```
file obj = open ("abc.txt", "w").
file obj = write (" computer science subjects" + "\n")
file obj = write ("DBMS in python in OS \n")
file obj .close ()

fileobject = open ("abc.txt", "a")
# read ()
str 1 = file obj .read ().
print (" the output of read method :", str 1)
fileobj . close ().
>>> (' The outputs of read method :', 'computers science
                in DBMS in python in OS\n') subjects

# read line ()
file obj = open ("abc . txt", "r")
str 2 = file obj . readline ()
print ("the output of readline method :", str 2)
file obj . close ()
>>> ('The output of readline method :', 'computer science
      subject \n')

# readlines ()
file obj = open ("abc . txt", "r")
str 3 = file obj . readlines ()
print ("the output of readlines method :", str 3)
file obj . close ()
>>> ('The output of readlines method :', ['computer science
      subject \n', 'DBMS \n', 'python \n', 'os \n'])
```

Practical No. 1

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 contents onto the file and than closing the file.

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

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

Step 04: Now open the file object in write mode write some another content close subsequently then again open the file object in 'w+' mode that is the update mode and write contents.

Step 05: open file object in read mode, display the update written contents and close open again in 'r+' mode with parameter passed and display the output subsequently.

Step 6: Now open file object in append mode open write method write content close the file object again open the file object in read mode and display the append output.

```
# file attributes.
a = fileobj.name
print ("name of file (name attribute):", a)
>>> ('name of file (name attribute):', 'abc.txt')
b = file obj.closed.
print ("(close) attribute :", b)
>>> ('(close) attribute =', 'True')
c = fileobje.mode
print ("file mode", c)
>>> ('file mode', 'r')
d = fileobj.softspace
print ('softspace", d)
>>> ('softspace:", 0)
```

```
# wt mode
fileobj = open ("abc.txt", "wt")
fileobj.write ("saurabh")
file obj.close()
```

```
# write mode.
fileobj = open ("abc.txt", "write")
fileobj = write ("DBMS")
file obj.close().
```

```
# r+ mode.
file obj = open ("abc.txt", "r+")
str1 = fileobj.read (6)
print ("output of r+", str1)
file obj.close().
>>> ('output of r+', 'saurabh')
```

```
# read mode.
file.obj = open ("abc.txt", "r")
str 2 = fileobj.read()
print ("output of read mode!",
        str 2)
>>> ('output of read mode !',
    'saurabh')
```

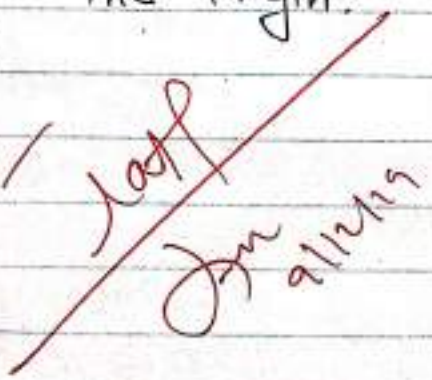
P9:

```
# append mode.
file obj = open ("abc.txt", "a")
file obj.write ("data structures")
fileobj.close()
fileobje = open ("abc.txt", "r")
str3 = fileobje.read()
print ("output of append mode:", str3)
fileobj.close().
>>> ('output of append mode:', 'sameabh', 'data structu')

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

# seek()
fileobj = open ("abc.txt", "r")
str4 = 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 = file obj.readlines()
print ("output:" str9)
for line in str9:
        print(len(line))
    fileobj.close()
>>> ("output:', ['ollege database'])
```

Step 07: open the fileobject in read mode. declare. a variable and perform fileobject dot tell method and store the output consequently in variable.

Step 08 : use the seek method with the arguements with opening fileobject read mode ~~dot tell method and store the closing~~ subsequently.

Step 09: open fileobject 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.



Program

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

Output

```
Neeraj
Raj
Pranay
Sachin
```

Program:

```
mytuple = ("Neeraj", "Raj", "Pranay", "Sachin")
for a in mytuple
    print(a)
```

output

```
Neeraj
Raj
Pranay
Sachin
```

Practical No. 2.

Aim: To display elements of a tuple using iterator method.

① Algorithm:

Step1: form a tuple with certain elements inserted in it.

Step2: Use iter method with tuple and assign it to a variable.

Step3: Use the next method with variable and print the elements.

Aim: To use iter method with for loop

② Algorithm:

Step1: form a tuple with certain elements inserted in it.

Step2: use the for conditional statement to access each element of tuple.

Step3: print the element of tuple.

③ progeam using the iterable method for display the set of odd numbers.

Step 1: Define a class and within that define the iter() which will initialize the first element within the container object.

step 2: Now use the next() and define the logic for collecting the odd values.

step3: Defines an object for the class and iter through the Object.

Step 4: Now display the values using the for conditional statement.

Source code

```
class odd:
    def __iter__(self):
        self.num = 1;
        return self;
    def __next__(self):
        if self.num < 100:
            num = self.num
            self.num += 2
            return num
        else:
            raise StopIteration.
myobj = odd()
myiter = iter(myobj)
for num in myiter.
    print(num).
```

Output:

```
1
3
5
7
9
11
13
15
17
19
21
23
25
27
29
31
33
35
37
39
41
43
45
.
.
.
99.
```

```
class myclass :
    def __iter__(self)
        self.a = 1
        return self
    def __next__(self):
        if self.a <= 20:
            x = self.a
            self.a += 1
            return x
        else:
            raise StopIteration.
myobj = myclass()
myiter = iter(myobj)
for x in myiter.
    print(x).
```

output

```
1
2
3
5
6
7
9
1
3
5
7
9
2
```

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

(4) Progeam using the iterable object to display set of first 20 number.

Step 1 :- Define iter( ) with an argument and initialize it to the first value.

