

```
file obj = open("abc.txt", "w") # file open (write mode)
file obj.write("Computer Science subject \n")
file obj.write("DBMS in python in DS \n") # filenwrite
file obj.close() # file close
```

```
file obj = open("abc.txt", "r") # read mode.
```

```
# read()
```

```
str1 = file obj.read()
```

```
print("The output of read method : ", str1)
```

```
file obj.close()
```

```
>>> ("The output of read method : ", 'Computer Science subject  
in DBMS in python in DS \n')
```

```
# readline()
```

```
file obj = open("abc.txt", "r")
```

```
str2 = file obj.readline()
```

```
print("The output of readline method : ", str2)
```

```
file obj.close()
```

```
>>> ('The output of readline method : ', 'Computer  
Science subject \n')
```

```
# readlines()
```

```
file obj = open("abc.txt", "r")
```

```
str3 = file obj.readlines()
```

```
print("The output of readlines method : ", str3)
```

```
file obj.close()
```

```
>>> ('The output of readlines method : ', ['Computer  
Science subject \n', 'DBMS \n', 'Python \n', 'DS \n'])
```

PRACTICAL - 01

11/19

* Objective :- Demonstrate the use of different file opening mode, different attributes read method.

step 1 :- Create a file object using open method and in the write mode followed by writing some content onto the file and then closing the file.

step 2 :- Now open the file in read mode and then we read(), readline() and readlines() and store the output in variables and finally display the content of variables.

step 3 :- Now use the file object for finding the name of the file, the file mode in which it is opened whether the file is still open or close and finally the output of the softspace attribute.

510

step 04 :- Now open the fileobject in write mode
write some another content and close open
subsequently then again open the fileobject
in "w+" mode that is the update mode
and write "content".

step 05 :- Open fileobject in read mode, display the
update written contents and close open
again in "r+" mode with parameter
based and display the output subsequently.

step 06 :- Now Open fileobject in appended mode
open with method write content close the
fileobject again Open the fileobject in
read mode and display the appended
with in output.

file attributes :-

```
a = fileobj.name  
print("name of file (name attribute) : ", a) 018  
>>> ('name of file(name attribute)', 'abc.txt')  
b = fileobj.closed  
print("closed attribute : ", b)  
>>> ('closed attribute', 'True')  
c = fileobj.mode  
print("file mode", c)  
>>> ('file mode', 'w')  
d = fileobj.softspace  
print("softspace", d)  
>>> ("softspace : ", 0)
```

w+ mode

```
fileobj = open("abc.txt", "w+")  
fileobj.write("Saurabh")  
fileobj.close()
```

write mode

```
fileobj = open("abc.txt", "w")  
fileobj.write("DBMS")  
fileobj.close()
```

r+ mode

```
fileobj = open("abc.txt", "r+")  
str1 = fileobj.read(6)  
print("Output of r+", str1)  
fileobj.close()  
>>> ('output of r+', 'Saurabh')
```

read mode

```
fileobj = open("abc.txt", "r")  
str2 = fileobj.read()  
print("Output of read mode", str2)  
>>> ('Output of read mode', 'Saurabh')
```

```
# append mode  
fileobj = open("abc.txt", "a")  
fileobj.write("data structure")  
fileobj.close()  
fileobj = open("abc.txt", "r")  
str3 = fileobj.read()  
print("Output of appended mode : ", str3)  
fileobj.close()  
>>> ('Output of appended mode : ', 'daurabin', 'data structure')
```

tell()

```
fileobj = open("abc.txt", "r")  
pos = fileobj.tell()  
print("tell() : ", pos)  
fileobj.close()  
>>> ('tell() : ', 0)
```

seek()

```
fileobj = open("abc.txt", "r")  
fileobj.seek(0, 0)  
str8 = fileobj.read(10)  
print("The beginning of the file : ", str8)
```

finding length of different lines exist within lines

```
fileobj = open("abc.txt", "r")  
str9 = fileobj.readlines()  
print("Output : ", str9)  
for line in str9:
```

print(len(line))

fileobj.close()

```
>>> ('Output : ', [1, 'college databases'])
```

step 07 :- Open the file object in read mode, declare a variable and perform file object dot tell method and store the output consequently in variable.

step 08 :- Use the seek method with the arguments with opening the file object in read mode and closing subsequently.

step 09 :- Open file object with read mode also use the readlines method and store the output consequently in and print the same for counting the length use the for condition statement and display the length.

~~Not
Done~~

* Program:-

```
mytuple = ("Abhinav", "Amar", "Abhay", "Aman")
myiter = iter(mytuple)
print(next(myiter))
print(next(myiter))
print(next(myiter))
print(next(myiter))
```

* Output:-

```
Abhinav
Amar
Abhay
Amar
```

* Program:-

```
mytuple = ("Abhinav", "Amar", "Abhay", "Aman").
```

= for a in mytuple:

```
    print(a).
```

* Output:-

```
Abhinav.
Amar.
Abhay.
Aman.
```

- * AIM:- To display elements of a tuple using iterator method :-
Algorithm:-
S1:- Form a tuple with certain elements inserted in it.
S2:- Use the iter method with tuple and assign it to a variable.
S3:- Use the next method with variable and print the elements.
- * AIM:- To use iter method with for loop.
Algorithm:-
S1:- Form a tuple with certain elements inserted in it.
S2:- Use the for conditional statement to access each element of tuple.
S3:- Print the elements of tuple.

* Algorithm:-

1. define a class within that define an iter method which will initialize the first element within the container object.
2. Now use the next method and define the logic for displaying the odd values.

try on another bin. bottom box 26 36 38
• controls with

goal of this basket will be of this

-Algorithm

• it is below. which makes it. I put a msg
• now in breadth. condition of the all
• after to track
• else to where it exist - ea

Program :-

022

Using odd:

```
def __iter__(self):  
    self.num = 1  
    return self  
  
def __next__(self):  
    num = self.num  
    self.num += 2  
    return num  
  
def __next__(self):  
    num = self.num  
    self.num += 2  
    return num
```

myobj = odd()

myiter = iter(myobj)

num = int(input("enter the no"))

for i in myiter:

if i < num:
 print(i)

* Output:-

Enter the no 5

1
3

Dr Amrit

(A) Program:-

```
try:  
    f = open("Abhinav.txt", 'w')  
    f.write("My Name is Abhinav")  
except IOError:  
    print("Environment error")  
else:  
    print("Successful")
```

Output:-

Environment Error.

(B) Program:-

```
try:  
    a = int(input("Enter"))  
except ValueError:  
    print("Arithmetic Error")  
else:  
    print("Successful")
```

Output:-

Enter : 14

Arithmetic error.

PRACTICAL :- 3

Exception handling demonstration is given below:-

- (A) Algorithm:- i.e. writes over existing file after reading it.
1. Use the try block to define the normal block of action.
2. Example :- Define a file object and open the file in the write mode, and writes some content onto the file.
3. Use an except block with I/O errors, as an environment error and convey the appropriate message to the user, else display the message that the operation or operation is carried out successfully.

~~2nd yr~~ : ESD

Ex Program to demonstrate the multiple exception V:2
Error and value error:-

S1:- Use the try block and define the file object
and open the file in write or read mode and
write some content onto the file.

S2:- Also , accept the value from the user and if
it is a valid value , display the entered value
and terminate the condition by using the
break statement .

