

```

for formalRun Opens Window Help
distance field.setRoot(0, 0xE)
duration = "0:00:00";
root.configure(bg="light green")
root.geometry("510x300")
headline = Label(root, text = "Welcome to distance time calculator",
                fg = "black", bg = "red")
label1 = Label(root, text = "Source",
                fg = "black", bg = "dark green")
label2 = Label(root, text = "Destination",
                fg = "black", bg = "dark green")
label3 = Label(root, text = "Distance between nodes",
                fg = "black", bg = "red")
label4 = Label(root, text = "Duration",
                fg = "black", bg = "dark green")
label5 = Label(root, text = "Duration",
                fg = "black", bg = "dark green")
headline.grid(row = 0, column = 1)
label1.grid(row = 1, column = 0, sticky="EW")
label2.grid(row = 2, column = 0, sticky="EW")
label3.grid(row = 3, column = 0, sticky="EW")
label4.grid(row = 4, column = 0, sticky="EW")
sourceField = Entry(root)
sourceField.grid(row = 1, column = 1)
destinationField = Entry(root)
destinationField.grid(row = 1, column = 1, ipadx = 100)
durationField = Entry(root)
durationField.grid(row = 2, column = 1, ipadx = 100)
distanceField = Entry(root)
distanceField.grid(row = 3, column = 1, ipadx = 100)
durationField = Entry(root)
durationField.grid(row = 4, column = 1, ipadx = 100)
button1 = Button(root, text = "CLEAR", bg = "red",
                 command = del_source)
button2 = Button(root, text = "CLEAR", bg = "red",
                 command = del_destination)
button3 = Button(root, text = "RESET",
                 bg = "red", fg = "black",
                 command = f1)
button4 = Button(root, text = "CLEAR ALL",
                 bg = "red", fg = "black",
                 command = del_all)
button5 = Button(root, text = "Train", command = train)
button6 = Button(root, text = "Driving", command = driving)

```

```

button1 = Button(root, text = "walking", command = walking)
button2 = Button(root, text = "driving", command = driving)
button3 = Button(root, text = "TRAIN",
                 fg = "black", bg = "red",
                 command = del_train)
button4 = Button(root, text = "DRIVING",
                 fg = "black", bg = "red",
                 command = del_driving)
button5 = Button(root, text = "CLEAR", command = f1)
button6 = Button(root, text = "CLEAR", command = f1)
root.mainloop()

```

24

welcome to distance time calculator

Dehradun

CLEAR

Delhi

CLEAR

Choose travelling modes:

Train

Driving

Walking

train

CLEAR

RESULT

326 km

6 hours 58 mins

CLEAR ALL

040

Step 8 Finally use the fetch 0 or fetchall() for displaying the values from the table using the cursor object.

Step 9 execute C \$ drop table syntax for terminal the datatype & finally write close().

048

/ Jan 12

#=2 FAD

```
import os.sqlite3  
conn = sqlite3.connect("employee.db")  
cur = conn.cursor()  
cur.execute("create table emp (Name char, RollNo int)")  
cur.execute("Insert into emp values ('Akash', 1793),  
           ('Akash', 111).  
  
conn.commit()  
cur.execute("select * from emp")  
print(cur.fetchall())  
conn.close()
```

Output

```
[('Akash', 1793), ('Akash', 111)]
```



046

#2

Step 1: Import corresponding library to make database connection, os & sqlite3.

Step 2: Now create the connection object using sqlite3 library & its connect() for creating New database

Step 3: Now create cursor object using db.cursor () from the connection object created.

Step 4: Now use db.execute() for creating the table with the column name & respective datatype.

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

Step 6: use db.commit() to complete the transaction using the connection object.

Step 7: use db.execute statement along with cursor object accessing the values from the database using db.select from where clause

780

Practical-6

Aim Database connectivity

#1

Step1: Import the (DBM) dbm library & used it for creating the database by specifying the name of the database along with corresponding flag.

