

## Practical No - 1

## Modes of file operation

## 1. Write mode (w)

# To Create a file

a = open("m", "w")

a.write("Python is an intended language")

# To write contents in the file.

a.close()

# To close a file

## 2. read mode (r)

a = open("m", "r")

# To open a file in read mode

w = a.read()

# To read a written file

print(w)

# To display the output

a.close()

# To close a file

## Read methods

## 1. read()

a = open("m", "r")

# To open a file in read mode

st1 = a.read()

# store in a variable

print(st1)

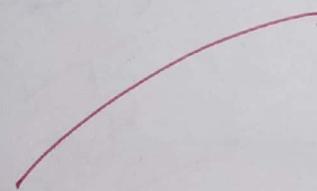
# Output

a.close()

# Close a file

Output :

Python is an intended language



Python is an intended language

Python is an interpreted language

1

2 [Python is an interpreted language]



## 2. readline()

```
a = open("m", "r")
st2 = a.readline()
```

# Open a file in read mode  
# Store output of readline method.

```
print(st2)
a.close()
```

# Output  
# Close a file.

## 3. readlines()

```
a = open("m", "r")
st3 = a.readlines()
```

# Open a file in read mode  
# Store output of readlines method

```
print(st3)
a.close()
```

# Output  
# Close a file.

~~incorrect~~

Objective : Demonstrate the use of different file accessing mode, different attributes Read method.

Step 1 : Create a file Object using open method and use the write access mode followed by writing some content onto the file and then closing the file.

Step 2 : Now open the file in read mode than use read(), readline() and readlines() and store the output in variable and finally display the contents of variable.

Step 3 : Now use the fileObject 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.

# open mode

27

a = open ("m.txt", "w")

a.write ("DS")

a.close()

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

sts3 = a.read()

print (sts3)

a.close

"DS"

# tell()

b = open ("m.txt", "r")

pos = b.tell()

print ("tell() : ", pos)

b.close

# seek()

c = open ("m.txt", "r")

sts4 = c.seek (10, 0)

sts8 = c.read (10)

print (sts8)

# finding length

d = open ("m.txt", "r")

sts9 = d.readlines()

print ("Output : ", sts9)

for line in sts9:  
 print (len (line))

d.close()

Code:

```
mytuple ("A", "B", "C", "D")
```

```
myiter = iter(mytuple)
```

```
print(next(myiter))
```

```
print(next(myiter))
```

```
print(next(myiter))
```

```
print(next(myiter))
```

Output:

A

B

C

D

```
my = ("A", "B", "C", "D")
```

```
for a in my:
```

```
    print(a)
```

Output:

A

B

C

D

Final ✓

## Practical No. 2

Aim : To display elements of a tuple using iterator method

Algorithm :

Step 1 : Form a tuple with elements.

Step 2 : Use iter method with tuple .

Step 3 : Use next method and print elements

Aim : To use iter method with for loop

Step 1 : Form a tuple with elements

Step 2 : Use for statements to access element .

Step 3 : Print element of tuple .

Journal

85

Aim :- To display the output in the file using read mode.

Step 1 :- Open the file in read mode

Step 2 :- Use while loop to print the elements

Step 3 :- Store the values in an variable

Aim :- To search the occurrence of word in the file

Step 1 :- Open the file in write mode and enter a line of statement.

Step 2 :- Open the file in read mode and store it in a variable.

Step 3 :- Enter a word you want to find

Step 4 :- Using while loop search for the word in the statement.

Step 5 :- Close the file

Code :

```
with open ("para", "r") as a:  
    s = 1  
    c = a.read(s)  
    while len(c) > 0:  
        print(c)  
        c = a.read(s)
```

Output

C  
+  
+  
9  
S  
a  
l  
a  
n  
g  
u  
a  
g  
e

Code :

```
a = open ("mahi", "w")  
a.write ("Python is an intended c++ cccc")  
a = open ("mahi", "r")  
str = a.read()  
c = 0  
n2 = input ("What to find ")  
for word in str:  
    if word == n2:  
        c = c + 1  
print (c)  
a.close()  
Output :  
What to find c
```

Code :

class m:

def \_\_iter\_\_(self):

self.a = 1

return self

def \_\_next\_\_(self):

if self.a <= 12:

x = self.a

self.a += 1

return x

else:

raise StopIteration

s = m()

myiter = iter(s)

for n in myiter:

print(n)

Output:

1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12

Aim : To print number of given range.

Step 1 : Define a class and define two functions in it.

Step 2 : First function will return self and other one will return the value of any object till the iteration is complete.

Step 3 : Use if conditional statement to check the conditions and return the value of object otherwise stop iteration.

Aim : To find odd numbers in given  
using map method.

Step 1 : Declare a list variable with  
elements then use map method  
with lambda function give  
two argument & play the output

Step 2 : Define a function even with a parameter  
then using conditional statement do  
check whether the number is even  
or odd.

Aim : To print odd numbers using iterator

Step 1 : Define a class and define the  
method which will initialise the  
first element within the construct  
object.

Step 2 : Now use the next() and define  
the logic for displaying odd  
value.

code :

```
a = [0, 4, 5, 7, 9, 11, 13, 15, 20, 19, 21]
```

```
b = list(map(lambda x: x % 5, listnum))
```

```
print(listnum)
```

```
def even(x):
```

```
    if (x % 2 == 0):
```

```
        return "Even"
```

```
    else:
```

```
        return "Odd"
```

```
list(map(even, listnum))
```

Output

```
[0, 4, 0, 2, 4, 1, 3, 0, 0, 4, 1]
```

code :

```
class Odd:
```

```
    def __iter__(self):
```

```
        self.num = 1
```

```
        return self
```

```
    def __next__(self):
```

```
        num = self.num
```

```
        self.num += 2
```

```
    def __next__(self):
```

```
        num = self.num
```

```
        self.num += 2
```

```
        return num
```

Q8

```
myobj = odd()  
myiter = iter(myobj)  
x = int(input("Enter a number : "))  
for i in myiter:  
    if (i < x):  
        print(i)
```

Output:

Enter a number 11

1  
3  
5  
7  
9

# Square and Cube

```
def square(x):  
    y = x * x  
    return y  
def cube(x):  
    z = x * x * x  
    return z  
func = [square, cube]
```

for i in range(5):

```
    value = list(map(lambda x: x(i), func))  
    print(value)
```

Output:

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

Step 3 : Define an object of a class

Step 4 : Accept a number from the user till which we want to display the odd number.

Aim : To print square and cube

Step 1 : Define a function name Square with a parameter which will obtain output of square value of the given number. In similar fashion declare cube of and get the value raised 3 and return same.

~~Step 2 : Call the declared function using function call~~

Step 3 : Using for conditional statement specifying the range use the list type casting with map method declare a 'lambda' i.e anonymous function and print the same.

Aim :- To print square without using map.

~~list = [1, 2, 3, 4, 5]~~

Step 1 :- Create a list with elements.

Step 2 :- Define an empty list.

Step 3 :- Use for loop to access the numbers.

Step 4 :- Print the result.

Aim :- To print the length of the line using readline method.

Step 1 :- Open the file in read mode.

Step 2 :- Use readline method store it in a variable.

Step 3 :- Use for loop to iterate in the variable.

Step 4 :- Print the output.