S3:- Define the except blocks for IOError and
ValueError

>>> Enter a number:- abc
 The value is invalid
 Enter a number:- pqr
 The value is invalid
 Enter a number : 73
 73

Program:-

```
fileobj = open ("abc.txt", "w")  

fileobj.write ("Python is an indented language in Python  

interpreted language")  

a = int (input ("Enter a number"))  

print (a)  

break;  

except IOError:  

print ("There is an environment error")  

except ValueError:  

print ("The value is invalid")
```

```
# match()
```

```
import  
pattern = r"\d+[^s]"
```

sequence = '444s represents computer science stream'

```
if re.match(pattern, sequence):
```

```
    print("matched pattern found!")
```

```
else:
```

```
    print("Not founded")
```

```
>>> matched pattern found!
```

```
# numerical values (regression)
```

```
import re
```

```
pattern = r'\d+'
```

```
string = 'hello123, howdy 789, 45 hours'
```

```
output = re.findall(pattern, string)
```

```
print(output)
```

```
>>> ['123', '789', '45']
```

```
# split()
```

```
import re
```

```
pattern = r'\d+'
```

```
string = 'hello123, howdy 789, 45 hours'
```

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

```
print(output)
```

```
>>> ['hello', 'howdy', 'hours']
```

PRACTICAL - 4

TOPIC:- Regular expression. Implementations of regular expression.

S1:- Import re module declare pattern and declare sequence we match method with declare arguments if arguments matched then print the same otherwise print pattern NOT FOUND!

S2:- Import re module declare pattern with literal and meta character declare string value. Use the findall() with arguments and print the same.

S3:- Import re module declare pattern with meta character use the split() and print the output.

750

QUESTION

S4:- import re module declare string and according
declare pattern replace blank space . use sligh no -
Use sub() with 3 arguments and print the
string without spaces

S5:- Import re module declare a sequence use search
method for finding subsequently use the : group()
with dot operator as search()

S6:- Import re module declare list with two
use the conditional statement . use if condition
for checking first number is either 8 or a and
next number are in range of 0 to 9 and
check whether the entered numbers are equal
to 10 if criteria matches print one number
matches otherwise print failed

```

# no-space:
import re
string = "abc def ghi"
pattern = r'\S+'
replace = ''
v1 = re.sub(pattern, replace, string)
print(v1)
>>> abcdefghi

# group():
import re
sequence = 'python is an interesting language'
v = re.search('IA python', sequence)
print(v)
v1 = v.group()
print(v1)
>>> <_sre.SRE_Match object at 0x02810F00>
python

# verify the given set of phone.numbers in fo:
import re
list1 = ['800456789', '9145673210', '7865432981',
         '6543210987654320']
for value in list1:
    if re.match(r'[\d]{3} [\d]{3} [\d]{4}', value) and len(value) == 10:
        print("criteria matched for all no!")
    else:
        print("criteria failed!")

```

```
>>> criteria matched for all no  
criteria matched for all no  
criteria failed!  
criteria matched for all number
```

usual

```
import re  
seq = 'plant is life overall'  
output = re.findall(r'[aeiouAEIOU]{wi}',  
print(output)  
>>> ['is', 'overall']
```

host and domain

```
import re  
seq = 'abc.tsc@edu.com , xyz@gmail.com'  
pattern = r'[ ]{wi}+[ ]{wi}'  
output = re.findall(pattern, seq)  
print(output)  
>>> ['abc', 'tsc', '@edu.com', 'xyz', '@gmail.com']
```

counting of first 2 letters

~~```
import re
s = 'mr. a , ms. b , ms. c , mr. t'
p = r'[{ms}/{mr}]+'
o = re.findall(p, s)
print(o)
m = 0
+ = 0
for v in o:
 if v[0] == 'm': m += 1
 else: + += 1
```~~

Step 7 :- import re module declare a string we  
the module declare a string we the module  
with.findall() for finding the vowels in the  
string and declare the same.

8:- Import re module declare host and domain  
name declare pattern for separating the host & domain  
name . use the.findall() and print the output  
respectively.

9:- Import re module enter a string we pattern to  
display only two elements of the particular string  
we find.all(), declares two variable with initial  
values as zero we for condition .

Jx-71

```
if (v == 'ms')
 f = f + 1
```

else

```
m = m + 1
```

```
print ("No of males is : ", m)
```

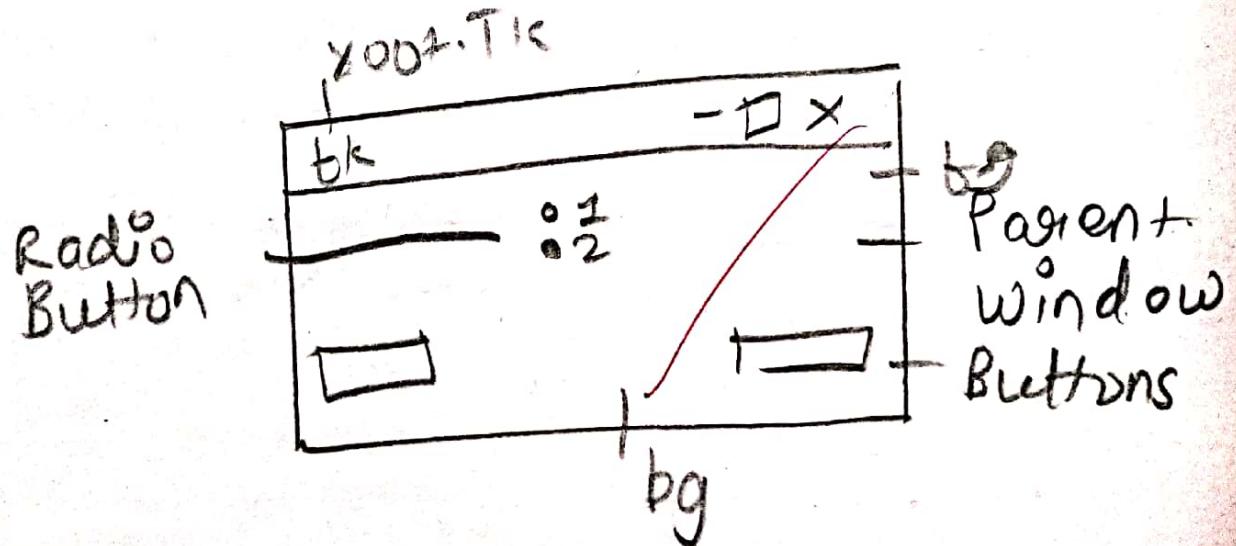
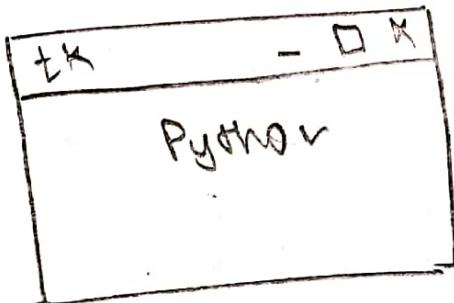
```
print ("No of females is : ", f)
```

```
>>> ['mx', 'ms', 'ms', 'mz']
('No of males is : 2')
('No of females is : 2')
```

### \* Program:-

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

### \* Output:-



## PRACTICAL-05

- \* Aim:- To make use of GUI application along with the basic pack method.
- \* Algorithm:-
  - 1:- Use the tkinter library for importing the features of entry widget.
  - 2:- Create a variable from a text variable & position it onto the parent window.
  - 3:- Use the pack() along with the object created from text method & use its parameters.
    - 1) side = TOP , padx = 20 , ipadx = 100 , ipady = 50
  - 4:- Use the mainloop method for triggering corresponding event.
  - 5:- Now repeat above step with a label method which takes the following arguments:-
    - 1) Name of parent window
    - 2) The background color
    - 3) The foreground color
    - 4) Now all pack() with relevant attributes

PSO.

- \* Aim:- To make use of radio button widget for selection of one out of the multiple options.

Algorithm:-

- S1:- Use the tkinter method to import the relevant method.
- S2:- define a function which will tell user about given selection mode from radio button.
- S3:- Use the config method along with label method.
- S4:- Now define the parent window of define using control var.

Now create object of radio button which will take following arguments for creation of radio button.

The syntax is as follows:

## Program:-