Step2: use the object to create for accessing the given website & corresponding regularname for the website

Step3: check whether the given url address matches with regular name of the page is not equal to none, display the message about particular found/match or else not found/unmatched

Step4: use the close() to terminate database library

#1:

```
>>> import dbm  
>>> db = dbm.open ("database", flags='c', mode=0666)  
>>> db["name"] = "name"  
>>> if db["name"] == None:  
    print("Database not empty // match")  
else:  
    print("Database empty // Not match")  
Database not empty // match  
>>> db.close()
```

047

✓ 27/12

b4: Button (root, text: "Tomato", command: key)
b5: pack (side = Bottom)
spot -mainloop()

def ter ()

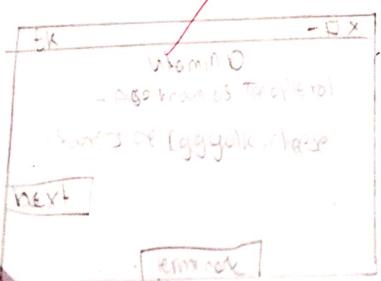
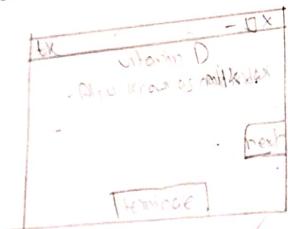
quit ()

b6: Button (root, text: "vitamins one",
command: main)

b7: pack ()

spot -mainloop()

output



045

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

No

580
steps: Define a function for initialization and call function method and finally call the first function created and trigger mainloop method

11.
12.
13.

No

```
from tkinter import *
```

```
root = Tk()
```

```
def main():
```

```
root = Tk()
```

```
root.config(bg="pink")
```

```
root.title("main")
```

```
root.minsize(100, 100)
```

```
l1 = Label(root, text="VC")
```

```
l1.pack()
```

l1 = label (root, text : " - Also known as etc etc in source are milk, tea etc")

```
l1.pack()
```

b1 = Button(root, text="next", command=set)

```
b1.pack(side=RIGHT)
```

b2 = Button(root, text="Tomato", command=tomato)

```
b2.pack(side=BOTTOM)
```

```
root.mainloop()
```

~~def set()~~

```
root = Tk()
```

```
root.config(bg="purple")
```

```
root.title("I")
```

```
root.minsize(100, 200)
```

l2 = label (root, text="vitamin D")

```
l2.pack()
```

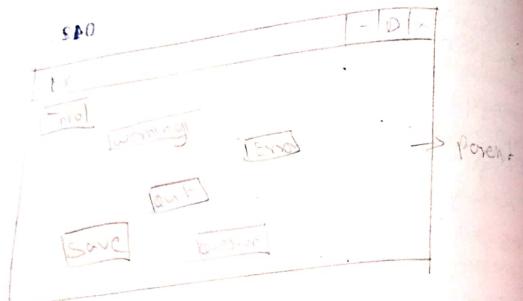
l3 = label (root, text : " - Also known as Tocopherol in source are Egg yolk, cheese etc")

```
l3.pack()
```

b3 = Button (root, text="next", command=main)

```
b3.pack(side=LEFT)
```

SAO



043

Traversing.

Step 1:- Define a function and create a object of the given window by using the three method namely config title and minimize

Step 2:- Create a button object and use the text and command attribute for triggering the given event and used grid method along specified and external padding specified and create another button object which will application to terminate

Step 3:- Define second function corresponding to second window with attributes: config , title ,minimize for a window object define one button object which will shift the focus onto the third window.