Step 5 :- Close the file.

code:

```
a = [1, 2, 3, 4, 5]
x = []
for i in a:
    x.append(i * * 2)
print(x)
```

Output

```
[1, 4]
[1, 4, 9]
+ 6 [1, 4, 9, 16]
- 25 [1, 4, 9, 16, 25]
```

code:-

```
a = open("m", "r")
x = a.readlines()
print(x)
for line in x:
    print(len(line))
a.close()
```

Output:

18

98  
# Factorial

def f(x):

if (x == 1):

return 1

else:

return (x \* f(x-1))

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

list = [ ]

list.append(x)

a = map(f, list)

print("The factorial is: ", a)

Enter a number : 5

The factorial is 120

Step 1 :- Define a function which will accept a input to find factorial.

Step 2 :- Then Define a empty list

Step 3 :- Use append method to pass the value of input

Step 4 :- Use map method with function name and a empty list.

Step 5 :- Print the output

Jan  
(11/11)

Z

## Practical - 3

Write a program using the exception block related to the environment Error?

Algorithm:

Step 1 :- Use the try block to define normal character of action for eg define the file object and open the file in the write mode & write some content onto the file.

Step 2 :- Use the except block with a IOError as an environment error and convey the appropriate error to the user else display the message that the operation is carried out successfully.

Source code:

34

```
try:  
    a=open("myself.txt","w")  
    a.write("Python is an"+ "intended language")  
except IOError:  
    print("Unsuccessfull")  
  
else:  
    print("You Rocked it")
```

Output:

You Rocked it

Source Code:

try :

a = open("myself", "x")

a.write("C is not an OOP")

except ValueError:

print("Invalid")

except IOError:

print("Not Applicable")

else :

print("Well done")

Output:

Well done

② while True :

try :

a = open("myself", "w")

a.write("C++ is an OOP")

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

print(x)

break

except IOError:

print("Environmental error")

except ValueError:

print("Value Error")

Output:

Enter a number - x

Value Error

Enter a number - 7

2. Write a program for demonstrating the use of ValueError in the given program statement.

Step 1 :- Except the value from the user & if it is a valid data display the entered value & terminate the condition by using break statement.

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

Step 3 :- We can define the multiple exception using the Except statement for finding the different categories of error.

Jm71

## Practical - 4

### Regular Expression

Step 1 :- Import re module declare pattern and declare sequence use match method with declare argument if arguments matched then print the same otherwise print pattern NOT FOUND!

Step 2 :- Import re module declare pattern with literal and meta character. Declare string value - use the.findall() with arguments and print the same.

Step 3 :- Import re module declare pattern with meta character use the.split() and print the output

```
#match()
import re
pattern = re.compile("FYCS")
sequence = "FYCS represents Computer Science"
if re.match(pattern, sequence):
    print("Matched pattern found!")
else:
    print("Not Found")
```

Output:  
Matched pattern found!

```
#numerical values (segregation)
```

```
import re
pattern = re.compile("\d+")
string = 'Hello123, Howdy143, 57Howsv'
output = re.findall(pattern, string)
print(output)
```

Output:  
['123', '143', '57']

```
#split()
```

```
import re
pattern = re.compile("\d+")
string = 'Hello123, Howdy143, 57Howsv'
output = re.split(pattern, string)
print(output)
```

Output:  
['Hello', 'Howdy', ' ', ' ', 'Howsv']

```
# no space
import re
string = 'abc def ghi'
pattern = r'\st'
replace = ''
v1 = re.sub(pattern, replace, string)
print(v1)
```

Output:

abcdefghi

```
# group()
```

```
import re
```

```
sequence = 'Python is an OOP'
```

```
v = re.search('Apython', sequence)
```

```
print(v)
```

```
v1 = v.group()
```

```
print(v1)
```

Output:

<\_sre.SRE\_Match object at 0x0281DF00>  
python

```
# verifying the given set of phone numbers
```

```
import re
```

```
list1 = ['8004567891', '9145673210', '786543210',
         '9876543201']
```

```
for value in list1:
```

```
    if re.match(r'[8-9]{1}\d{1}[0-9]{9}')
```

value or len(value) == 10:

```
            print("criteria matched for cell number")
```

```
        else:
            print("criteria failed!")
```

Step 4 : import re module declare string and accordingly declare pattern replace the blank space with no-space use sub() with 3 arguments and print the string without spaces.

Step 5 : import re module declare a sequence use search method for finding subsequently use the group() with dot operator as search() gives memory location using group() it will show up the matched string.

Step 6 : import re module declare list with numbers. use the conditional statement here we have used up the for condition statement. use if condition for checking first number is either 8 or 9 and next numbers are in range of 0 to 9 and check whether the entered numbers are equal to 10 if criteria matches print cell number matches otherwise print failed.

58

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

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

Step 9 : import re module enter a string as pattern to display Only two characters of the partition string . Use.findall() declare two variables with initial value as zero use for condition and subsequently use the if condition check whether the condition satisfied add up the or else increment value and display the value subsequently

output:

criteria matched for cell number

criteria matched for cell number

criteria failed!

criteria matched for cell number

# vowels

import re

str1 = 'C++ is an oop'

output = re.findall(r'1b[aeiouAEIOU]1wt', str1)

print(output)

Output:

[ 'is', 'oop' ]

# host and domain

import re

seq = 'abc.tcc@edu.com. XYZ@gmail.com'

pattern = r'[w1.-]+[w1.-]+'

output = re.findall(pattern, seq)

print(output)

Output:

[ 'abc.tcc', 'edu.com', 'XYZ', 'gmail.com' ]

Jmz

88

# Counting of first 2 letters

import re

s = 'mr.a, ms.b, mr.c, mr.t'

p = s'[ms/mr]+

o = re.findall(p, s)

print(o)

m = 0

f = 0

for v in o:

if (v == 'ms'):

f = f + 1

else:

m = m + 1

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

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

Output :

['mr', 'ms', 'ms', 'mr']

('No. of males is : 1, 2)

('No. of females is : 1, 2)

EE

## Practical - 5

Aim :- To make use of GUI.

Algorithm :-

Step 1 :- Use tkinter library to import widgets.

Step 2 :- Create a variable and position it onto the parent window.

Step 3 :- Use the pack() with object created and use parameters  
side = TOP  
padx = 20  
ipadx = 40  
ipady = 50

Step 4 :- Use mainloop() for triggering corresponding events.

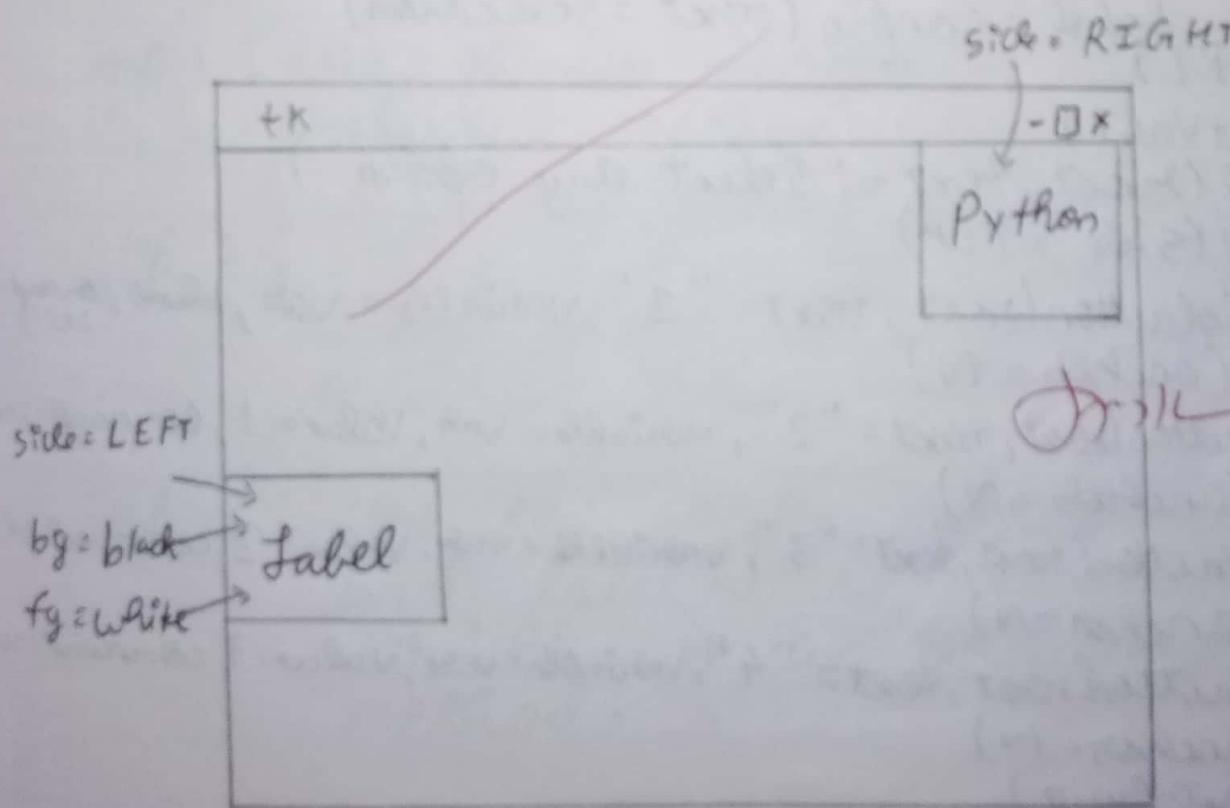
Step 5 :- Now repeat steps with label method which takes argument.