Step 2 :- for extracting the next element from the container, use the next( ) with an argument and compare the no. of elements required in a container by using the conditional statement.

Step 3 :- Now create an object from the given class and pass this object as an argument to the iter( ) method.

Step 4 :- Now using the conditional statement display all the values from the given container.

⑤ program for printing the square and cube of given set of number using the map ().

Step 1: Define a square function with an arguement and return the square of the nomber.

Step 2: Define a function cube with an arguement and return the cube of the number.

Step 3: Declare a list variable and call the functions square and cube in the list

Step 4: Use the conditional statement and use the map() to find the square and cube given set of Numbers.

source code:

```
def square (x):
    return (x**2)
def cub (x):
    return (x**3)
func1 = [square, cube]
for r in range (5):
    Value out = list (map (lambda x: x(r), func1))
    print (Value out)
```

>>> output

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

08

Source code:

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

Output:

10

16/12/19

⑥ T:

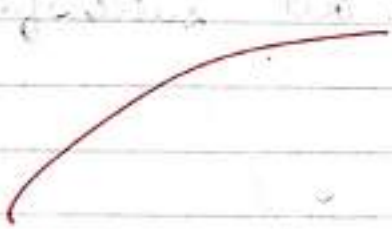
Step 1: open the file in read mode

Step 2: use readlines 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.



Practical No. 3.

* write a progeam using the exception block related to the environment error.

step 1: use the try block to define the normal force of action
for eg: Define file object & open the file in write mode & write some content into the file.

step 2: use the except block with the I/O environment error & convey the appropriate message to the user, else display the message that the operation is carried out successfully.

Source code

```
try:
    fileobj = open('abc.txt", "w")
    file obj . write("python is an indented language")
except IOError:
        print ("there is an environment error")
else: print ("operation successfull")
```

output

>>> operation successfull.

② Source code:

```
try:
    a = int(input("Enter the number"))
except. Value Error:
    print ("Arithmatic error")
else:
    print ("successfull")
try:
    Fo = open ("abc1.txt", "w")
    Fo.write ("hello, these how are you ??")
except Error:
    print ("Environmental error")
else:
    print ("successful")
```

Output:

```
Enter the number : 14
Arithmatic error:
```

* write a program to demonstrating the use of value error in the given program statement

Step 1: Accept the value from the use & if it is a valid value display the enter value & terminate the condition by using break statement.

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

Step 3: we can define the multiple exception using the except statement for finding the different category of errors.

#8E

Practical No. 4.

Topic : Regular expression.

Step 1 : Import re module declare pattern and declare sequence use match method with declare arguments if arguments matched then print the same otherwise print pattern Not found!

Step 2 : Import re module declare pattern with literal and into character declare string value - use the findall() with argument and print the same

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

Step 4 : import re module declare string and accordingly declare pattern replace the blank space with no_space use sub() with 3 argument and print the string without spaces

```
# match()
import re
pattern = r"FYCS"
sequence = "FYCS represents computer science stream"
if re.match(pattern, sequence):
    print("matched pattern found!")
else:
    print("Not FOUND")
```

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

```
# numerical values (segregation)
import re.
pattern = r'\d+'
string = 'hello123, howdy789, 45howru'
output = re.findall(pattern, string)
print(output)
```

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

```
# split()
import re
pattern = r'\d+'
string = 'hello123, howdy789, 45howru'
output = re.split(pattern, string)
print(output)
```

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

```
# 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 ('\Apython', Sequence)
print(V)
V1 = V.group()
print (V1)
```

```
>>> <_sre.SRE_Match object at 0x028FDF00>
    python.
```

verifying the given set of phone Nomber.

```
import re
list1 = ['8004567891', '9145673210', '7865432981',
         '9876543201']
for value in list1:
    if re.match (r'[8-9]{1}[0-9]{9}',
    value or len (value) == 10):
    print ("criteria matched for all number!")
    else:
      print("criteria failed!")
```

```
>>> criteria matched for all number
    criteria match for all number.
    criteria falical!
    criteria matched for all number.
```

valeds.

```
import re
str1 = 'plant is life overall'
output = re.findall (r'\b[aeiouAEIOU]\w+', str1)
print (output)
```

```
>>> ['is', 'overall']
```

Step 5: import re module. declare a sequence. use search method for finding subsequently use the groups() with dot operator ds search() given 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 ~~the~~ for condition statement. use if condition for checking first number is either 8 or 9 and next number are in range of 0 to 9 and check whether the entered number are equal to 10 if criteria matches. print ok number matches otherwise print failed

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

~~Step~~:

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

step 9: import re module enter a string use pattern to display only two elements of the particular string use findall(). declare two variables with initial value as zero use for condition and subsequently use the if conditions check whether condition satisfy add up the or else increment value and display the values subsequently.

host of domain

```
import re
seq = 'abc.tcsc@edu.com, xyz@gmail.com'
pattern = r'[\w\.-]+[\w\.-]'
output = re.findall(pattern, seq)
print(output)
```

```
>>> ['abc.tcsc', 'edu.com', 'xyz', 'gmail.com']
```

counting of first 2 letters.