```
from tkinter import *
def sel1():
 selection = "Neeraj"
 label.config(text=selection)
def sel2():
 selection = "Raj"
 label.config(text=selection)
root = Tk()
var = IntVar()
L1 = Label(root, text="select any avoid number")
L1.pack(side=TOP)
R1 = Radiobutton(root, text="1721", variable=var, value=0, command=sel1)
R2 = Radiobutton(root, anchor=N, text="1722", variable=var, value=1, command=sel2)
```

031

```
R3 = Radiobutton(root, text="1723", variable=var, value=2, command=sel3)
R4 = Radiobutton(root, text="1733", variable=var, value=3, command=sel4)
R5 = Radiobutton(root, text="1734", variable=var, value=4, command=sel5)
```

Label = Label(root)

Label.pack(side=bottom)

PRACTICAL :- 3 (FRAMES)

- \* Aim:- (Frame) One Component (Buttons).
- s1:- Import the necessary method from tkinter library.
- s2:- Import tk message box.
- s3:- Define a parent window object along with its parent window.
- s4:- define a fun which will use tk message box with showinfo method along with info window attribute.
- s5:- Define a button with parent window object along with a command attribute.
- s6:- place the button widget onto the parent window and finally call mainloop() for triggering of the events called above.

# message box

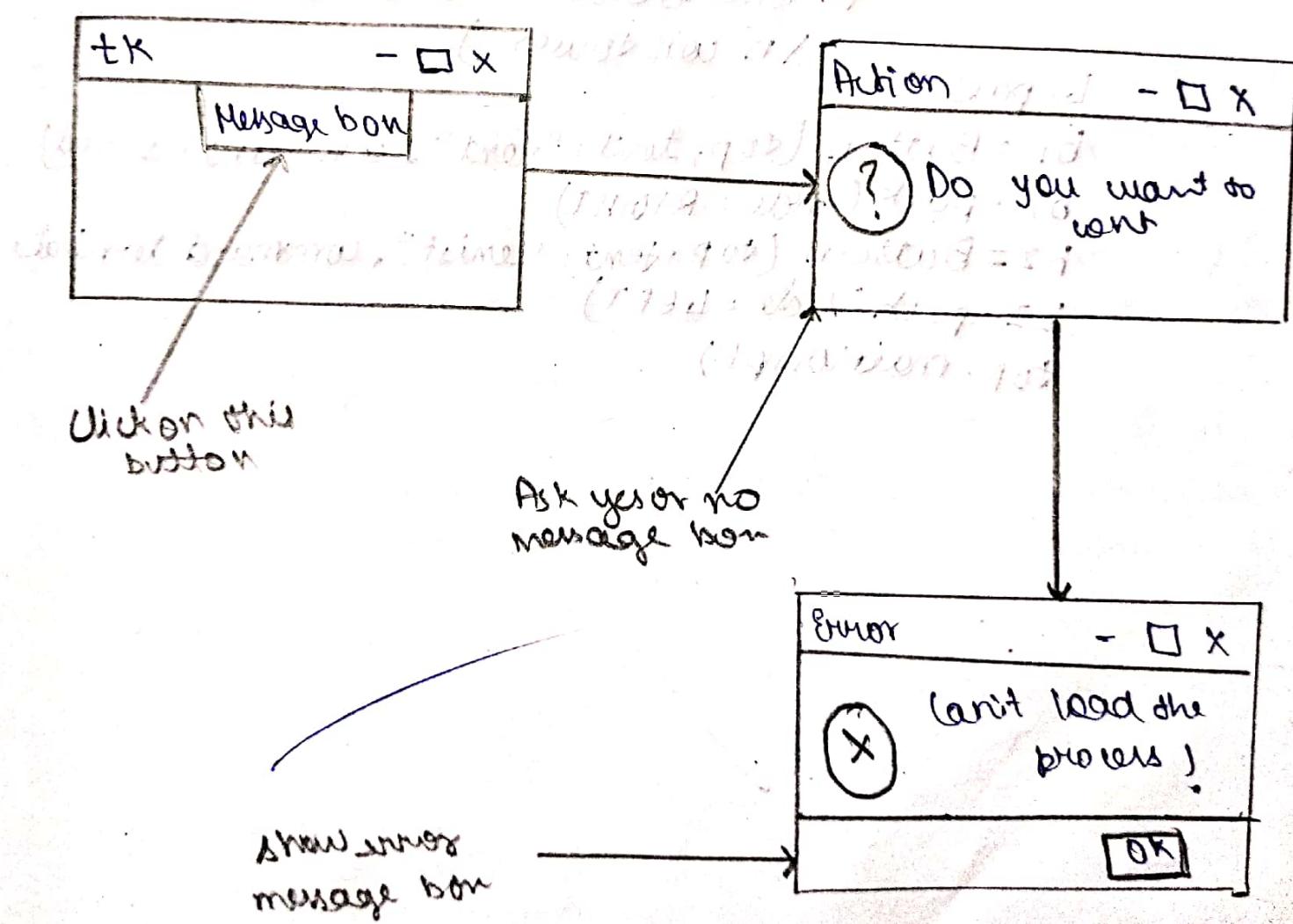
```
from Tkinter import *
import tkMessageBox
root = TK()

def function():
 tkMessageBox.showinfo("info window", "python")

 b1 = Button(root, text = "python", command = function)
 b1.pack()