## Source Code :

40

```
from tkinter import *
root = Tk()
T1 = Text(root)
T1.insert(END, "Python")
T1.pack(side=RIGHT, padx=20, pady=30, ipadx=40, ipady=50)
L1 = Label(root, text="Label", bg="black",
           fg="white")
L1.pack(side=LEFT, ipady=10, ipadx=20, ipady=50)
root.mainloop()
```

## Output:



Source ~~Op~~ Code :-

```
from tkinter import *
def sel 1():
    selection = "Mahesh"
    Label.config(text=selection)
def sel 2():
    selection = "Python"
    Label.config(text=selection)
def sel 3():
    selection = "GVI"
    Label.config(text=selection)
def sel 4():
    selection = "COP"
    Label.config(text=selection)
root = Tk()
var = IntVar()
L1 = Label(root, text="Select any option")
L1.pack(side=TOP)
R1 = Radiobutton(root, text="1", variable=var, value=1)
R1.pack(anchor=N)
R2 = Radiobutton(root, text="2", variable=var, value=1, command=R2.pack(anchor=N))
R3 = Radiobutton(root, text="3", variable=var, value=2, command=R3.pack(anchor=N))
R4 = Radiobutton(root, text="4", variable=var, value=3, command=R4.pack(anchor=N))
label = Label(root)
label.pack(side=BOTTOM)
root.mainloop()
```

(i) Name of parent window

(ii) Text attribute

(iii) Background colour

(iv) Foreground colour

and use pack() with relevant attributes.

### RadioButton

Aim : To make use of RadioButton widget for selection of one of the option.

Algorithm :

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

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

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

Step 4 : Now define the parent window and define option using control variable.

14

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

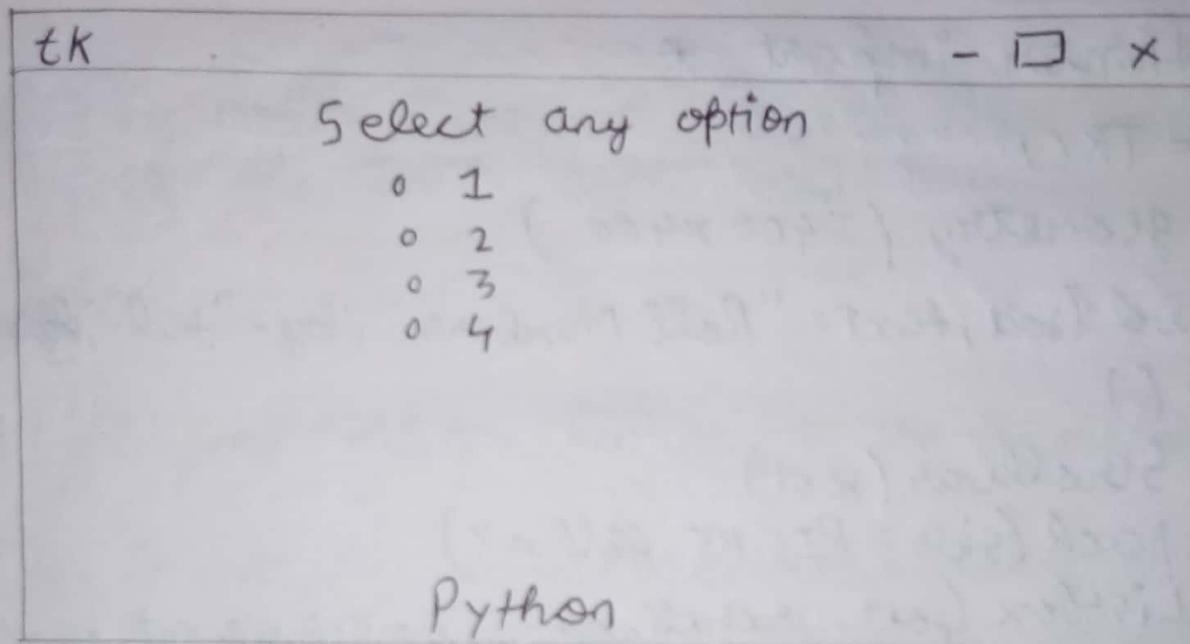
- ① Positioning on Parent Window
- ② Text Variable
- ③ Define variable argument
- ④ Corresponding value and trigger the given function.

Step 6 :- Now call the pack method on corresponding Radio object created and specify argument as an anchor .  
Attributes

Step 7 :- Now define a label object and place it onto parent window using pack method and for its mainloop method.

OUTPUT:

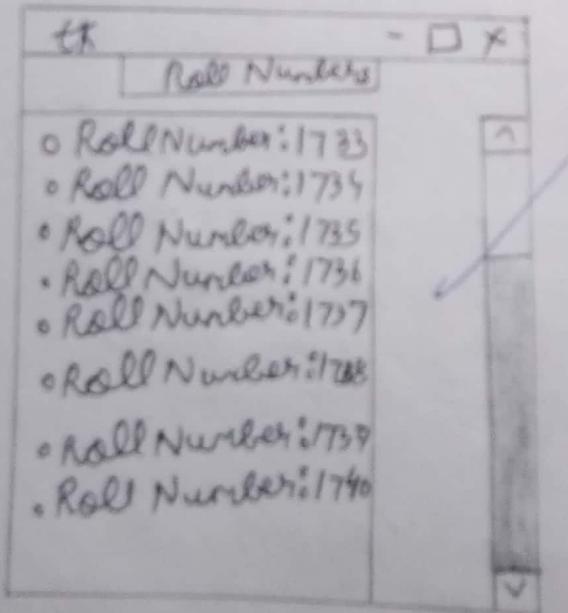
42



## Source Code :

```
from tkinter import *
root = Tk()
root.geometry ("400x400")
L = Label (root, text = "Roll Numbers", bg = "red", fg = "blue")
L.pack ()
scroll = Scrollbar (root)
scroll.pack (side = RIGHT, fill = Y)
mylist = Listbox (root, yscrollcommand = scroll.set, bg = "yellow")
for num in range (1, 40):
    mylist.insert (END, "o Roll Number: " + str (num))
mylist.pack (side = LEFT, fill = BOTH)
scroll.config (command = mylist.yview)
root.mainloop ()
```

## Output :



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

Algorithm :

Step 1 : Import tkinter library to use scrollbar widget.