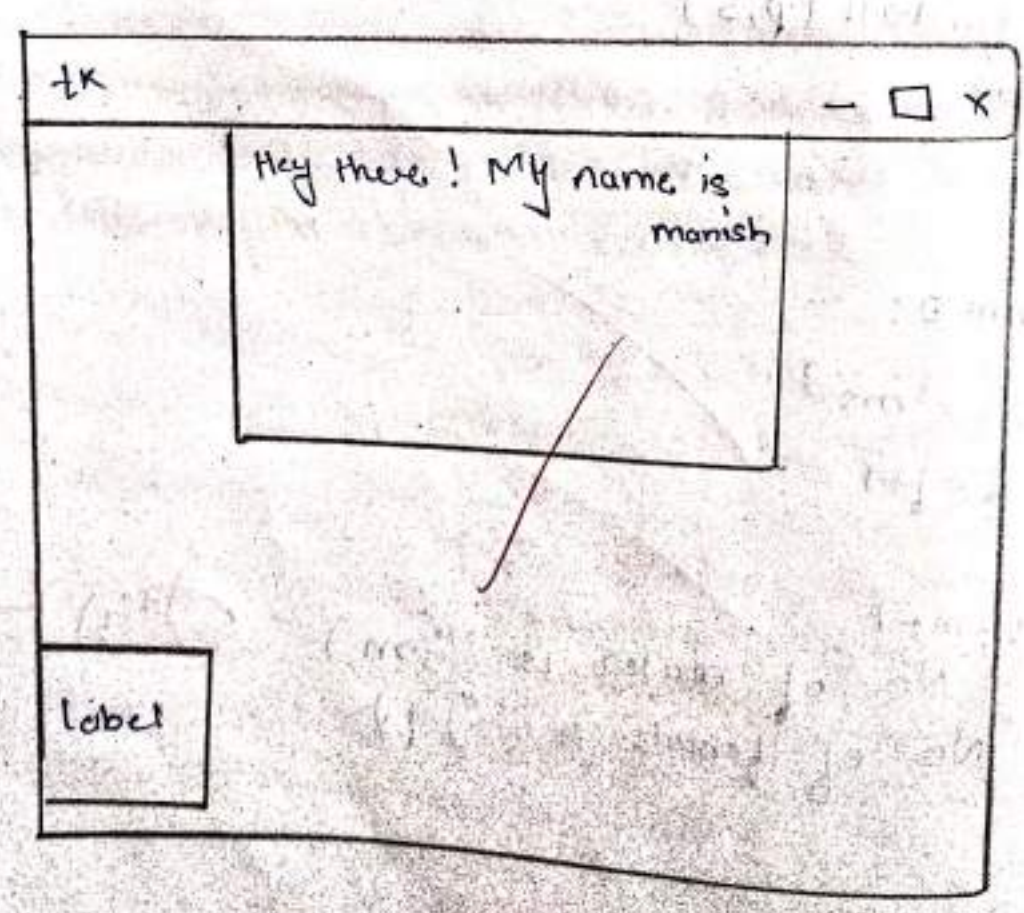
```
import re
s = 'mr.a, ms.b, ms.c, mr.t'
p = r'[/ms|mr/]+
o = re.findall(p,s)
print(o)
m=0
f=0
for v in o:
if(v=='ms'):
    f=f+1
else
    m=m+1
print("No. of males is :", m)
print("No of femals is :", f)
```

```
>>>
```

Program:

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

Output



Practical No. 5.

Aim: To make use of GUI application along with the basic pack method

Algorithm:

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

Step 3: Use the pack() along with the object create from text method & use the parameter.
1) side = Top, padx = 30, ipadx = 40, ipady = 50

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

Step 5: Now export above step with a label method which takes the following argument.
1) Name of parent window
2) Text attribute which defines the string
3) The background colour (bg)
4) The foreground colour (fg)
5) Now use pack() with relevent attribute

Aim: To make use of Radio Button widget for selection of one of the multiple option.

Algorithm:

Step1: Use the tkinter method to import the relevant method.

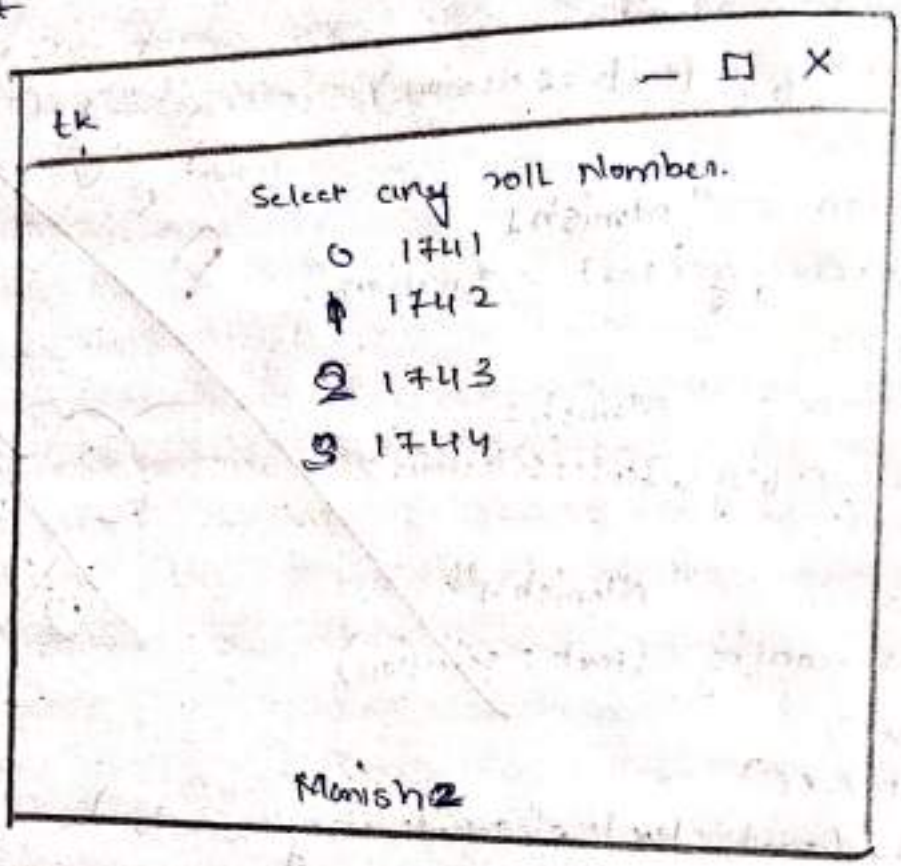
Step 2: Define a function which tells user about given selection mode from multiple option available.