Step 4:- Create third window object and in this create two button object for moving on to first window for restarting the process and second button for terminating

140

write a program for message box (widget) and its method, which this widget may call in different algorithm

1. Import relevant method from tkinter library
2. Create an object corresponding to the parent window using Tk()
3. Define a function which will use the one click b method derived from tk.messagebox library
4. The attribute click() given method takes will specify window and another corresponds to its message display
5. Now create an object from Button() and plots it onto parent window with its title of button object specified and finally use its command attribute to encode its relevant function
6. Terminate the program by executing mainloop()

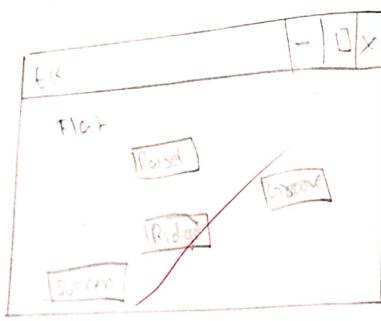
042

```
from tkinter import
    top=tk()
def msgb1():
    messagebox.showinfo("Info", "Python is interpreted lang")
def msgb2():
    messagebox.showwarning("Warning", "Wrong step")
def msgb3():
    messagebox.showerror("Error", "Syntax error")
def msgb4():
    messagebox.askquestion("Out", "Do you want to quit?")
def msgb5():
    messagebox.askokcancel("Save", "Do you want to save changes?")
def msgb6():
    messagebox.askquestion("Question", "Question")
B1=Button(top, text = "Info", command=msgb1)
B1.pack(side=LEFT)
B2=Button (top, text = "Warning", command=msgb2)
B2.pack()
B3=Button (top, text = "Error", command=msgb3)
B3.pack(side=RIGHT)
B4=Button (top, text = "Out", command=msgb4)
B4.pack()
B5=Button (top, text = "Save", command=msgb5)
B5.pack(side=LEFT)
B6=Button (top, text = "Question", command=msgb6)
B6.pack()
top.mainloop()
```

```

040
from tkinter import *
top = Tk()
B1 = Button(top, text="Flat", relief=FLAT)
B1.pack(side=LEFT)
B1 = Button(top, text="Raised", relief=Raised)
B2 = Button(top, text="Ridge", relief=RIDGE)
B2.pack()
B3 = Button(top, text="Groove", relief=GROOVE)
B3.pack(side=RIGHT)
B4 = Button(top, text="Sunken", relief=SUNKEN)
B4.pack()
B5 = Button(top, text="Left", relief=LEFT)
B5.pack()
top.mainloop()

```



Practical note

041

Components of GUI (Custom Attribute message box) write a program or various attribute which a button widget may assume related to its relief attribute.

Algorithm

- 1 Import relevant method from tkinter library
- 2 Create an object corresponding to the parent window using Tk()
- 3 Define a button object and place it onto the object corresponding to the parent window
- 4 use the text attribute for specifying the title for the button object
- 5 use the relief attribute with one style of a line involved for event triggering
- 6 use either the pack() or grid() for positioning the widget object onto the parent window & trigger the corresponding event by calling the mainloop()

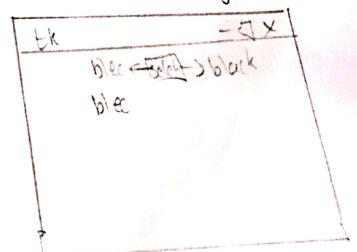
Q80

- Step 8: Import the relevant module from Tkinter library.
- Step 9: Define the object corresponding to the window in terms of no. of pixels.
- Step 10: Now define the frame object from the method and place it onto the parent window.
- Step 11: Create another frame object in left frame and place it onto parent window on its left side.
- Step 12: Similarly if define the right frame and subsequently define the button objects placed onto the given frame with their attributes as text, active bg and fg.
- Step 13: Now use the pack method along with the side attribute.
- Step 14: Similarly create the button object but it onto the frame object with side equal to right attribute set.
- Step 15: Add another button and put it on right frame object and term it as exit.
- Step 16: Use the pack method for all the objects and finally use the mainloop method.