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

Step 3 : Create an object using Label method and place it on parent window.

Step 4 : Use pack method and use side and fill

Step 5 :- Create another object using Listbox method and ~~use~~<sup>place it on</sup> parent window with attribute yscroll command.

Step 6 : Use for loop to insert values to ~~in~~<sup>to</sup> object of listbox using insert method.

Step 7 : Use config method along with scroll bar object and use command attribute.

Step 8 : Finally call mainloop.

Aim:- Write a program to display image concept of frame, toolbar, grid and Button Method.

Algorithm :

Step 1 :- Create an object corresponding to parent window and use following 3 methods ① title ② maxsize ③ config.

Step 2 :- Create left frame object from the frame and place it on parent window with height, width and bg. Use the grid method with the row, column, padx and pady attribute.

Step 3 :- Now, create a right frame from frame method with width and height specified and the row and column value.

Step 4 :- Create a label object from label method and place it on to the left frame with text attribute as raised value and use grid method with row, column and columnspan.

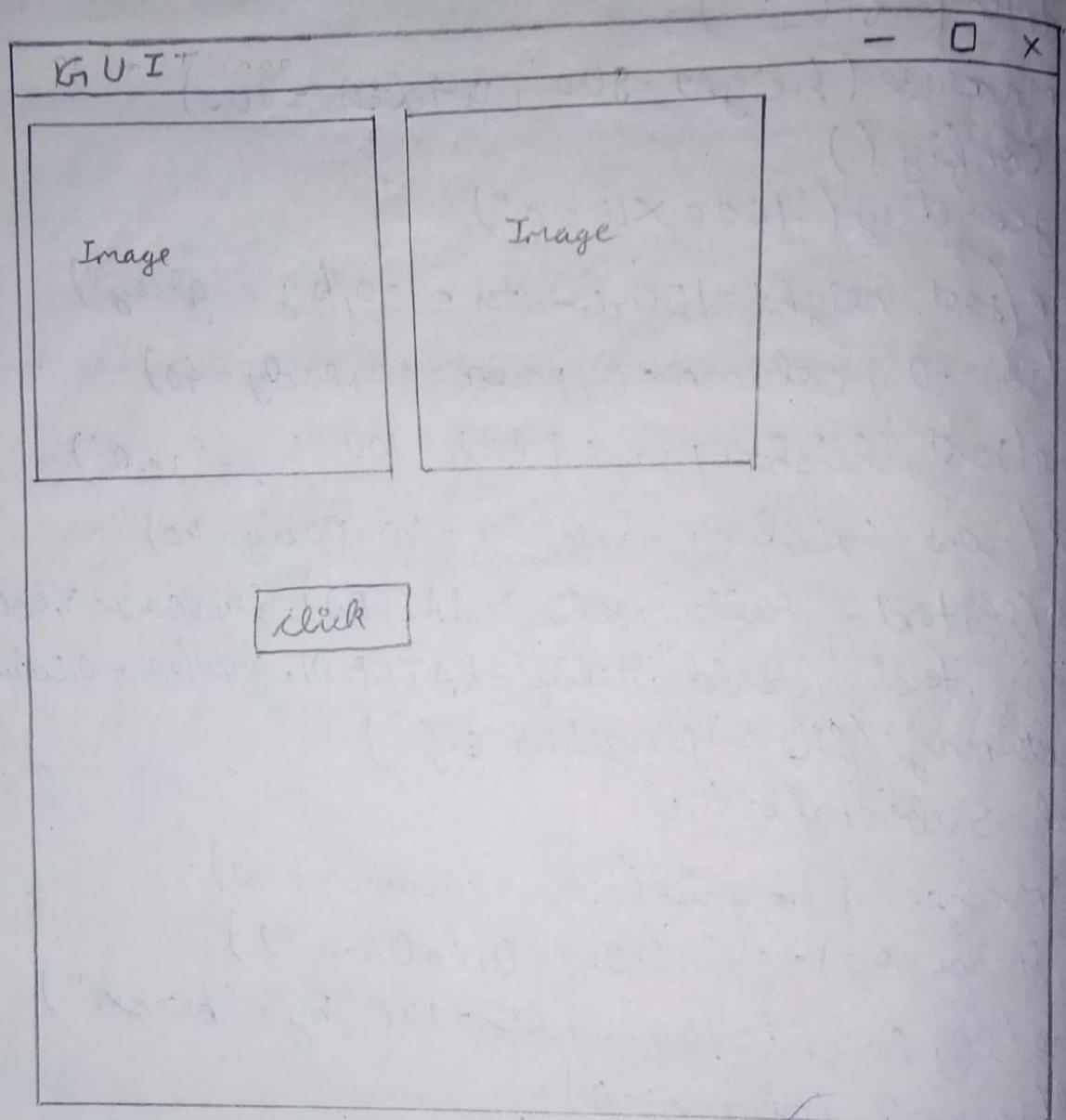
Source code:

Output & Source code:

44

```
from tkinter import *
root = Tk()
root.title("GUI")
root.maxsize(height=800, width=800)
root.config()
root.geometry("1000x1000")
lf = Frame(root, height=100, width=100, bg="grey")
lf.grid(row=0, column=0, padx=40, pady=40)
rf = Frame(root, height=100, width=100, bg="red")
rf.grid(row=0, column=1, padx=40, pady=40)
l1 = Label(lf, text="Hello", relief=RAISED).grid(row=0, column=0)
l2 = Label(rf, text="World", relief=RAISED).grid(row=0, column=1)
mh = PhotoImage(file="penguins.gif")
og = mh.subsample(3, 4)
Label(lf, image=og).grid(row=0, column=0)
Label(rf, image=og).grid(row=0, column=1)
tb = Frame(lf, height=100, width=100, bg="black")
tb.grid(row=2, column=0)
def ab():
    print("Well done")
    Label(tb, text="Tools", relief=SUNKEN, fg="red").grid(row=2, column=0)
    Button(tb, text="Click", command=ab).grid(row=2, column=0)
root.mainloop()
```

Output :



Step 5 :- Now, use the photo image method with the file attribute specified.

Step 6 :- Use the sub-sample method with the object of the image and give x and y co-ordinate values.

Step 7 :- Use label method and position it onto the left frame and use grid method for positioning in first row.

Step 8 :- Create another label object positioning right frame and specifying the image and bg attribute with row and column attribute.

Step 9 :- Create tool bar object from frame method and position it onto the left frame with height and width specified and position it onto the second row.

Step 10 :- Now, define the various function for the different toolbar option provided in the left frame.

Step 11 :- From Label() position text onto the tools we relief and corresponding grid value and incorporate internal padding as well.

Step 12 :- Create label method position it onto the parent with next file as next and position it on the same row but next column.

Step 13 :- Use Button widget and call the function using command.

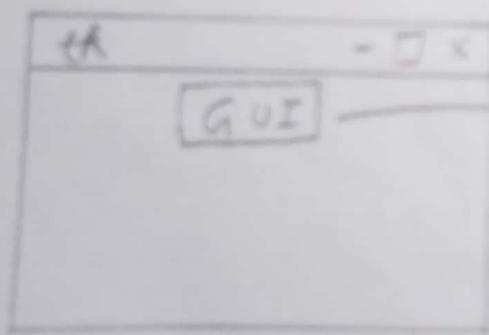
Step 14 :- Finally, use mainloop() to trigger all events.