root.mainloop()
```

- Output :- 90.00000000000001



# Multiple window

# different button (Relief(1))

from 'tkinter import'

root = TK()

root.minsize(300,300)

def main():

top = TK()

top.config(bg="black")

top.title("HOME")

top.minsize(300,300)

L = label (top, text = "SAN FRANCISCO In Plaza")

interest : In golden gate bridge

In lombardstreet In chinatown

In coil tower")

b1.pack()

b1 = Button (top, text = "next", command = second)

b1.pack(side = RIGHT)

b2 = Button (top, text = "exit", command = terminal)

b2.pack(side = LEFT)

top.mainloop()

- \* Aim :- To learn Grid with button in GUI
- s1 :- Import the relevant method from the tkinter library along with parent window object declared
- s2 :- Use parentwindow object along with minsize fun for window size.
- s3 :- Define a fun, main, declare parent window object and use config(), title(), minsize(), label() as well as button() and use pack() & mainloop() simultaneously.
- s4 :- Similarly define the function second which will use the quit method to terminate the program.
- s5 :- Now create an object of main window and use various methods like config, title, geometry etc.
- s6 :- Define two buttons which will be placed on the main window ; one to draw another window and the other to terminate the programs.

880.

S7:- Define another fun on which will carry value  
button placed on this window. Define two  
buttons respect and use the grid method  
along the two buttons.

Finally call the mainloop

Now click on the window and see the  
two buttons are displayed on the screen.

Now click on either one of the buttons which  
will change all actions to the help out. Now  
we can continue it to right or down right  
or pressing a left mouse button which

is now break of the driver created with  $\theta = 0$   
and the now open window is just opening when  
clicking onto it's input is with out bid

4. leding

from tkinter import \*

def main():

root = TK()

root.geometry("450x450")

root.config(bg="lightgreen")

root.title("Window")

B1 = Button(root, text="next", command=main)

B1.grid(ipadx=50, ipady=40, padx=20, pady=30)

B2 = Button(root, text="exist", command=term)

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

def term():

quit()

tos = TK()

tos.geometry("450 x 500")

tos.config(bg="purple")

tos.title("main window")

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

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

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

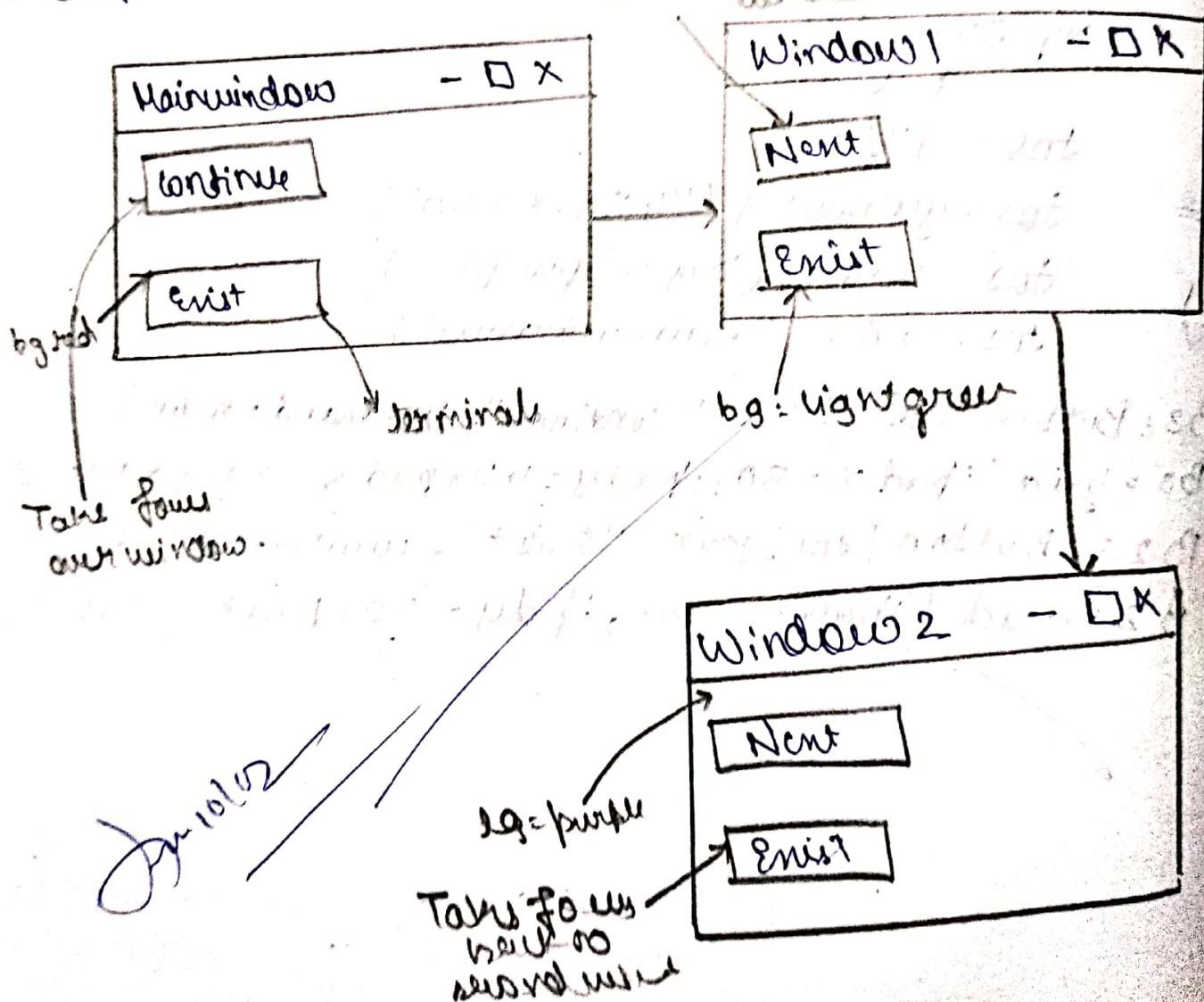
B2.grid(ipadx=50, ipady=40, padx=20, pady=30)

```

def main():
 top = Tk()
 top.geometry("480 * 500")
 top.config(bg="purple")
 top.title("Windows 1")
 B1 = Button(top, font="bold", command=main)
 B1.grid(ipadx=50, ipady=40, padx=20, pady=20)
 B2 = Button(top, font="bold", command=main)
 B2.grid(ipadx=50, ipady=40, padx=20, pady=20)
 mainloop()

```

\* Output:-

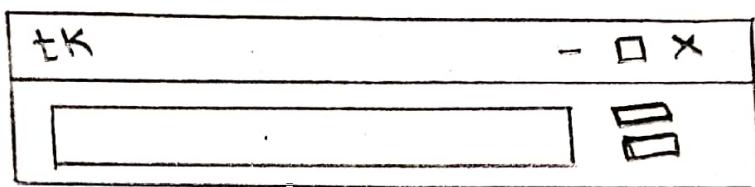


380  
589)

\* Source code:-

```
from tkinter import *
root = Tk()
s1 = Spinbox(root, from_=0,to=10)
s1.pack(anchor=S) ← from_=0
root.mainloop()
```

\* Output:-



\* Aim:- To learn use of spin box method in GUI.

Step 1:-

- Create an obj from the tk method and subsequently create an obj from the spinbox method

Step 2:-

- Make the obj created onto the parent window and trigger the corresponding event

Step 3:-

- Use the anchor pack method to provide the direction using anchor method

Step 4:-

- Use the mainloop method to terminate the wait for user input

80

Aim:- To learn the use of Paned windows

s1:- Create an obj from paned windows and use the pack method with the attribute fill and expand.

s2:- Create an obj from the label method and put it onto the paned window with the fill attribute and use the add method to embed the new obj.

s3:- Similarly create a second paned window obj and add it onto the 1<sup>st</sup> paned window with orientation specified.

s4:- Now create another label obj and place it onto the 2<sup>nd</sup> paned window obj and add the obj onto the 2<sup>nd</sup> pane.

s5:- Now use the mainloop method to start window

#### \* Program:-

```
from tkinter import*
root = Tk()
p = PanedWindow(bg="red")
p.pack(fill=BOTH, expand=1)
l1 = Label(p, text="PYTHON GUI", bg="green")
p.add(l1)
p1 = PanedWindow(p, orient=VERTICAL, bg="blue")
p.add(p1)
l2 = Label(p1, text="ABHINAV", bg="yellow")
p1.add(l2)
root.mainloop()
```

038

| TK         |         |
|------------|---------|
| PYTHON GUI | ABHINAV |

880

\* Program:-  
from tkinter import \*

root = Tk()

root.title("abhinav")

c = Canvas(root, height=600, width=600, bg="yellow")

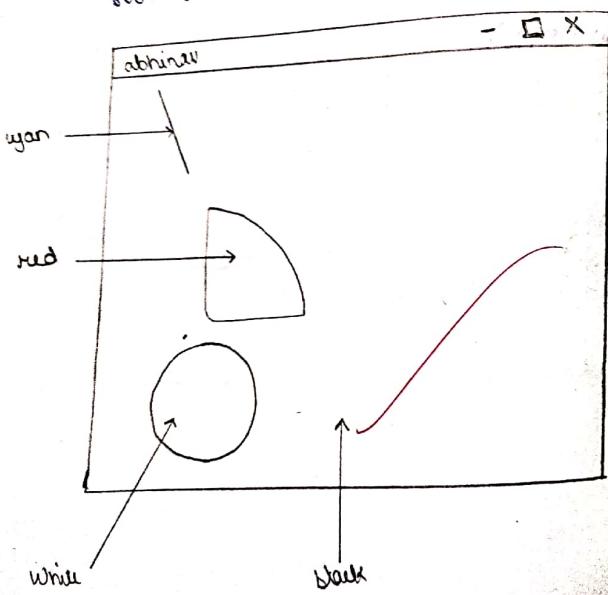
arc = c.create\_arc(10, 20, 30, 40, start=0, extent=30)

oval = c.create\_oval(50, 60, 70, 80, fill="red", fill="red")

line = c.create\_line(10, 20, 30, 40, fill="blue", width=2)

c.pack()

root.mainloop()



039

\* Aim:- To learn the use of canvas widget in Python GUI.

Step 1:- Use the tkinter method and create an object from the canvas method and use the attribute height, weight, bg color and the parent window obj. also assign various values to the attributes.

Step 2:-

Use the method create oval, create line and create arc along with the canvas obj. so created and use the coordinate value also use the fill attribute to assign various colors.

Step 3:-

Now call the pack method and mainloop method.



### PRACTICAL - 06

database library!

- \* Aim:- To learn use of database library and use the open method
- \* Step 1:- Import db library and use the data base by specifying name after creating the data base along with the correct port of the data base along with the correct port flag.

Step 2:- Use the object for connecting to given URL

WAP sign and the corresponding regular

for the web sign

- \* Step 3:- Check whether the given URL is address with the regular or the regular is equal to None. Then display. If the regular is NOT new age from URL address else NOT formed.

Code :-  
import dbm  
db = dbm.open ("data base", "flag", "")  
if db ["www"] == None:  
 print ("bad")  
else:  
 print ("good")

Output:-

bad

040

\* Program:-

