150, width=150, bg="red")
rightframe.grid(row=0, column=2, padx=20, pady=30)

l1 = Label(leftframe, text="original", relief=Raised)
l1.grid(row=0, column=0, padx=20, pady=30)

photo = PhotoImage(file="t1.gif")
oriimage = photo.subsample(3, 4)

l1 = Label(leftframe, image=photo)
l1.grid(row=1)

l2 = Label(rightframe, image=oriimage)
l2.grid(row=0, column=1)

root.mainloop()
    
```

Dinesh

Aim : To make use of PhotoImage method of the GUI application

Algorithm :

Step 1 : Import the relevant method from tkinter library

Step 2 : Create an object corresponding to the parent window and use the three methods namely title, geometry and config

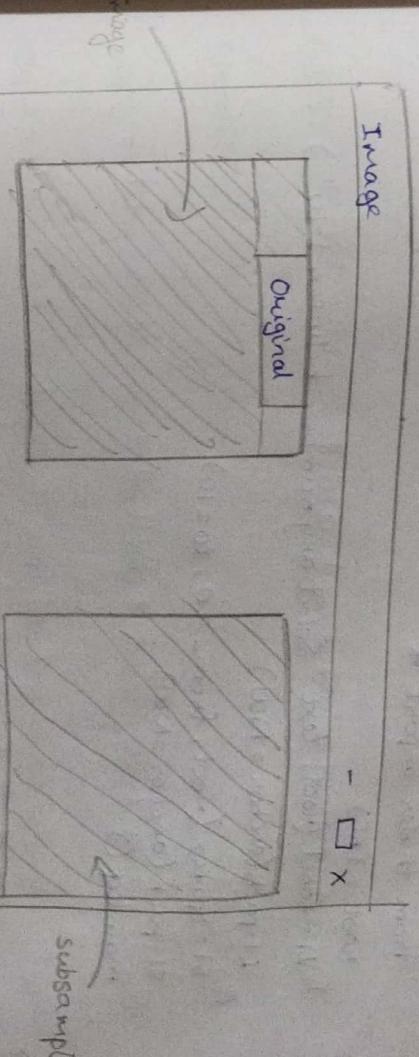
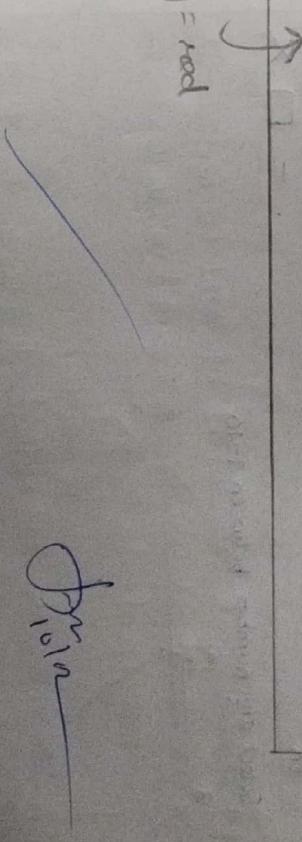
Step 3 : Create a left frame object from the frame method and place it onto the parent with height and width specified also specify the background colour.

Step 4 : Use the grid method along with the left frame object and create another frame object named rightframe and use grid method along with right frame object

Step 5 : Create a label object and place it onto the leftframe with text attribute and with relief attribute as RAISED. Subsequently use the grid method with row, column value specified as 0, 0 with some external padding values.

Step 6 : Use the PhotoImage method with file attribute specif

- Step 7 : use the subsample method with object of image and give x & y coordinate
- Step 8 : use label method & position on sub sampling using grid method on left frame. Similarly use the right frame object and place original image on frame
- Step 9 : Finally trigger the mainloop method

OUTPUT :ImageOriginalsubsamplebg = redPrin

From tkinter import *

root = Tk()

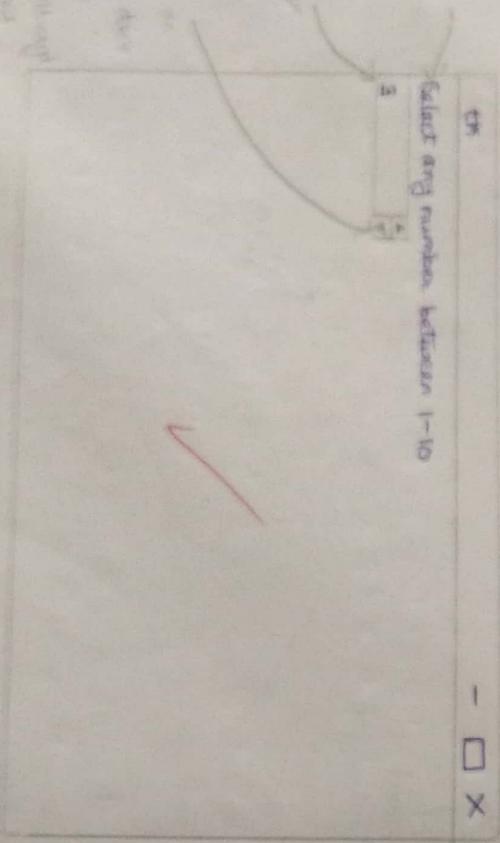
L1 = Label(root, text="Select any number between 1-10")

L1.pack(anchor = NW)

S1 = Spinbox(root, from_ = 0, to = 10)

S1.pack(anchor = NW)

mainloop()



Program 2: To make use of Spinbox widget from GUI application

Position :

Step 1 : Import the relevant method from tkinter library .

Step 2 : Create an object of the parent window .

Step 3 : Create an object of the Spinbox method and place it onto the parent window with attributes from and to with certain values specified .

Step 4 : Now use the pack method along with object of spinbox method and finally call the mainloop method .

~~1. Create a label which contains the string "Select any number between 1-10".~~

~~2. Create a spinbox which displays a number between 1-10.~~

~~3. Create a button which displays the word "Submit".~~

PROGRAM :

054