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

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

Step 5: Now create object of Radio Button which will take following arguments
1) positioning on Parent window.
2) Text variable
3) Define Variables argument
4) Corresponding value and trigger the given function.

output



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 in onto parent window using pack method & falling use the main loop method.

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

Algorithm:

Step1 : Import tkinter library to use Scroll Bar widget.

Step 2 : Create an object corresponding to Scroll parent window & create an object from Scroll bar & place it on the parent window so created.

Step3 : Create an object of label method to provide a heading and place it on parent window.

Step4 : Use pack method along with object of Scrollbar method & use argument side & fill.

Step5 : Create an object of listbox method and place it onto parent window with attribute yscroll command.

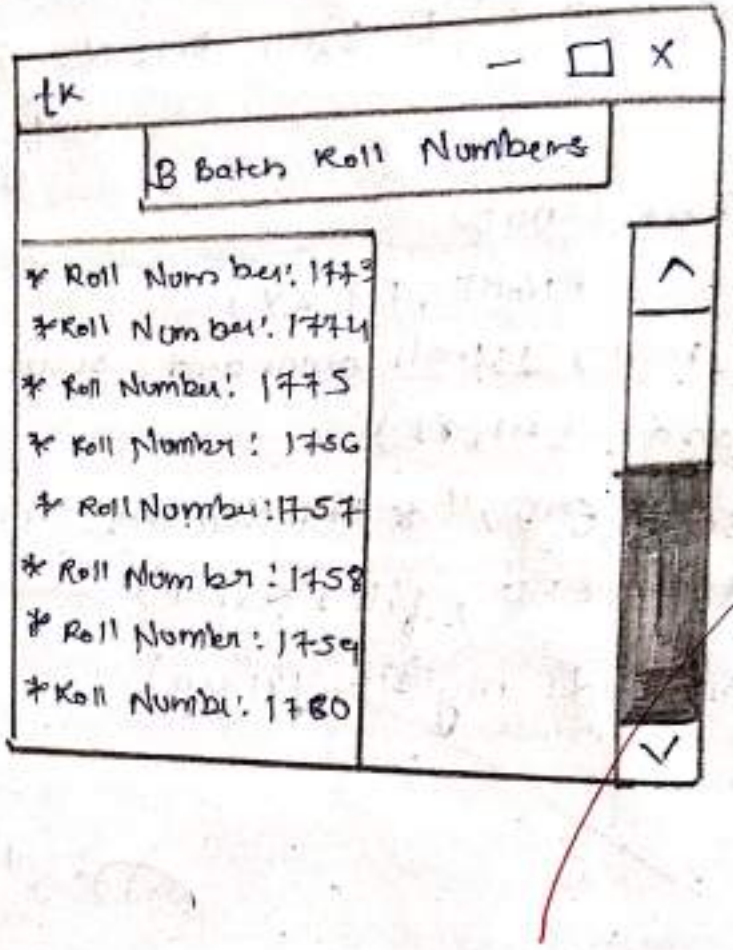
Step 6 : use for loop to insert values in the object of list box by using insert method

Program:

```
from tkinter import *
root = Tk()
root.geometry ('450x400')
L=.label (root, text = " B Batch Roll Numbers : "bg="black",
                                                bg = "white")
1.pack()
scroll = scrollbar (root)
scroll.pack (side = RIGHT, Fill =Y)
mylist = listbox (root, yscroll command = scroll.set, bg = "light
                                                blue")
for num in range (41, 81):
    mylist.insert (END, " * Roll Number : 17", str (num))
mylist.pack (side = LEFT, fill : BOTH)
scroll.config (command = mylist.yview)
root.mainloop()
```

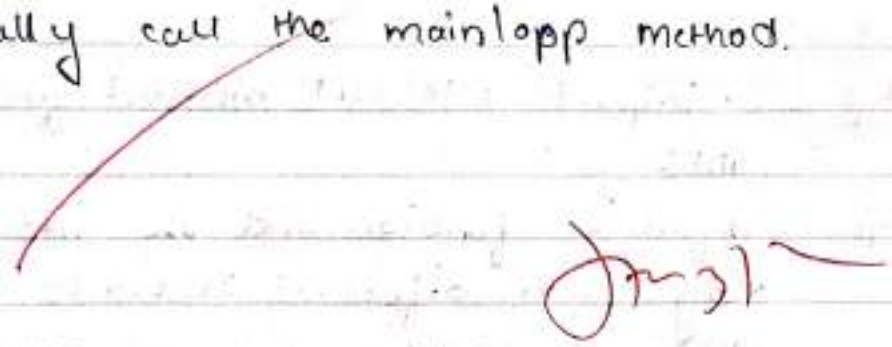
11

Output


tk
B Batch Roll Numbers
* Roll Number: 1773
*Roll Number: 1774
* Roll Number: 1775
* Roll Number: 1756
* Roll Number: 1757
* Roll Number: 1758
* Roll Number: 1759
*Roll Number: 1760

Step 7: Use config method along with scroll bar object & use common & attribute.

Step 8: finally call the mainloop method.



Aim: To make use of messagebox 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 continues one or more argument.