```
import os.sqlite3
connection = sqlite3.connect("website.db")
c1 = connection.cursor()
c1.execute('create table website ("domain", cost, profit')
<sqlite3.Cursor object at 0x02D2F020>
c1.execute('insert into website ("Abhinav",2000,600)
<sqlite3.Cursor object at 0x02D2F020>
c1.execute('insert into website ("Rushal",5000,6000)
<sqlite3.Cursor object at 0x02D2F020>
c1.execute('drop table websites')
connection.commit()
```

c1.execute('select \* from student')

c1.fetchall()

c1.execute("Drop table websites")

\* Output:-

```
[{"Abhinav":2000,600},
 {"Rushal":5000,6000}]
```

041

- \* AIM:- To learn the use of cursor object.
- \* S1:- Import the corresponding library taking of database connection.
- S2:- Now create connection objects using sqlite library and connecting method for create the database.
- S3:- Now create the cursor object using cursor method from the connection object create in steps.
- S4:- Now with the cursor object we insert the entering the values in-ordering into different fields considering the data types.
- S5:- In the commit method to complete the transaction use the connection object.

ANSWER

180

### PROJECT:- 1

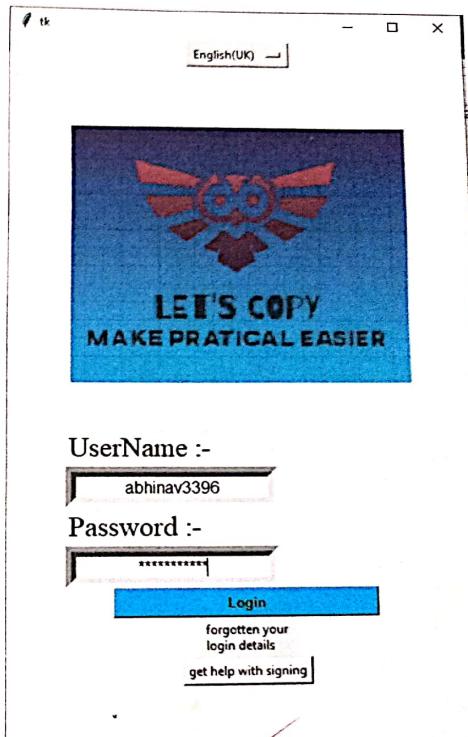
- \* Aim:- To build the GUI by merging almost all the widget in python.

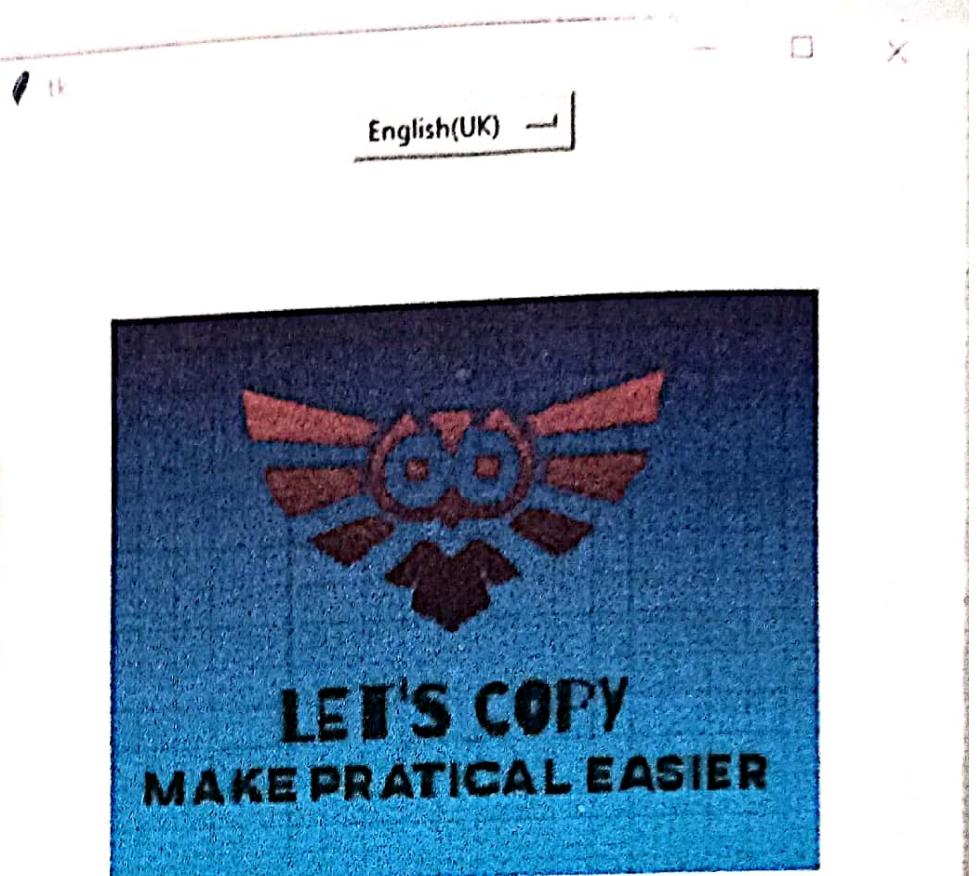
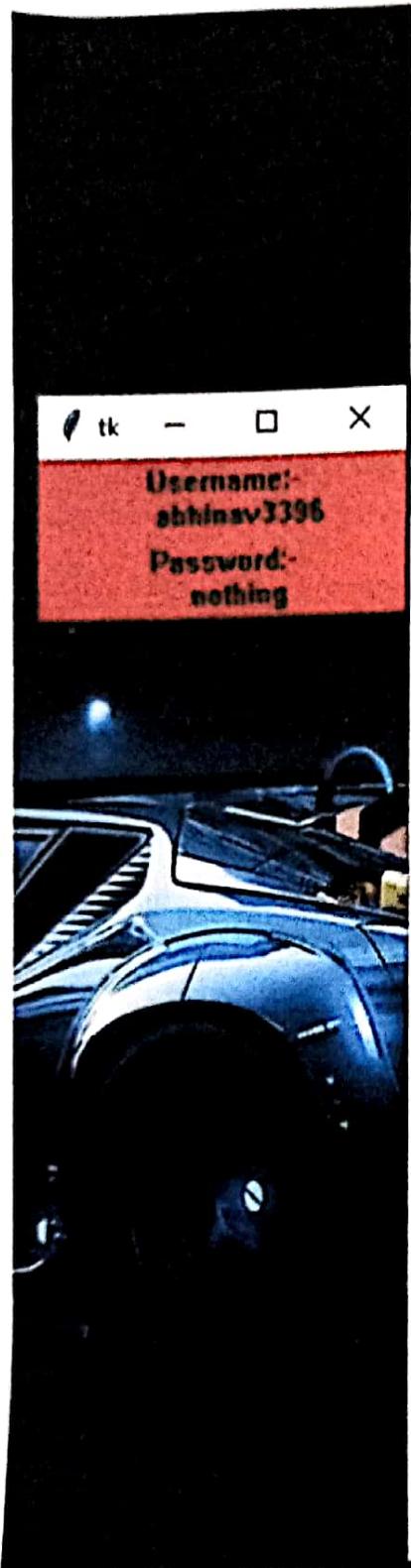
#### Algebra

#### SUMMARY

- Option button is used at top for selection of language.
- Image is inserted using photoimage widget.
- Label is used to print the data for user.
- Entry widget is also used to take the input from the user.
- Button widget is used for further continuation.
- And many more widget are used this will give the code of the selected subject and practical.

042





UserName :-

Password :-

**Login**

forgotten your  
login details

[get help with signing](#)

code final