31

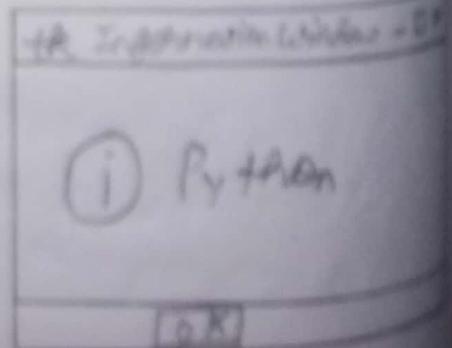
### Source code

```
from tkinter import *
import messagebox
root = Tk()
def x():
    messagebox.showinfo("Information Window",
                        "GUI")
    b1 = Button(root, text = "python", command = x)
    b1.pack()
root.mainloop()
```

### Output:



After click on the button root window will appear and pop up message.



Aim :- To demonstrate use of message box.

Step 1 : Import relevant methods from tkinter library

Step 2 : Import tk messagebox

Step 3 : Define a parent window object along with the parent window.

Step 4 : Define a function which will use tk messagebox with showinfo method along with info window attribute.

Step 5 : Declare a button with parent window object along with command attribute.

Step 6 : Place the button widget onto the parent window and finally call mainloop() for triggering of the events called above.

Jn 101'2

Aim : To perform program on traversal of window.

Algorithm :-

Step 1 :- Import the relevant methods from tkinter library.

Step 2 :- Define a function and create object of a 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 4 :- Define another function which will use the quit method to terminate the program.

source code :

```

from tkinter import *
def main():
    root = Tk()
    root.geometry("500x500")
    root.config(bg="light green")
    root.title("GUI")
    root.minsize(height=200, width=300)
    B3 = Button(root, text="Next", command=main)
    B3.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("500x500")
tos.config(bg="red")
tos.title("Trial")
tos.minsize(height=200, width=300)

B1 = Button(tos, text="Continue", command=main)
B1.grid(ipadx=50, ipady=40, padx=20, pady=30)
B2 = Button(tos, text="Exit", command=term)
B2.grid(ipadx=50, ipady=40, padx=20, pady=30)

```

def main1():

top = Tk()

top.geometry ("500x500")

top.config (bg = "blue")

top.title ("Traversing")

top.minsize (height = 200, width = 300)

B1 = Button (top, text = "Main page", command =

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

B2 = Button (top, text = "Exit", command = exit)

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

def main2():

obj = Tk()

obj.geometry ("500x500")

obj.config (bg = "red")

obj.title ("Python")

obj.minsize (height = 200, width = 300)

mainloop()

Step 5 :- Now Create an object of main window and use various methods like config, title, geometry etc.

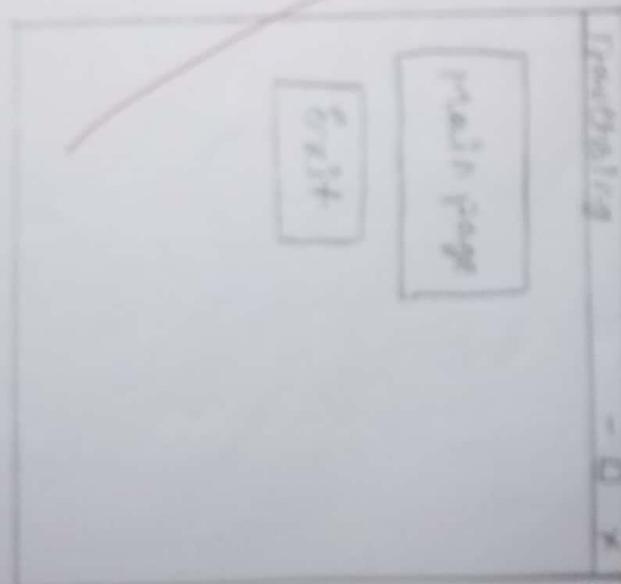
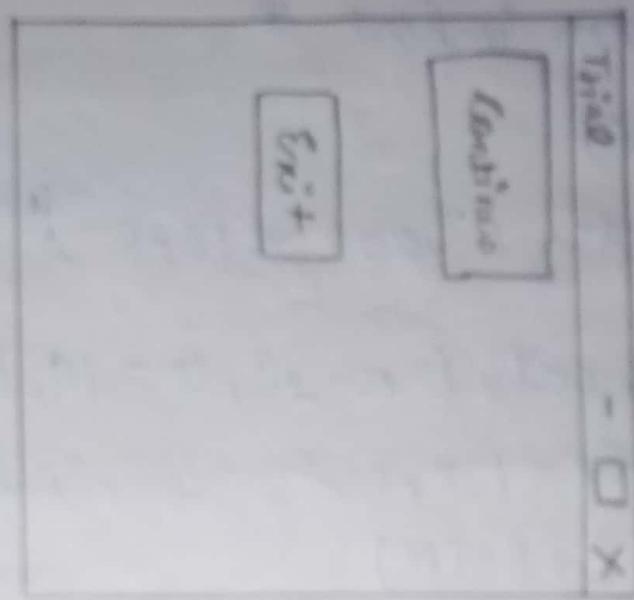
Step 6 :- Define two buttons which will be placed on the main window; One to cover another window and the other to terminate the program.

Step 7 :- Define another function which will carry various button placed on this window. Define two buttons respectively and use the grid method along with two buttons.

Step 8 :- Finally call the mainloop method.

Output:

50



source file:  
from tkinter import \*

for root = Tk()

root.geometry ("1000x1000")

s1 = Spinbox(root from=0 to = 10, bg = "orange")

s1 . pack (padx = 10, pady = 10, ipadx = 10, ipady,

(fill = BOTH)

label s1, text = "SPINBOX", bg = "black", fg = "white"  
pack (fill = Y, padx = 10, pady = 10)

root . mainloop()

## Spinbox

Aim :- To implement use of Spinbox widget

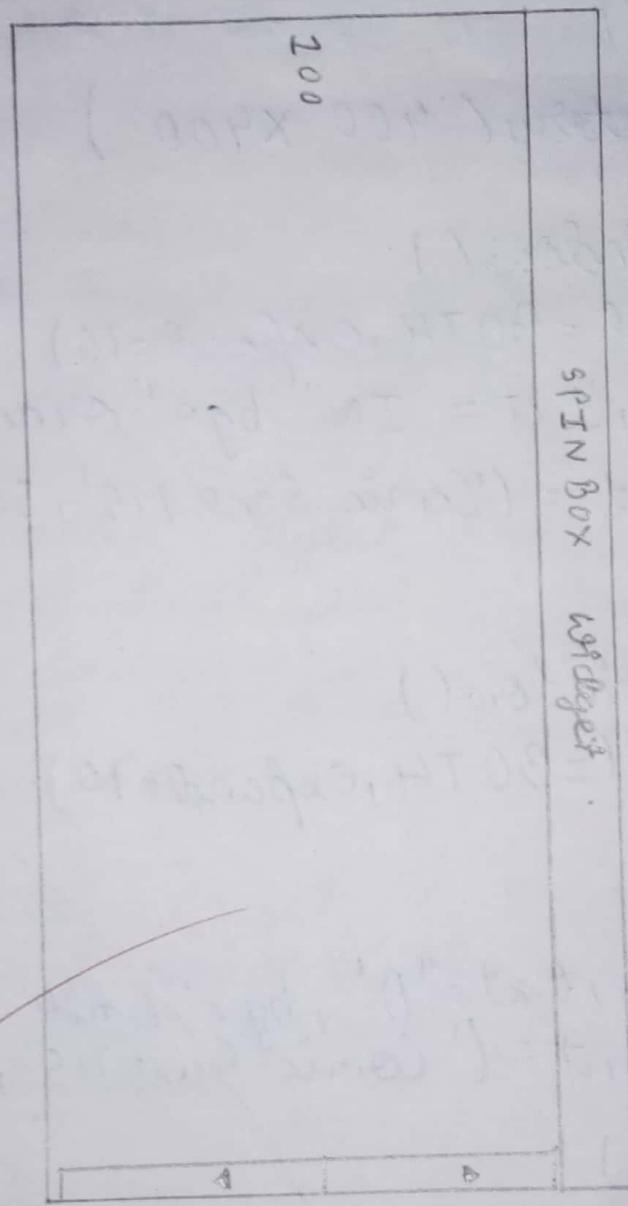
Algorithm :-

Step 1 :- Create object from TK() and  
Subsequently create object from  
Spinbox()

Step 2 :- Make the object so created onto  
the parent window and trigger  
the corresponding events.