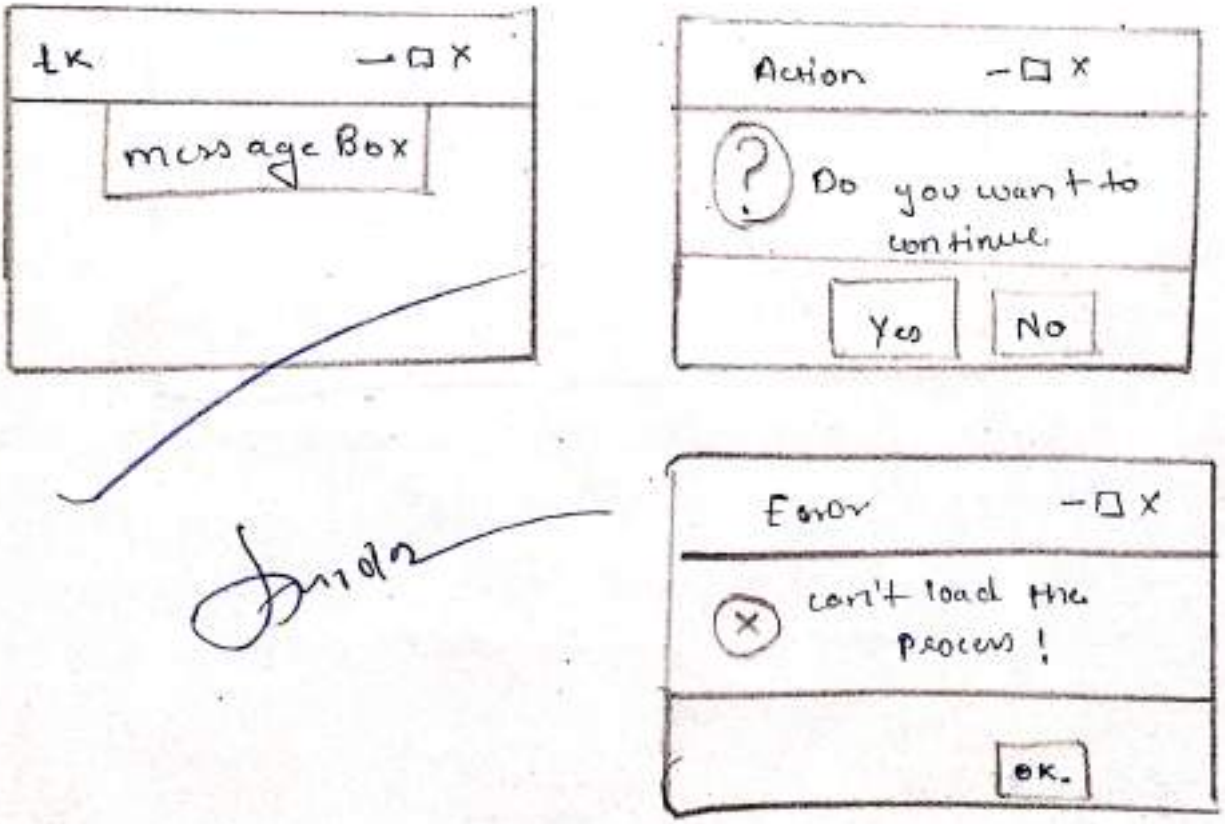
step 3: Create an object from button method and places it onto the parent window with text and command attribute specified.

step 4: Use pack method and finally use the mainloop method.

Source code:

```
From tkinter import *
from tkinter import messagebox
def msgbox():
    messagebox.~~showw~~ askyesorno("Action", "Do you want to
    messagBox.showerror("Error", "can't load the
root=Tk()                                    process!")
root.config(bg="grey")
Bl= Button(root, text="message Box", bg:"Blue", command
                                              =msgbox)
Bl.pack().
root.mainloop()
```

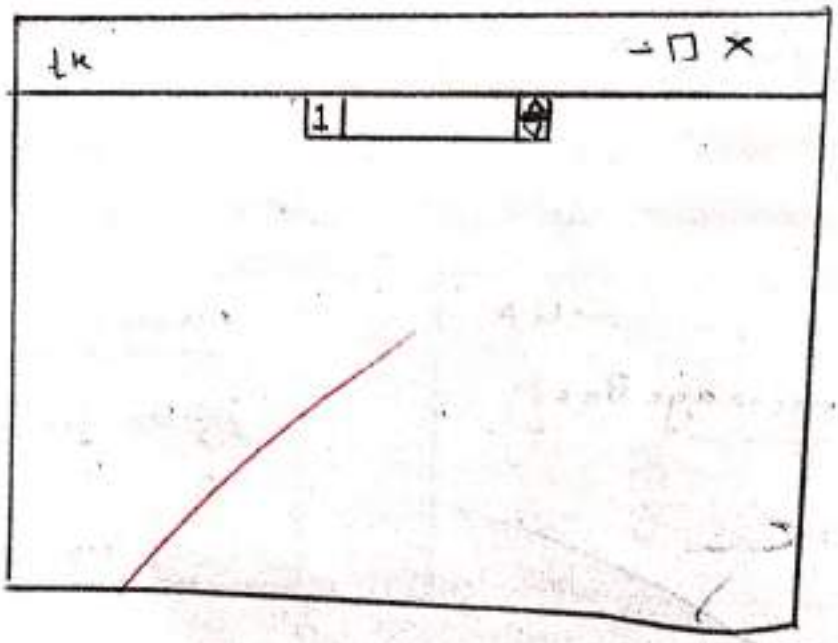
output:



Source code:

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

output



Aim: To make the use of spinbox method of the GUI application.

Algorithm:

Step 1: Import relevant method from tkinter library.

Step 2: Create a object from the tk() & subsequently create an object from the spinbox().

Step 3: make the object so created on to the parent window and trigger the corresponding event.

Step 4: use pack method and finally use the mainloop method.

Aim: To make the use of panedwindow method of the GUI application.

Algorithm:

Step 1: Import relevant method from tkinter library.

Step 2: Create an object from panedwindow method use the pack with. ~~att~~ method with attribute fill & expand.

Step 3: Create one object from the label method and put it on to the panedwindow with text attribute and use the add method to implement the new object.

Step 4: Similarly create a second Panedwindow object and add on to the first panedwindow with orientation specified.