```
from tkinter import *
w=Tk()
def login():
 h=Tk()
 def c_prog():
 print("ee")
 def pyt():
 print("hello")
 def ds():
 c=Tk()
 c.configure(background="orange")

 var=IntVar()
 def aa11():
 A=Tk()
 l=Label(A,text="""hi
my jkhdihsij
rwelk
rgj""")
 l.pack()
 A.mainloop()
 def aa22():
 A=Tk()
 l=Label(A,text=""""
""")
 l.pack()
 A.mainloop()
 def aa33():
 print("hi")
 def aa44():
 print("hi")
 def aa55():
 print("hi")
 def aa66():
 print("hi")
 def aa77():
 print("hi")
 def aa88():
 print("hi")
 def aa99():
 print("hi")
 def aa1010():
 print("hi")
 def aa1011():
 print(""""

")
 def sort(arr,l,m,r):
 n1=m-l+1
 n2=r-m
 L=[0]^(n1)
 R=[0]^(n2)
 for i in range(0,n1):
 L[i]=arr[l+i]
 for j in range(0,n2):
 R[j]=arr[m+1+j]
 i=0
 j=0
 k=l
 while i<n1 and j<n2:
 if L[i]<=R[j]:
 arr[k]=L[i]
 i+=1
 else:
 arr[k]=R[j]
 j+=1
 k+=1
 while i<n1:
 arr[k]=L[i]
 i+=1
```

code final

```

 k+=1
 while j<n2:
 arr[k]=R[j]
 j+=1
 k+=1
def mergesort(arr,l,r):
 if l<r:
 m=int((l+(r-1))/2)
 mergesort(arr,l,m)
 mergesort(arr,m+1,r)
 sort(arr,l,m,r)
arr=[12,23,34,56,78,45,86,98,42]
print(arr)
n=len(arr)
mergesort(arr,0,n-1)
print(arr)"""
def aa1111():
 print("hi")
r=Label(c,text="Select the practical no",font="times 20 bold")
r.pack(pady=20)
r1=Radiobutton(c,text="Search using linear unsorted list",font="times
10",variable=var,value=1,command=aa11)
r1.pack(anchor=NW,padx=55,pady=10)
r2=Radiobutton(c,text="Search using linear sorted list",font="times
10",variable=var,value=2,command=aa22)
r2.pack(anchor=NW,padx=55,pady=10)
r3=Radiobutton(c,text="Search using binary search",font="times
10",variable=var,value=3,command=aa33)
r3.pack(anchor=NW,padx=55,pady=10)
r4=Radiobutton(c,text="Bubble sort",font="times
10",variable=var,value=4,command=aa44)
r4.pack(anchor=NW,padx=55,pady=10)
r5=Radiobutton(c,text="Use of stack",font="times
10",variable=var,value=5,command=aa55)
r5.pack(anchor=NW,padx=55,pady=10)
r6=Radiobutton(c,text="use of queue add & delete",font="times
10",variable=var,value=6,command=aa66)
r6.pack(anchor=NW,padx=55,pady=10)
r7=Radiobutton(c,text="Circular queue ",font="times
10",variable=var,value=7,command=aa77)
r7.pack(anchor=NW,padx=55,pady=10)
r8=Radiobutton(c,text="post fix expression",font="times
10",variable=var,value=8,command=aa88)
r8.pack(anchor=NW,padx=55,pady=10)
r9=Radiobutton(c,text="Sort random variable using selection
sort",font="times 10",variable=var,value=9,command=aa99)
r9.pack(anchor=NW,padx=55,pady=10)
r19=Radiobutton(c,text="binary tree and traversal",font="times
10",variable=var,value=10,command=aa1010)
r19.pack(anchor=NW,padx=55,pady=10)
r29=Radiobutton(c,text="Merge sort",font="times
10",variable=var,value=12,command=aa1011)
r29.pack(anchor=NW,padx=55,pady=10)
r39=Radiobutton(c,text="sort data in quick sort",font="times
10",variable=var,value=11,command=aa1111)
r39.pack(anchor=NW,padx=55,pady=10)

```

```

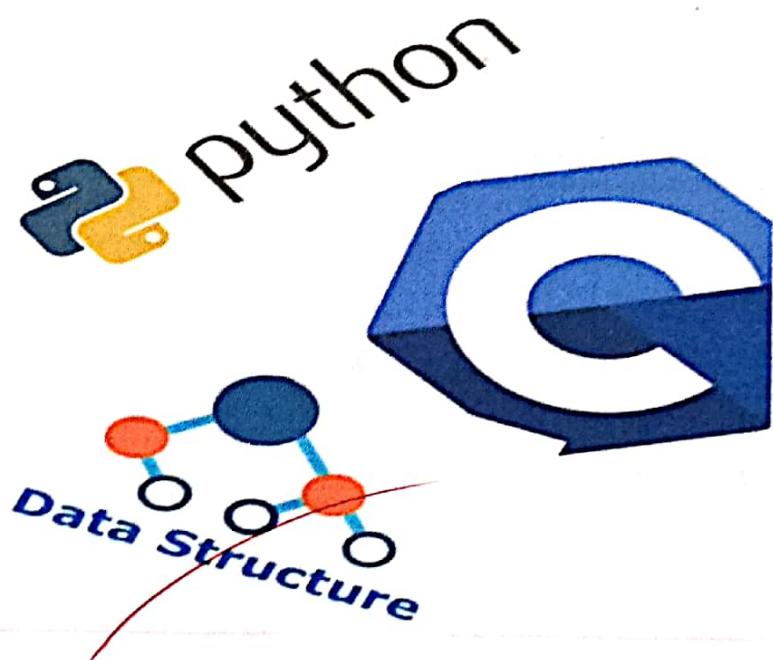
h.geometry('450x600')
h.configure(background="white")
l=Label(h,text="Appropriate selection ",bg="light blue",font="TIMES 30
Page 2