```
from tkinter import *
root = Tk()
P1 = PanedWindow()
P1.pack(fill=BOTH, expand=1)
P1.add(Label(P1, text="test1", bg="red"))
L1 = Label(P1, text="test2", bg="yellow")
P1.add(L1)
P2 = PanedWindow(P1, orient=VERTICAL)
P1.add(P2)
P2.add(Label(P2, text="test3", bg="green"))
P2.add(Label(P2, text="test4", bg="blue"))
P2.add(Label(P2, text="test5", bg="black"))
P2.add(Label(P2, text="test6", bg="white"))
P2.add(Label(P2, text="test7", bg="grey"))
P2.add(Label(P2, text="test8", bg="brown"))
P2.add(Label(P2, text="test9", bg="purple"))
P2.add(Label(P2, text="test10", bg="pink"))
P2.add(Label(P2, text="test11", bg="cyan"))
P2.add(Label(P2, text="test12", bg="magenta"))
P2.add(Label(P2, text="test13", bg="olive"))
P2.add(Label(P2, text="test14", bg="teal"))
P2.add(Label(P2, text="test15", bg="lime"))
P2.add(Label(P2, text="test16", bg="darkblue"))
P2.add(Label(P2, text="test17", bg="darkred"))
P2.add(Label(P2, text="test18", bg="darkgreen"))
P2.add(Label(P2, text="test19", bg="darkyellow"))
P2.add(Label(P2, text="test20", bg="darkcyan"))
P2.add(Label(P2, text="test21", bg="darkmagenta"))
P2.add(Label(P2, text="test22", bg="darkolive"))
P2.add(Label(P2, text="test23", bg="darkteal"))
P2.add(Label(P2, text="test24", bg="darklime"))