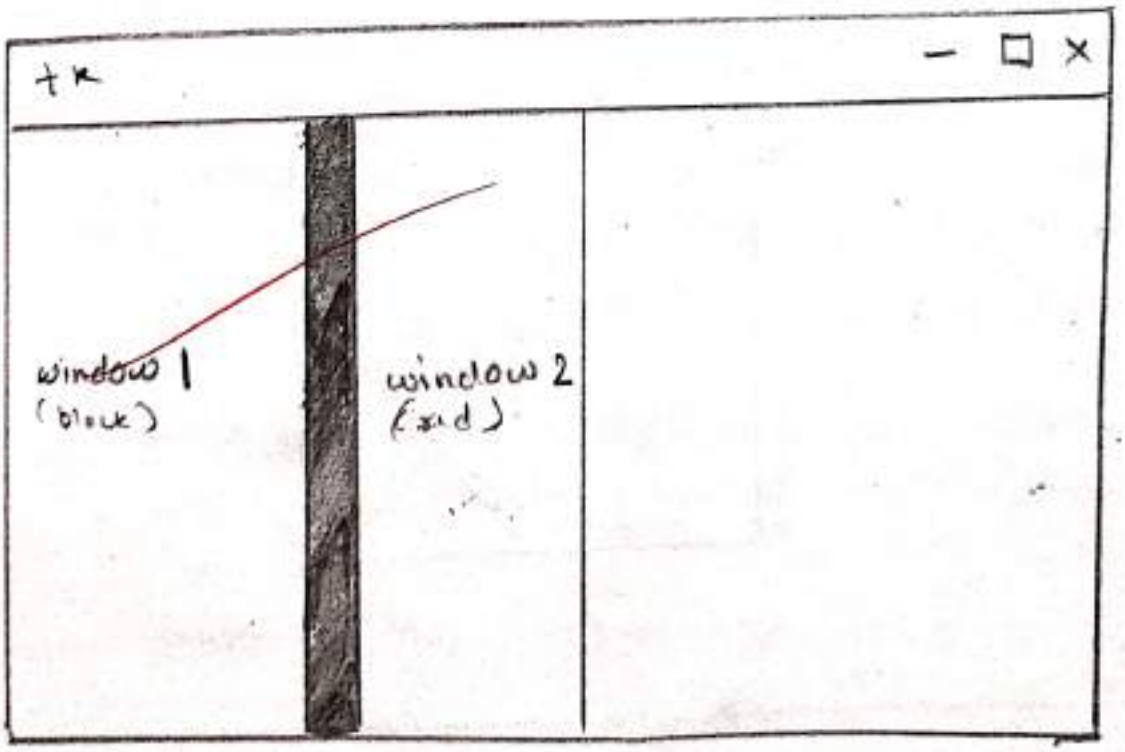
Step 5: Create a label object & placed in the second panedwindow and add on to the second panedwindow.

Step 6: finally use the mainloop method.

Source code:

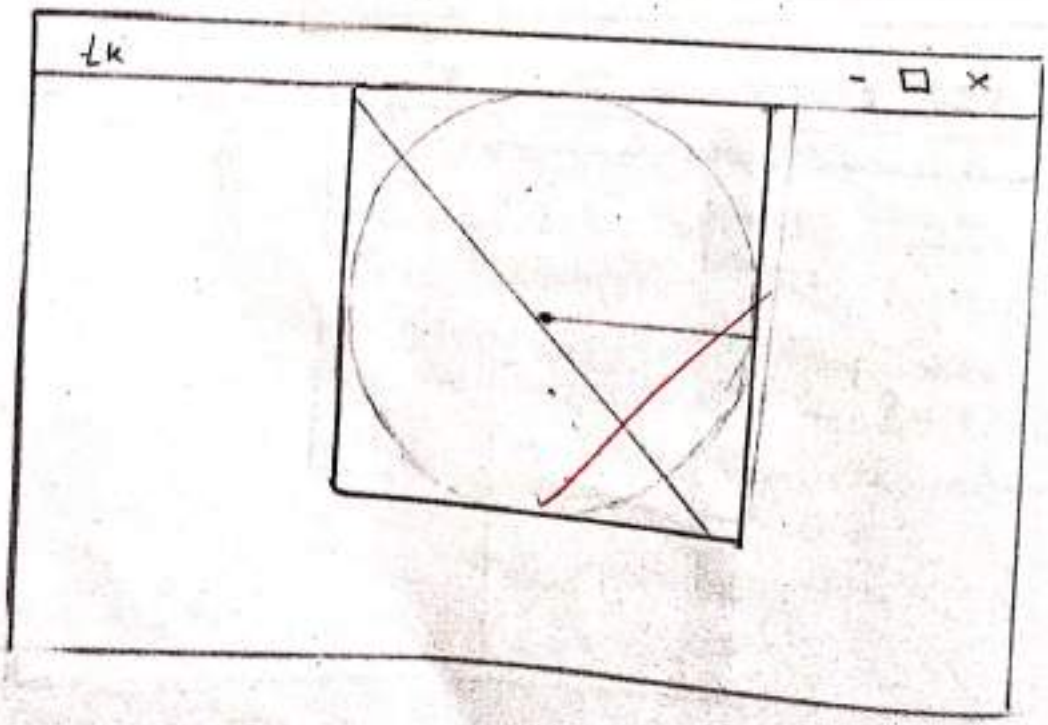
```
From tkinter import *
root = Tk()
p = PanedWindow()
p.Pack(fill = BOTH, expand = 10)
L = Label(p, text = "window 1", bg = "black", bg = "white")
p.add(L)
P1 = PonedWindow(orient = VERTICAL)
p.add(L1)
L1 = Label(p, text = "window 2", bg = "red")
p.add(L1)
P2 = PanedWindow(orient = HOIVERTICAL)
p.add(P2)
root.mainloop()
```

output



```
from tkinter import *

root = Tk()
c = Canvas(root, height = 200, width = 300, bg = "white")
coord = (3, 3, 300, 300)
arc = c.create_arc(coord, start = 0, extent = 359,
                   fill = "green")
c.pack()
coord = (100, 500, 500, 100)
line = c.create_line(coord, fill = "yellow")
c.pack()
root.mainloop()
```

output.