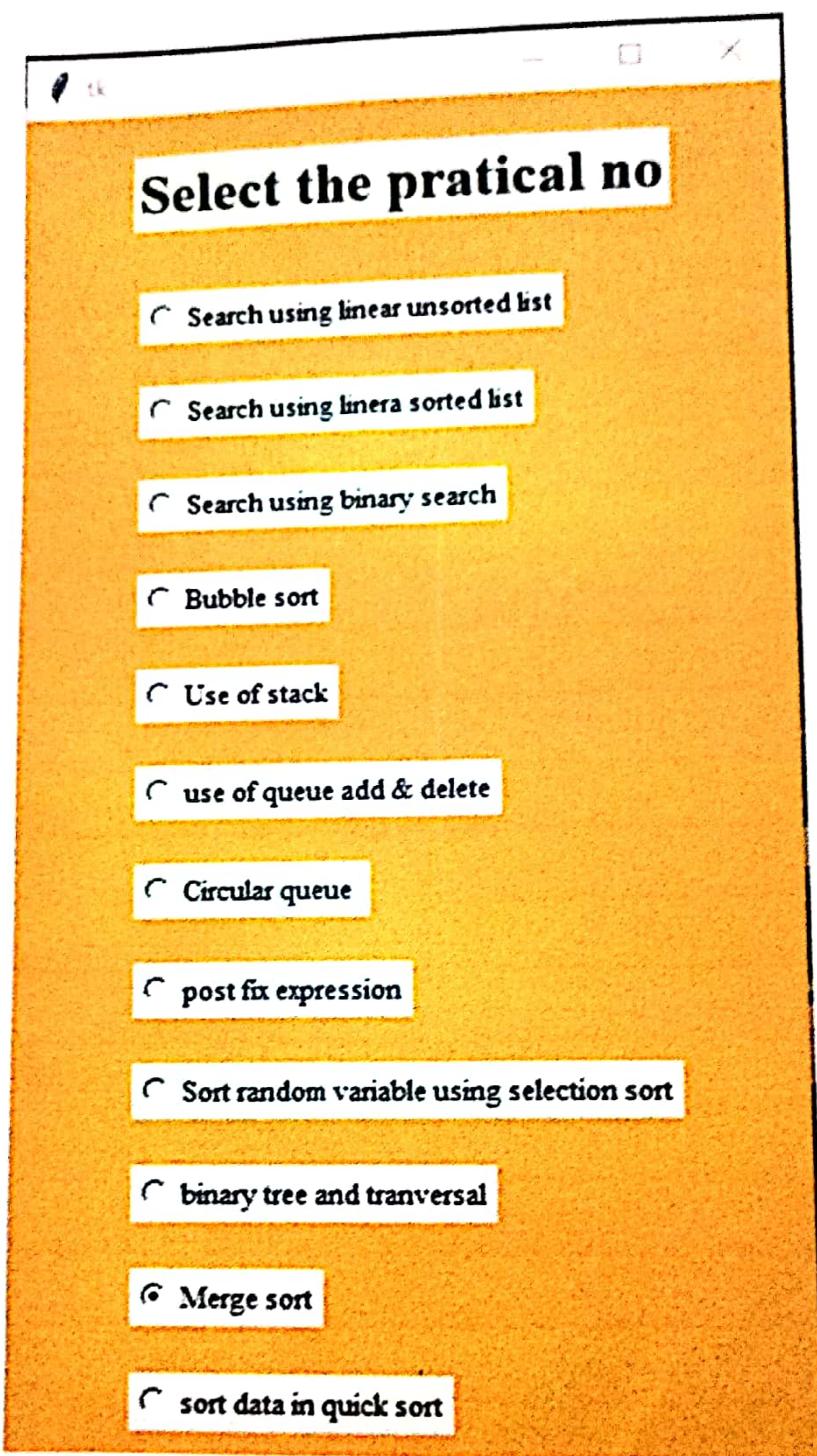
```

# *Appropriate selection*

**Select the programming language subject**

- (DS) Data Sturcture
- (PYT) Python
- (C) C Programming





```

 italic")
 l.pack(pady=20)

 code final
 12=Label(h)
 12.pack(pady=10)

 11=Label(h,text="Select the programming language subject ",font="Comic 13
bold")
 11.pack(anchor=NW,padx=45)
 v=IntVar()
 r1=Radiobutton(h,text="(DS) Data Sturcture",font="times
10",variable=v,value=1,command=ds)
 r1.pack(anchor=NW,padx=55)
 r2=Radiobutton(h,text="(PYT) Python",font="times
10",variable=v,value=2,command=pyt)
 r2.pack(anchor=NW,padx=55)
 r3=Radiobutton(h,text="(C) C Programming",font="times
10",variable=v,value=3,command=c_prog)
 r3.pack(anchor=NW,padx=55)
 h.mainloop()

def password():
 k=Tk()
 k.configure(background="red")
 l=Label(k,text="""
Username:- abhinav3396""",font="System 10",bg="red")
 l.pack()
 l1=Label(k,text="""
Password:- nothing""",font="System 10",bg="red")
 l1.pack()
 k.mainloop()

w.configure(background="white")
w.geometry('450x600')
#option menu#
languageList = ["English(UK)", "ENGLISH", "shiv sir"]
selectedLanguage=StringVar()
selectedLanguage.set(languageList[0])
languageMenu=OptionMenu(w,selectedLanguage,*languageList)
languageMenu.pack()
l=Label(w)
l.pack(anchor=NW,pady=20)
c=Canvas(w,width=330,height=264)
c.pack()
i=PhotoImage(file="LOGO.gif")
c.create_image(0,0,anchor=NW,image=i)
l=Label(w)
l.pack(anchor=NW,pady=10)

l=Label(w,text="UserName :-",font=" Times 20")
l.pack(anchor=NW,padx=55)

entry=Entry(w,bd=10,insertwidth=1,font=14,justify='center')
entry.pack(anchor=W,padx=55)

l=Label(w,text="Password :- ",font=" Times 20")
l.pack(anchor=NW,padx=55)

entry=Entry(w,bd=10,insertwidth=1,show="*",font=14,justify='center')
entry.pack(anchor=W,padx=55)

b=Button(w,text="Login",font="Comic 10
bold",bg="#00FFFF",width=30,command=login)
b.pack(anchor=CENTER,padx=20)

```

code final

```
mess=Message(w,text="forgotten your login details")
mess.pack(anchor=N)
b1=Button(w,text="get help with signing",bg="white",command=password)
b1.pack()
w.mainloop()
```

## Python 3.7.4 Shell

File Edit Shell Debug Options Window Help

Python 3.7.4 (tags/v3.7.4:e09359112e, Jul 8 2019, 19:29:22) [MSC v.1916 32 bit (Intel)] on win32  
Type "help", "copyright", "credits" or "license()" for more information.

>>>

===== RESTART: C:\Users\Shree\Desktop\python project\abhinav\fisrtpage.py =====

```
def sort(arr,l,m,r):
 n1=m-l+1
 n2=r-m
 L=[0]*n1
 R=[0]*n2
 for i in range(0,n1):
 L[i]=arr[l+i]
 for j in range(0,n2):
 R[j]=arr[m+i+j]
 i=0
 j=0
 k=l
 while i<n1 and j<n2:
 if L[i]<=R[j]:
 arr[k]=L[i]
 i+=1
 else:
 arr[k]=R[j]
 j+=1
 k+=1
 while i<n1:
 arr[k]=L[i]
 i+=1
 k+=1
 while j<n2:
 arr[k]=R[j]
 j+=1
 k+=1
def mergesort(arr,l,r):
 if l<r:
 m=int((l+(r-1))/2)
 mergesort(arr,l,m)
 mergesort(arr,m+1,r)
 sort(arr,l,m,r)
arr=[12,23,34,56,78,45,86,98,42]
print(arr)
```

```
Python 3.7.4 (tags/v3.7.4:e09359112e, Jul 8 2019, 20:34:20) [MSC v.1916 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> import os,sqlite3

>>> connection=sqlite3.connect("population.db")
>>> c=connection.cursor()