Jn 17/2

Output



SPIN BOX  
W/ Lever

Source code:

from tkinter import \*

root = Tk()

root.geometry("400 x400")

P1 = PanedWindow()

P1.pack(fill=BOTH, expand=10)

L1 = Label(P1, text="IN", bg="orange", fg="black", font=("Comic Sans MS", 50))

P1.add(L1)

P2 = PanedWindow()

P2.pack(fill=BOTH, expand=10)

P1.add(P2)

L2 = Label(P2, text="O", bg="white", fg="black", font=("Comic Sans MS", 50))

P2.add(L2)

P3 = PanedWindow()

P2.add(P3)

L3 = Label(P3, text="IA", bg="green", fg="black", font=("Comic Sans MS", 50))

P3.add(L3)

mainloop()

## Paned Window

Aim : To implement use of paned window .

Algorithm:

Step 1 :- Create object using `paned window()` and use the `pack method` with attribute `fill` and `expand`.

Step 2 :- Create an object from `label` method and put it onto the paned window with `text attribute` and use the `add method` to embed the new object.

Step 3 :- Similarly, create a second paned window object and add it onto the first paned window with orientation .

Step 4 :- Now, create another label object and place it onto the

Step 5:- Finally call the mainloop  
to trigger the events.

Output:

54

white

- 0 x

grey

I N

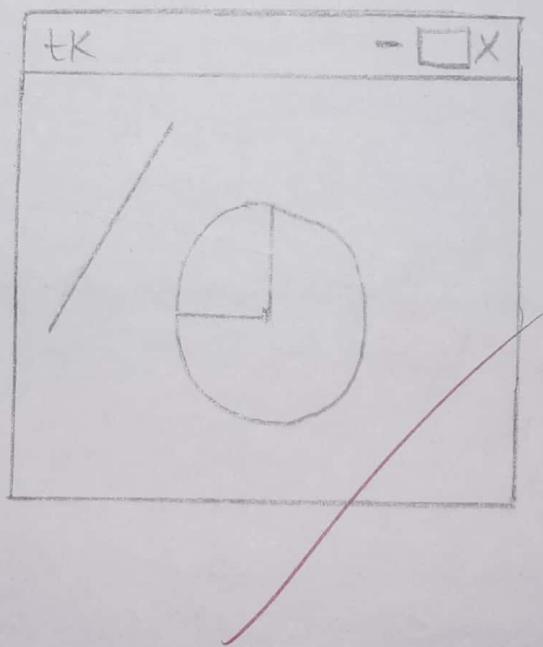
D

I A

green

## Source code:

```
from tkinter import *
root = TK()
c1 = canvas(root, height=400, width=400, bg="black")
o = c1.create_oval(20, 140, 150, 250, fill="white")
arc = c1.create_arc(30, 140, 50, 60, fill="green")
line = c1.create_line(20, 140, 150, 60, fill="blue")
c1.pack()
root.mainloop()
```



## Canvas

Aim :- To implement use of Canvas widget.

Algorithm :-

Step 1 :- Create an object from canvas method and use the attribute height, width, bg, parent window object.

Step 2 :- Use the method create\_line, create oval and create arc along with the canvas object so created and used the co-ordinate values.

Step 3 :- Similarly, use other method and call the pack method and the mainloop method

## Practical - 6

### Client Server Setup

Aim : To implement connection in python of Sqlite 3 and execute different operations.

#### Algorithm :

Step 1 :- Import corresponding library for making database connection OS, Sqlite 3.

Step 2 :- Now create the connection object using Sqlite 3 library and the connect method for creating the new database.

Step 3 :- Now create the cursor object using the cursor method from the connection Object created in Step ②

Step 4 :- Now use the execute method for creating the table with the column and the respective datatype

## Source Code and Output :-

56

```
import os,sqlite3  
connection = sqlite3.connect("student.db")  
cursor1 = connection.cursor()  
cursor1.execute("Create table tsc(stl_id int,  
roll int, name varchar, course varchar)")  
> {sqlite3::cursor object at 0X02DCAAA0}  
cursor1.execute('Insert into tsc values(1701,1760,"rahesh",  
"CS")')  
> {sqlite3::cursor object at 0X02DCAAA0}  
cursor1.execute('Insert into tsc values(1702,1761,"Dhruv",  
"CS")')  
> {sqlite3::cursor object at 0X02DCAAA0}  
cursor1.execute('Insert into tsc values(1703,1762,"Pranay",  
"CS")')  
> {sqlite3::cursor object at 0X02DCAAA0}  
cursor1.execute('Insert into tsc values(1704,1763,  
"Sushant","CS")')  
> {sqlite3::cursor object at 0X02DCAAA0}  
cursor1.execute('Insert into tsc values(1704,1762,  
"rahesh","CS")')  
> {sqlite3::cursor object at 0X02DCAAA0}  
connection.commit()
```

cursor1.execute('Select name, std\_id, course from tsc where soll > 1760')

→ {sqlite3 Cursor object at 0x02DCAA0}

cursor1.fetchall()

[('Dhev', 1702, 'CS'), ('Pranay', 1703, 'CS'), ('Sushant',  
1704, 'CS'), ('Rahul', 1704, 'CS')]

cursor1.execute('Drop table tsc')

cursor1.close()

Step 5 :- Now with the cursor object use the insert statement for entering the values corresponding to the different fields considering the datatype.

Step 6 :- Use commit method to complete the transaction using connection object.

Step 7 :- Use execute statement along with cursor object for accessing the values from database using the select from where clause.

Step 8 :- Use fetchall () for displaying the values from the table using cursor object.

Step 9 :- Using execute () drop table syntax for terminating the database and finally use the close method.

Jn 17/2

Jn 21/2

## DDL, DML statements

Algorithm:

Step 1:- Import library .OS, Sqlite3

Step 2:- Now, create the connection object using Sqlite3 library and the connect method for creating new database.

Step 3:- Create cursor object using cursor method from the connection object created in step ①

Step ④ :- Use execute method for creating a table.

Step ⑤ :- Using cursor object use insert statement to enter values.

```
import SQLite3  
connection = SQLite3.connect('test.db')  
c1 = connection.cursor()  
c1 = connection.execute  
<SQLite3::Cursor object at 0x02EC6EE0>  
c1.execute('CREATE TABLE m(name  
varchar, fees int)')  
c1.execute('Insert into m Values ("Rahil",  
1900)')  
<SQLite3::Cursor object at 0x02EC6EE0>  
c1.execute('Insert into m Values ("Dhruv",  
1500)')  
<SQLite3::Cursor object at 0x02EC6EE0>  
connection.commit()  
c1.execute('UPDATE m SET name = "Rakesh"  
WHERE fees = 1900')  
<SQLite3::Cursor object at 0x02EC6EE0>  
c1.fetchall()  
[ ]
```

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

<Sqlite3::Cursor object at 0x02EC6EE0>

c1. fetchall()

[('Mahesh', 1900), ('Dheeru', 1500)]

c1. execute('Drop table m')

<Sqlite3::Cursor object at 0x02EC6EE0>

c1. close()

<Sqlite3::Connection object at 0x02EC6E0>

Step 6:- use commit method.