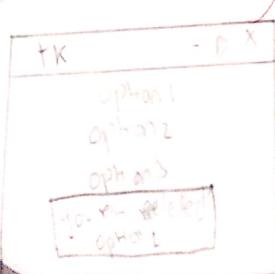
Q80

Source code:

```
# frame object:
from tkinter import *
top=Tk()
top.geometry('100x200')
frame=Frame(top)
frame.pack()
leftFrame=Frame(frame)
leftFrame.pack(side=LEFT)
# b1=Button(frame, text="select", activebackground="red",
#           fg="black")
b1.pack()
b2=Button(frame, text="modify", activebackground="blue",
          fg="purple")
b2.pack()
b3=Button(frame, text="Add", activebackground="yellow",
          fg="red")
b3.pack()
b4=Button(frame, text="EXIT", activebackground="green",
          fg="black")
b4.pack()
top.mainloop()
```



Radio button
 Step 1: Importing Tkinter module
 from tkinter import *
 root = Tk()
 def set1()
 selection = "None selected"
 if v.get() == 1:
 selection = "You selected the option " + str(v.get())
 l.config(text=selection)
 l = Label(root, text="Select your option")
 l.pack(anchor="center")
 v = IntVar()
 v1 = Radiobutton(root, text="Option 1", variable=v, value=1, command=set1)
 v1.pack()
 v2 = Radiobutton(root, text="Option 2", variable=v, value=2, command=set1)
 v2.pack()
 v3 = Radiobutton(root, text="Option 3", variable=v, value=3, command=set1)
 v3.pack()
 root.mainloop()



039
 # Radio button
 WAP making use of Tk control variable and button widget for selection of the given option
 Step 1: use the tkinter import relevant method
 Step 2: Define a function which tells the user about the given selection need of the multiple options available.
 Step 3: use the configuration method along with the label object and call the variable as an argument within the method.
 Step 4:- Now define the parent window and define the option using control variable.

Steps:- Now create an object from a radio button method which will take a call argument
 1) positioning on parent window.
 2) defining the text variable [1,2,3,4]
 3) Define the variable argument
 4) Corresponding value and trigger in given function.

Step 5:- pack method for the corresponding radio object is created and specify the attribute as anchor attribute.

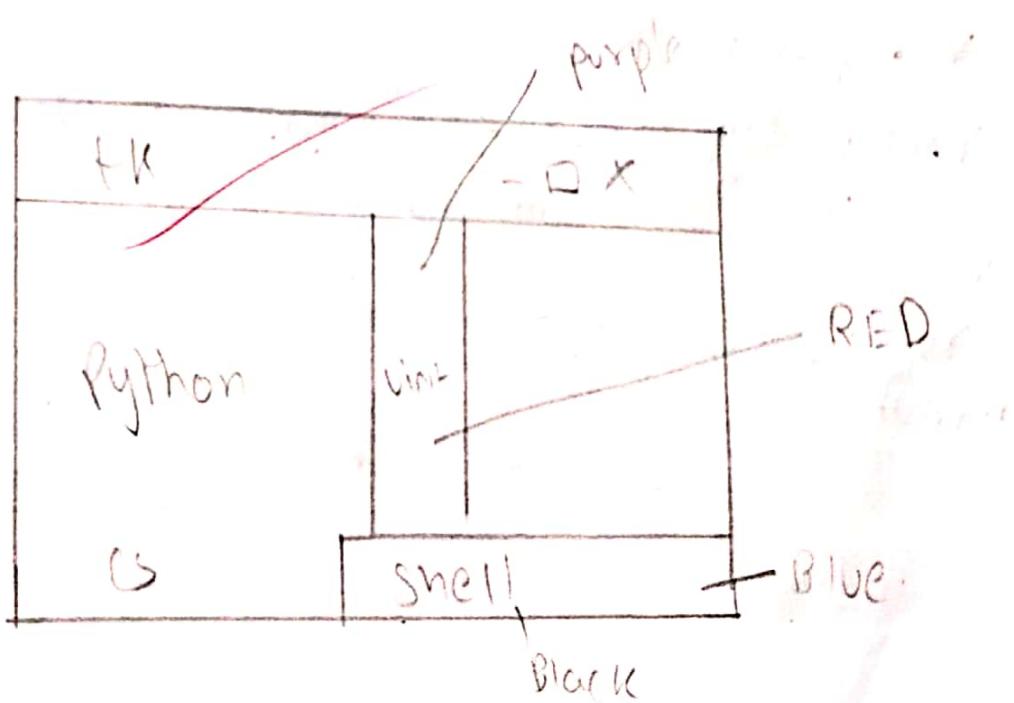
Step 6:- Now define the label object from the corresponding subsequently use pack method for this window and make use of the main loop method.