>>> c.execute("create table AndhraPradesh(District char,Population int,Area_kmsq int)")
<sqlite3.Cursor object at 0x000001985DF0A340>
>>> c.execute("insert into AndhraPradesh values('Anantapur',4083315,19130)")
<sqlite3.Cursor object at 0x000001985DF0A340>
>>> c.execute("insert into AndhraPradesh values('Chittoor',4170468,15152)")
<sqlite3.Cursor object at 0x000001985DF0A340>
>>> c.execute("insert into AndhraPradesh values('EastGodavari',5151549,10207)")
<sqlite3.Cursor object at 0x000001985DF0A340>
>>> c.execute("insert into AndhraPradesh values('Guntur',4889230,11391)")
<sqlite3.Cursor object at 0x000001985DF0A340>
>>> c.execute("insert into AndhraPradesh values('YSR Kadapa',2884524,15351)")
<sqlite3.Cursor object at 0x000001985DF0A340>
>>> c.execute("insert into AndhraPradesh values('Krishna',4529009,8727)")
<sqlite3.Cursor object at 0x000001985DF0A340>
>>> c.execute("insert into AndhraPradesh values('Kurnool',4046601,17658)")
<sqlite3.Cursor object at 0x000001985DF0A340>
>>> c.execute("insert into AndhraPradesh values('Nellore',296608,13076)")
<sqlite3.Cursor object at 0x000001985DF0A340>
>>> c.execute("insert into AndhraPradesh values('Prakasam',3392764,17626)")
```

## PRACTICAL (Project) done

- \* Aim:- To learn database management in python & sql.

It will help us to recognize the growth rate of the every state with the base of population.

```
<sqlite3.Cursor object at 0x000001985DF0A340>
>>> c.execute("insert into AndhraPradesh values('Srikakulam',2699471,5837)")
<sqlite3.Cursor object at 0x000001985DF0A340>
>>> c.execute("insert into AndhraPradesh values('Visakhapatnam',4288113,11161)")
<sqlite3.Cursor object at 0x000001985DF0A340>
>>> c.execute("insert into AndhraPradesh values('Vizianagaram',2342868,6539)")
<sqlite3.Cursor object at 0x000001985DF0A340>
>>> c.execute("insert into AndhraPradesh values('WestGodavari',3934782,7742)")
<sqlite3.Cursor object at 0x000001985DF0A340>
>>> c.execute("create table Kerala(District char,Population int,Area_kmsq int)")
<sqlite3.Cursor object at 0x000001985DF0A340>
>>> c.execute("insert into Kerala values('Malappuram',4110956,3550)")
<sqlite3.Cursor object at 0x000001985DF0A340>
>>> c.execute("insert into Kerala values('Thiruvananthapuram',3307284,2192)")
<sqlite3.Cursor object at 0x000001985DF0A340>
>>> c.execute("insert into Kerala values('Ernakulam',3279860,3086)")
<sqlite3.Cursor object at 0x000001985DF0A340>
>>> c.execute("insert into Kerala values('Thrissur',3110327,3032)")
<sqlite3.Cursor object at 0x000001985DF0A340>
>>> c.execute("insert into Kerala values('Kozhikode',3089543,2344)")
<sqlite3.Cursor object at 0x000001985DF0A340>
>>> c.execute("insert into Kerala values('Palakkad',2810892,4480)")
<sqlite3.Cursor object at 0x000001985DF0A340>
>>> c.execute("insert into Kerala values('Kollam',2629702,2491)")
```

```
<sqlite3.Cursor object at 0x000001985DF0A340>
>>> c.execute("insert into Kerala values('Kannur',2525637,2966)")
<sqlite3.Cursor object at 0x000001985DF0A340>
>>> c.execute("insert into Kerala values('Alappuzha',2121943,1414)")
<sqlite3.Cursor object at 0x000001985DF0A340>
>>> c.execute("insert into Kerala values('Kottayam',1979384,2208)")
<sqlite3.Cursor object at 0x000001985DF0A340>
>>> c.execute("insert into Kerala values('Kasaragod',1302600,1992)")
<sqlite3.Cursor object at 0x000001985DF0A340>
>>> c.execute("insert into Kerala values('Pathanamithitta',1195537,2637)")
<sqlite3.Cursor object at 0x000001985DF0A340>
>>> c.execute("insert into Kerala values('Idukki',1107453,4358)")
<sqlite3.Cursor object at 0x000001985DF0A340>
>>> c.execute("insert into Kerala values('Wayanad',816558,2131)")

<sqlite3.Cursor object at 0x000001985DF0A340>
>>> connection.commit()

>>> c.execute('select * from AndhraPradesh')

<sqlite3.Cursor object at 0x000001985DF0A340>

>>> c.fetchall()
[('Anantapur', 4083315, 19130), ('Chittoor', 4170468, 15152), ('EastGodavari', 5151549, 10807), ('Guntur', 4889230, 11391), ('YSR Kadapa', 2884524, 15351), ('Krishna', 4529009, 8727), ('Kurnool', 4046601, 17658), ('Nellore', 296608, 13076), ('Prakasam', 3392764, 17626), ('Srikakulam', 2699471, 5837), ('Visakhapatnam', 4288113, 11161), ('Vizianagaram', 2342868, 6539), ('WestGodavari', 3934782, 7742)]

>>> c.execute('select * from Kerala')

<sqlite3.Cursor object at 0x000001985DF0A340>
```

```
>>> c.fetchall()
[('Malappuram', 4110956, 3550), ('Thiruvananthapuram', 3307284, 2192), ('Ernakulam', 3279860, 3086),
('Thrissur', 3110327, 3032), ('Kozhikode', 3089543, 2344), ('Palakkad', 2810892, 4480), ('Kollam', 2629702,
2491), ('Kannur', 2525637, 2966), ('Alappuzha', 2121943, 1414), ('Kottayam', 1979384, 2208), ('Kasaragod',
1302600, 1992), ('Pathanamithitta', 1195537, 2637), ('Idukki', 1107453, 4358), ('Wayanad', 816558, 2131)]
>>> c.execute('select District,Population from AndhraPradesh union select District,Population from Kerala')
<sqlite3.Cursor object at 0x000001985DF0A340>
>>> c.fetchall()
[('Alappuzha', 2121943), ('Anantapur', 4083315), ('Chittoor', 4170468), ('EastGodavari', 5151549), ('Ernakula
3279860), ('Guntur', 4889230), ('Idukki', 1107453), ('Kannur', 2525637), ('Kasaragod', 1302600), ('Kollam',
2629702), ('Kottayam', 1979384), ('Kozhikode', 3089543), ('Krishna', 4529009), ('Kurnool', 4046601),
('Malappuram', 4110956), ('Nellore', 296608), ('Palakkad', 2810892), ('Pathanamithitta', 1195537), ('Prakasai
3392764), ('Srikakulam', 2699471), ('Thiruvananthapuram', 3307284), ('Thrissur', 3110327), ('Visakhapatnam
4288113), ('Vizinagaram', 2342868), ('Wayanad', 816558), ('WestGodavari', 3934782), ('YSR Kadapa', 288452
>>> c.execute('select count(District) from AndhraPradesh')
<sqlite3.Cursor object at 0x000001985DF0A340>
>>> c.fetchall()
[(13,)]
>>> c.execute('select count(District) from Kerala')
<sqlite3.Cursor object at 0x000001985DF0A340>
>>> c.fetchall()
[(14,)]
>>> c.execute('select sum(Population) from AndhraPradesh')
<sqlite3.Cursor object at 0x000001985DF0A340>
>>> c.fetchall()
[(46709302,)]
```

```
>>> c.execute('select sum(Population) from Kerala')
```

Q40

```
<sqlite3.Cursor object at 0x000001985DF0A340>
>>> c.fetchall()
[(33387676,)]
>>> c.execute("drop table Andhrapradesh")
<sqlite3.Cursor object at 0x00000269D76DB5E0>
>>> c.execute("drop table Kerala")
<sqlite3.Cursor object at 0x00000269D76DB5E0>
>>> c.fetchall()
[]
```

Q50 ✓

Q51 ✓