Aim: To make use of canvas method of the GUI Application.

Algorithm:

Step 1: import relevant method from tkinter Library.

Step 2: use the canvas method to create line, create oval, create arc along with the canvas object so create and use the co-ordinate values.

Step 3: Similary use other method and call the pack method and the mainloop method.

SEM Practical No. 6.

To implement the data base.

dbm.

Algorithm.

1. import the dbm library & use the open method for creating the database by specifying the name of the database along with the corresponding flag.
2. Assume three string values c, N & w where c & N are used for creating a new database & while w flag is used for writing data to the existing database
3. use the object so created for accessing the given website and for the corresponding & regular value for the website.
4. check whether the given URL address with the regular values of the page is or not equal to value then print the URL found else print not found

dbm.

```
import dbm
db = dbm.open ("database", 'c')
if db["https://www.google.com"] = "database"
```

```
import dbm
db = dbm.open ("database", Flag="c")
db["http://www.google.com"] = "database"
if db["http://www.google.com"] != None:
    print ("URL is found")
else:
    print ("URL not found")
db.close()
```

output:

>>> URL is found

code:

```
import dbm, os, sqlite3
connection = sqlite3.connect("students.db")
cursor1 = connection.cursor()

cursor1.execute('create table stud (name CHAR, rollno
<sqlite3.cursor object at 0X004104A0>
cursor1.execute('insert into stud values ("Manish",
<sqlite3.cursor object at 0X004107A0>
cursor1.execute('insert into stud values ("shubhang",
<sqlite3.cursor object at 0X004107A0>
connection.commit()

cursor1.execute('select * from stud')
<sqlite3.cursor object at 0X004D7A0>
cursor1.execute('select Name from where RND=12:
<sqlite3.cursor object at 0X004107A0>
cursor1.fetchall()
```

output:

```
[('shubhang',)]
```

database collectivity.

step 1 : input the corresponding libraries for making the data base. connection which are os & the sqlite 3.

step 2 : Now create the connection object using sqlite 3 library & the connect method for creating the new db

step 3: Now create the cursor object using the curser method from the connection object created in step ②

step 4: Now use the create() for creating the table with the colomns name & respective data type.

step 5: Now with cursor object use the insert statement for entering the values corresponding to different field considering the data type

step 6: use the commint() to complete the transaction using connection object

step 7: use the execute statement along with the cursor object for accessing the values from the data base using the select from clause.

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

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

Project

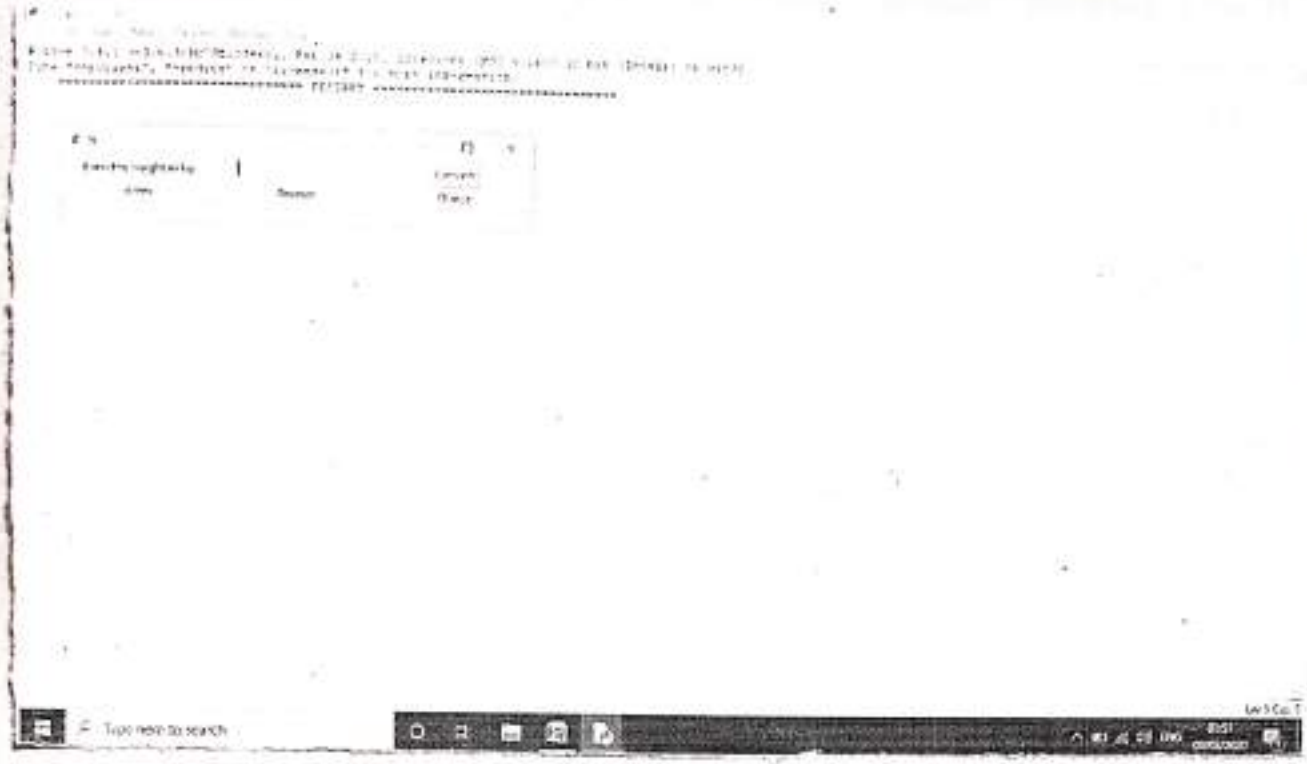
→ To convert weight from kg to pounds, gram, ounces.

source code.