Step 7 :- use fetchall for displaying values.

Step 8:- using execute() drop table syntax and finally close method to be used.

Dr. NIT

# GUI Project using Tkinter

Aim :- To guess the correct word of jumbled letters

Source Code :-

```
from tkinter import *
import random
from tkinter import messagebox
answers=[  
    "python",  
    "java",  
    "swift",  
    "canada",  
    "india",  
    "america",  
    "london",  
    "delhi",  
    "germany",  
    "bhutan",  
    "russia",
```

02

## GUI Project using Tkinter

"australia",

"brazil",

"beijing",

"mexico",

"mumbai",

"tokyo",

"cairo",

"japan",

"france",

"singapore",

"britian",

"indonesia",

"italy",

"sweden",

]

words=[

"nptoyh",

# GUI Project using Tkinter

"wfsit",  
"cdanaa",  
"aidin",  
"aiearcm",  
"odnlon",  
"dlhei",  
"graeynm",  
"buahtn",  
"irsaus",  
"asuairlt",  
"rbaizl",  
"gnijieb",  
"xicome",  
"aibumm",  
"kytoo",  
"airoc",  
"ajnap",  
"nfcaer",

# GUI Project using Tkinter

```
"ingasrepo",  
"aitirb",  
"iedsanino",  
"ialyt",  
"deswen",  
]  
  
num=random.randrange(0,25,1)  
  
def res():  
  
    global words,answers,num  
  
    num=random.randrange(0,25,1)  
  
    lb1.config(text=words[num])  
  
    e1.delete(0,END)  
  
def default():  
  
    global words,answers,num  
  
    lb1.config(text=words[num])  
  
def checkans():  
  
    global words,answers,num  
  
    var=e1.get()
```

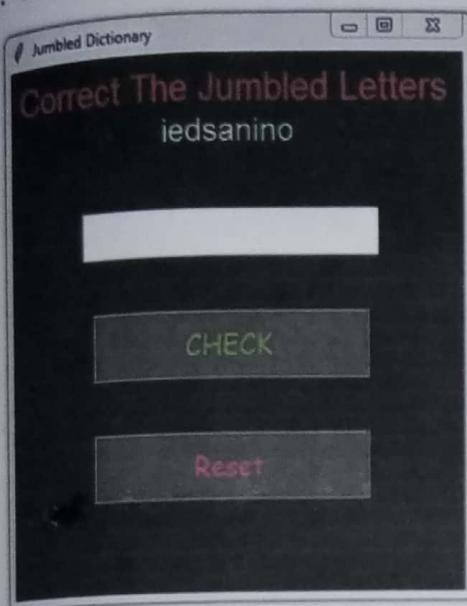
# GUI Project using Tkinter

```
if var==answers[num]:  
    messagebox.showinfo("Success","This is Correct Answer")  
  
    e1.delete(0,END)  
  
    num=random.randrange(0,25,1)  
  
    lb1.config(text=words[num])  
  
else:  
  
    messagebox.showerror("Error","This is not a Correct Answer")  
  
    messagebox.showinfo("Answer is ",answers[num])  
  
    e1.delete(0,END)  
  
    num=random.randrange(0,25,1)  
  
    lb1.config(text=words[num])  
  
root=Tk()  
  
root.geometry("365x400+400+300")  
  
root.title("Jumbled Dictionary")  
  
root.config(background="black")  
  
lb1=Label(root,text="You are Here",font=("Arial",18),bg="black",fg="white")  
  
lb1.pack(pady=30,ipady=10,ipadx=10)
```

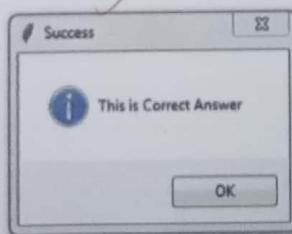
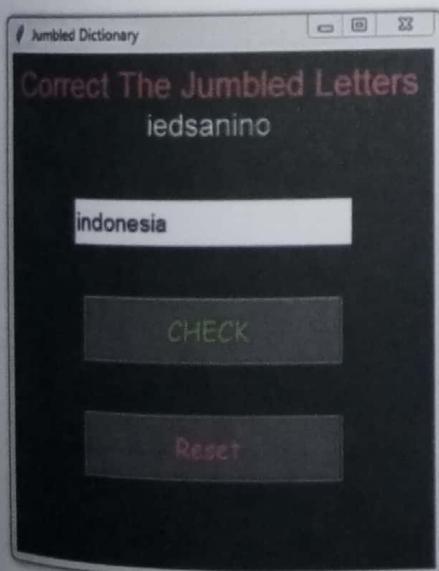
# GUI Project using Tkinter

```
lb2=Label(root,text="Correct The Jumbled Letters",font=("Arial",20),bg="black",fg="red")  
lb2.place(x=5,y=5)  
  
ans1=StringVar()  
  
e1=Entry(root,font=("New Times Roman",14),textvariable=ans1)  
  
e1.pack(ipady=5,ipadx=5)  
  
check=Button(root,text="CHECK",font=("Comic sans  
ms",16),width=16,relief=GROOVE,bg="#4C4B4B",fg="#6ab04c",  
command=checkans)  
  
check.pack(pady=40)  
  
btnreset=Button(root,text="Reset",font=("Comic sans  
ms",16),width=16,bg="#4C4B4B",fg="#EA425C",relief=GROOVE,  
command=res)  
  
btnreset.pack()  
  
default()  
  
root.mainloop()
```