P2.add(Label(P2, text="test25", bg="darkdarkblue"))
P2.add(Label(P2, text="test26", bg="darkdarkred"))
P2.add(Label(P2, text="test27", bg="darkdarkgreen"))
P2.add(Label(P2, text="test28", bg="darkdarkyellow"))
P2.add(Label(P2, text="test29", bg="darkdarkcyan"))
P2.add(Label(P2, text="test30", bg="darkdarkmagenta"))
P2.add(Label(P2, text="test31", bg="darkdarkolive"))
P2.add(Label(P2, text="test32", bg="darkdarkteal"))
P2.add(Label(P2, text="test33", bg="darkdarklime"))
```

ALGORITHM:

Step 1 : Import the relevant method from tkinter library

Step 2 : Create an object of parent window and also an object of Paned window method

Step 3 : Now use the pack method with attributes fill and expand specified.

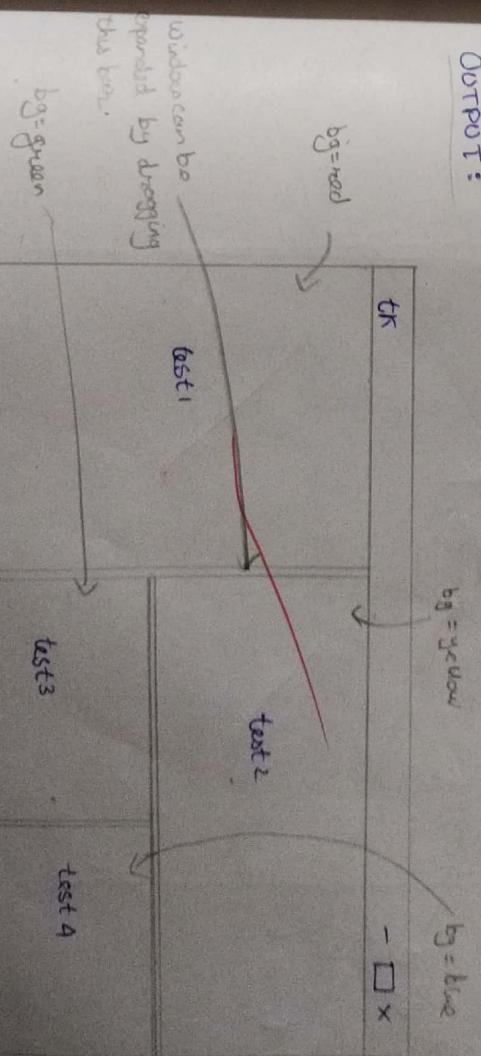
Step 4 : Now create an object of label method and place it on the Paned Window so created with text and background colour specified.

Step 5 : Now use the add method along with object of Paned window to add the label on it

Step 6 : Similarly create few more objects from Paned window and label and simultaneously arrange them accordingly

Step 7 : Finally trigger the mainloop method

OUTPUT:

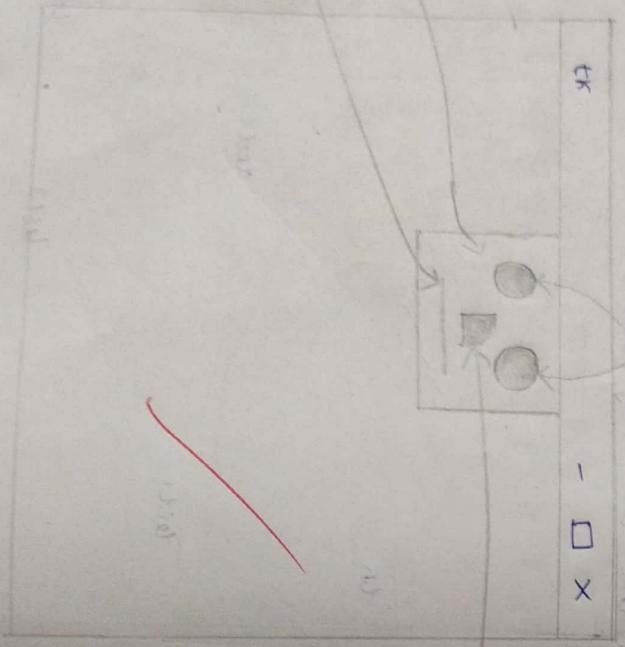


Program:

```

from tkinter import *
root = Tk()
c1 = Canvas(root, height=100, width=100, bg="red")
c1.create_oval(30, 60, 70, 100, fill="blue")
c1.create_oval(30, 60, 70, 100, fill="white")
c1.create_line(30, 90, 80, 90, fill="green")
c1.create_oval(10, 30, 40, 60, fill="green")
c1.pack()
mainloop()

```

Output:-

Aim : To make use of canvas widget in GUI application

ALGORITHM:

Step 1 : Import the relevant method from tkinter library.

Step 2 : Create an object of canvas method and place it onto parent window with height, width, background, color specified.

Step 3 : Create an object of the using the create_oval method with set the x1,y1 and x2,y2 co-ordinates specified with fill attribute.

Step 4 : Create an line using the process followed in previous step by using create_line method.

Step 5 : Create an oval using the process followed in previous step by using create_oval method.

Step 6 : Now use pack method along with object of canvas method and call the mainloop method.

*Dm
Dm*

Practical - 6

7.30

Aim : To make use of database library
Algorithm :

Step1: Import the dbm library to implement various features of database.

Step2: Use the open method for creating database by specifying name of database along with corresponding flag.

Step3: Use the object so created for accessing given website in corresponding name for the website.

Step4: Use the if conditional statement and print the statement accordingly.

import dbm
db = dbm.open("db1", "n")
db["www. bookology.com"] = "Book World"
if db["www. www. paterson.com"] == None:
 print ("Page Found")
else:
 print ("Page Not Found")

Program:

CASE1:

```
import dbm  
db = dbm.open("db1", "n")  
db["www. bookology.com"] = "Book World"  
if db["www. www. paterson.com"] == None:  
    print ("Page Found")  
else:  
    print ("Page Not Found")
```

CASE2:

```
import dbm  
db = dbm.open("db1", "n")  
db["www. paterson.com"] = "Fangs"  
if db["www. www. paterson.com"] == None:  
    print ("Page Found")  
else:  
    print ("Page Not Found")
```

056

Output:

Output:

Page Found

Page Not Found

PROGRAM:

```
input os, sqlite
connection = sqlite3.connect("student.db")
```

cursor = connection.cursor()

```
cursor.execute('create table data (Name VARCHAR, RNO INT, DOB DATE)')
```

```
cursor.execute('Insert into data values ("Neonay", 1741, "1941-12-09-2001")')
```

```
cursor.execute('Insert into data values ("Raj", 1742, "12-09-2001")')
```

```
cursor.execute('Insert into data values ("Ranay", 1743, "09-01-2001")')
```

```
connection.commit()
```

Step 1

```
cursor.execute('Select * from data')
```

```
cursor.execute('Select Name from data')
```

```
cursor.execute('Select D.O.B from data where RNO = 1741')
```

```
cursor.fetchall()
```

OUTPUT:

```
[('Neonay', 1741, '13-12-2001'), ('Raj', 1742, '12-09-2001'), ('Ranay', 1743, '09-01-2001')]
```

Step 2 : Now use connection object using `sqlite3` library and `connect` method for creating new database.

Step 3 : Now create cursor object using `cursor` method from `connection` object created in earlier steps

Step 4 : Now use `execute` method for creating table with column name of respective datatype.

Step 5 : Now with cursor object use `insert`, `delete` and `modify` statement

Step 6 : Use `commit` method to complete transaction using `connection` object

Step 7 : Use the `select` from cursor where clause to display the data as per the condition.

Step 8 : Use the `fetchall` method for displaying value and finally use the `close` method

ALGORITHM:
Aim : To make use of various statements of database.