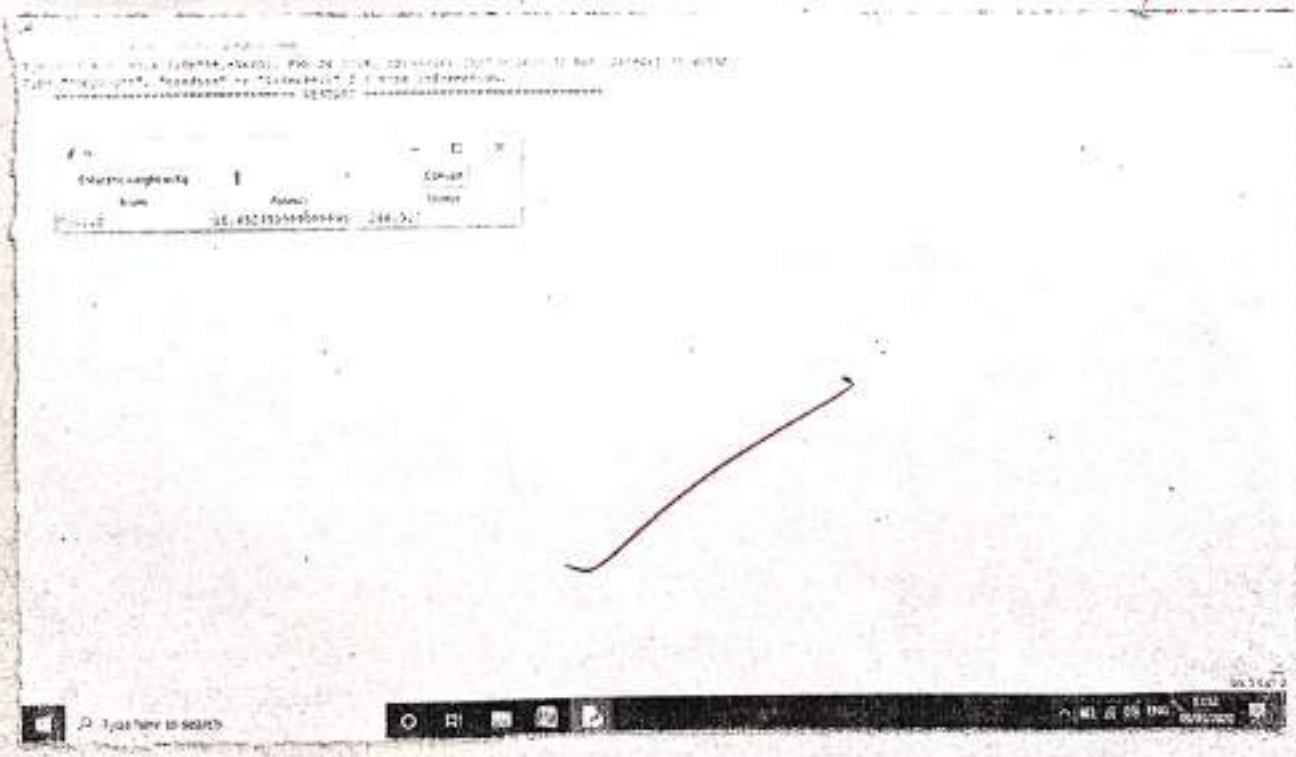
```
from tkinter import *
window = Tk()
def from_kg():
    gram = float(e2_value.get())*1000
    pound = float(e2_value.get())* 2.20462
    ounce = float(e2_value.get())* 35.274
    t1.delete("1.0", END)
    t1.insert(END, gram)
    t2.delete("1.0", END)
    t2.insert(END, pound)
    t3.delete("1.0", END)
    t3.insert(END, ounce)
~~et = tabel~~
e1 = label(window, text = "Enter the weight in kg")
e2_value = StringVar()
e2 = Entry(window, textvariable = e2_value)
e3 = label(window, text = 'gram')
e4 = Label(window, text = 'pounds')
e5 = Label(window, text = 'Ounce')
```

```
t1 = Text (window, height = 1, width = 20)
t2 = Text (window, height = 1, width = 20)
t3 = Text (window, height = 1, width = 20)

b1 = Button (window, text = "convert", command =
                                        from_kg)

e1.grid (row = 0, column = 0)
e2.grid (row = 0, column = 1)
e3.grid (row = 1, column = 0)
e4.grid (row = 1, column = 1)
e5.grid (row = 1, column = 2)
t1.grid (row = 2, column = 0)
t2.grid (row = 2, column = 1)
t3.grid (row = 2, column = 1)
b1.grid (row = 0, column = 2)

window.mainloop()
```

→ Before entering value.



→ After entering value.



Database project:

```
import dbm, os, sqlite3
connection = sqlite.connect ("fruits.db")
c1 = connection.cursor()
c1.execute ('create table fruits (Name, price, weight)')
<sqlite3.Cursor object at 0x0302DF20>

c1.execute ('insert into fruits values ("Apple", 100, 1kg)')
<sqlite3.Cursor object at 0x03D20F20>

c1.execute ('insert into fruits values ("mango", 120, 1kg)')
<sqlite3.Cursor object at 0x06D20F20>

c1.execute ('insert into fruits values ("banana", 40, 1
<sqlite3.Cursor object at 0x03D20F20> dozen)')

c1.execute ('insert into fruits values ("chikoo", 60, 1kg)')
<sqlite3.Cursor object at 0x03D20F20>

connection.commit()
```

```
c1.execute ('select Price from fruits where Name=
<sqlite3.Cursor object at 0x02E6D65E0> Apple)')
```