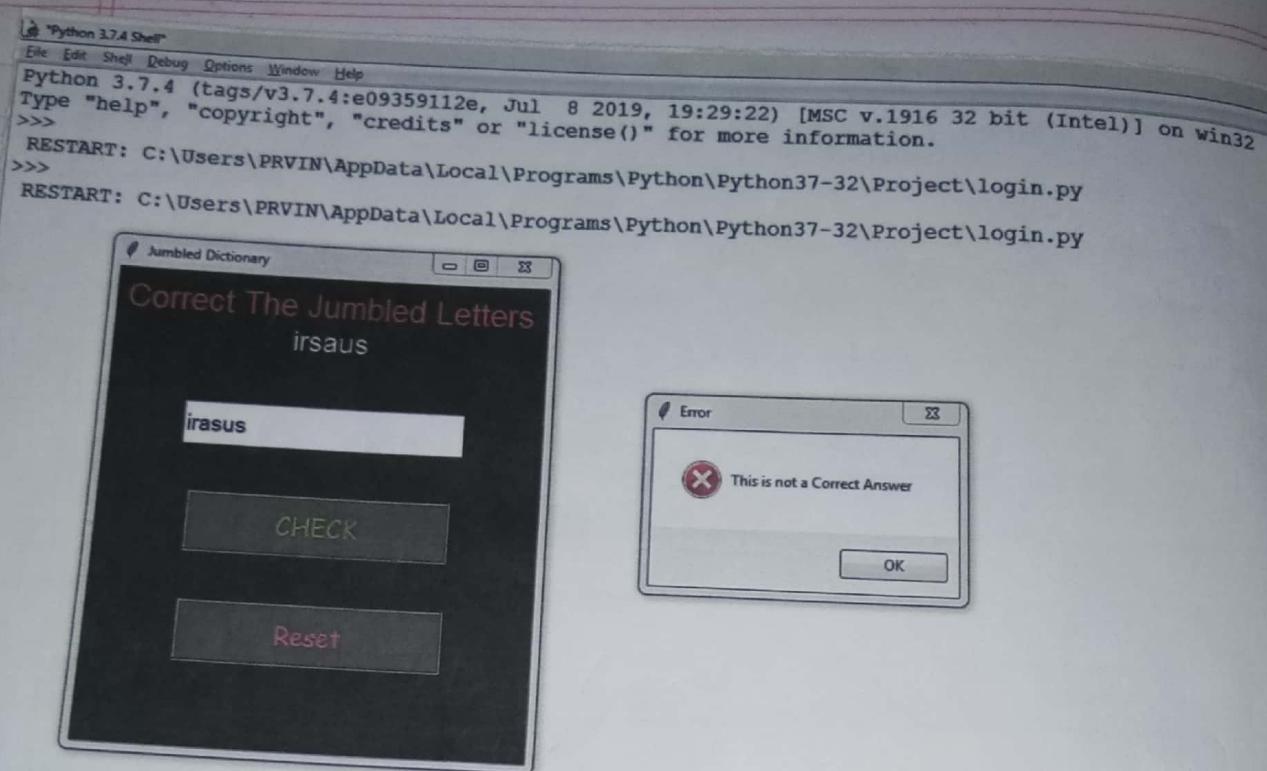
```
Python 3.7.4 Shell
File Edit Debug Options Window Help
Python 3.7.4 (tags/v3.7.4:e09359112e, Jul  8 2019, 19:29:22) [MSC v.1916 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> RESTART: C:\Users\PRVIN\AppData\Local\Programs\Python\Python37-32\Project\login.py
>>> RESTART: C:\Users\PRVIN\AppData\Local\Programs\Python\Python37-32\Project\login.py
```



```
Python 3.7.4 Shell
File Edit 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\PRVIN\AppData\Local\Programs\Python\Python37-32\Project\login.py
>>> RESTART: C:\Users\PRVIN\AppData\Local\Programs\Python\Python37-32\Project\login.py
```



Dr. / 087



"Python 3.7.4 Shell"

File Edit Shell Debug Options Window Help

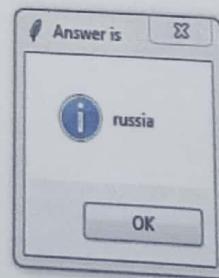
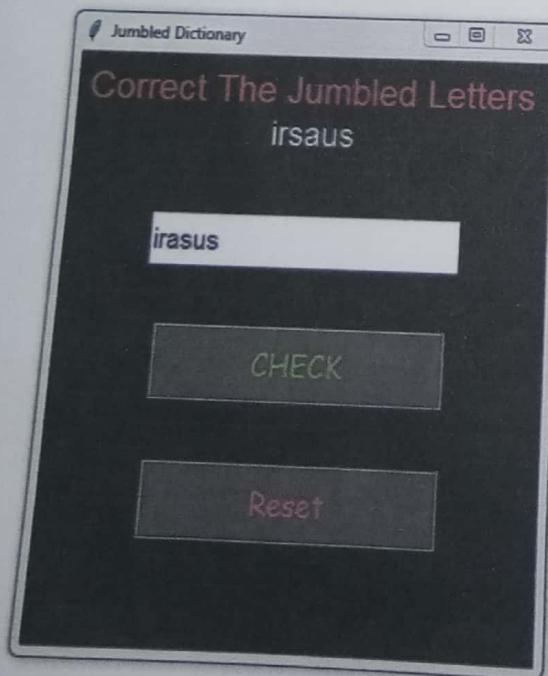
Python 3.7.4 (tags/v3.7.4:e09359112e, Jul 8 2019, 19:29:22) [MSC v.1916 32 bit (Intel)] on win32

Type "help", "copyright", "credits" or "license()" for more information.

>>>

RESTART: C:\Users\PRVIN\AppData\Local\Programs\Python\Python37-32\Project\login.py

RESTART: C:\Users\PRVIN\AppData\Local\Programs\Python\Python37-32\Project\login.py



13

# Database using Python and Tkinter

- □ X

Name :

Phone :

update Name :

Provide Phone No. :

Delete :

## TKinter Project using sqlite3

Source Code:

```

from tkinter import *
import sqlite3

root = Tk()
root.geometry('410x450')
root.title("Database using Sqlite3 and Tkinter")
root.config(background="powder blue")

textin = StringVar()
textin.set("Enter your Query here")

textin2 = StringVar()
textin2.set("Result will appear here")

menu = Menu(root)
root.config(menu=menu)

def help():
    help(sqlite3)

subm = Menu(menu)
menu.add_cascade(label="Help", menu=subm)
subm.add_command(label="Sqlite3 Docs", command=help)

```

`db = sqlite3.connect('myeq.db')`  
`cursor = db.cursor()`  
`cursor.execute("CREATE TABLE IF NOT EXISTS people(name TEXT, phone TEXT)")`  
`db.commit()`

`lab = Label(root, text='Name:', font=('none 13 bold'))`  
`lab.place(x=0, y=0)`

`entname = Entry(root, width=20, font=('none 13 bold'), textvar=textin)`  
`entname.place(x=80, y=0)`

`entphone = Entry(root, width=20, textvar=textin)`  
`entphone.place(x=80, y=40)`

`lab1 = Label(root, text='Phone:', font=('none 13 bold'))`  
`lab1.place(x=0, y=40)`

def insert():

name1 = textin.get()

phone1 = textin.get()

conn = Sqlite3.connect('mysql.db')

with conn:

cursor = conn.cursor()

cursor.execute('Insert into people

(name, phone) VALUES(?, ?)',

(name1, phone1,))

db.close()

def show():

conn = Sqlite3.connect('mysql.db')

cursor = conn.cursor()

cursor.execute('Select \* from people')

for row in cursor.fetchall():

print(row)

name = StringVar()

phone = StringVar()

def updateContact():

nam = name.get()

ph = phone.get()

conn = sqlite3.connect('myeq.db')  
cursor = conn.cursor()  
cursor.execute("update people  
set name = ? where phone = ?"  
(name, ph))  
conn.commit()

dell = StringVar()

def del():  
 dee = dell.get()  
 conn = sqlite3.connect('myeq.db')  
 cursor = conn.cursor()  
 cursor.execute("Delete from  
 people where name = ?" , (dee,))  
 conn.commit()

def drop():

conn = sqlite3.connect('myeq.db')  
 cursor = conn.cursor()  
 cursor.execute("Drop Table people")  
 conn.commit()

~~root.mainloop()~~

bbutton = Button(root, text='Drop table',  
 command=drop),  
 bbutton.place(x=180, y=380)

bupdate = Button(root, text='Update', command=  
 update(contact),  
 bupdate.place(x=80, y=280)

labelname = Label(root, text='Update Name :')  
 labelname.place(x=0, y=200)

ename = Entry(root, textvar=name)  
 ename.place(x=160, y=200)

lphone = Label(root, text='Provide Phone No.')  
 lphone.place(x=0, y=240)

update = Entry(root, textvar=phone)  
 update.place(x=210, y=240)

delete = Label(root, text='Delete')  
 delete.place(x=0, y=340)

edele = Entry(root, textvar=dell)  
 edele.place(x=90, y=340)

23

butdel = Button (root, command=del)  
butdel.place (x=90, y=380)

but = Button (root, text='Submit',  
command=insert)  
but.place (x=60, y=100)

ses = Button (root, text='Show',  
command=show)  
ses.place (x=160, y=100)

root = mainloop()

27/09