Code:-

038

```
from tkinter import *
root = Tk()
l1 = Label(root, text = "Python")
l1.pack(side = LEFT, pady = 30)
l2 = Label(root, text = "Vim", bg = "purple", fg = "red")
l2.pack(side = TOP, pady = 40)
m1 = Label(root, text = "CS")
m1.pack(side = LEFT, padx = 20)
m2 = Label(text = "Shell", bg = "blue", fg = "black")
m2.pack(side = LEFT, padx = 50)
root.mainloop()
```

Output



Aim:- GUI components

Step 1:-

use Tkinter library for importing its feature of text widget

Step 2:-

Create a variable from text method and position it on parent window.

Step 3:-

use the pack method along with the object created from text method.

Step 4:- use the mainloop method for integrating of corresponding events

Step 5:- use Tkinter Library for importing its feature of text widget.

Step 6:- Create a variable from text method and position it onto parent window.

Step 7:- use the pack method along with the object created from text method and use the parameters

Step 8 side = LEFT , padx : 20

 side = LEFT , padx : 30

 side = TOP , padx : 40

 side = TOP , padx : 50

6) **NEO**
 import re
 string = "April 2016 02 09.2001 sun 202 01-09-2001"
 pattern = "\d{2}\d{3}-\d{2}\d{3}-\d{2}\d{3}"
 result = re.findall(pattern, string)
 print(result)
 output
 [u'2016-04-2001', u'01-09-2001']

7) code
 string = "abc@gmail.com"
 p1 = "^\w+@"
 p2 = "[\w+-]+\w+"
 s1 = re.findall(p1, string)
 s2 = re.findall(p2, string)
 s3 = re.findall(p3, string)
 print(s1)
 print(s2)
 print(s3)
 output
 >>> [u'abc']
 [u'gmail.com']
 [u'abc', u'gmail.com']

035

- Q) write a re for extracting date in dd/mm/yyyy by using the find all method where the string has format
 To achieve the above objective we will use find all ()
 To extract the value we will use the pattern ~~\d{2}~~
 \d{2}\d{3}-\d{2}\d{3}-\d{2}\d{3}
- Q) write a regular expression for extracting the username from the mail ID, hostname and both username and hostname

~~lalit~~
 Jyoti

880

- Q) Write a regular expression to check whether the given mobile no starts with digit 8 or 9 and the total length of the digit should be almost 10.

To achieve the above objective we will use `match()`. To extract value & check condition we will use for ond if else conditional statement.

- Q) Work on extracting the word from a given string along with the space character between the word and word without space.

To achieve the above objective we will be using `findall()`. To extract the word along with space we will use `\w*` without space `\w+`.

- Q) Write a regular expression for extracting the first 2 words and last from a given string.

To achieve the above objective we will be using `findall()`.

To extract the first word we will use `\w+` and last word by `\w+\$`.

Q) code

```
034  
import re  
li = ["234567891", "8945634334", "9345678193"]
```

```
for val in li:  
    if(re.match("8|9", li[0][0:2], val)):  
        print("correct phone no")  
    else:
```

```
        print("incorrect phone no")  
output >>> incorrect phone no  
                    correct phone no  
                    correct phone no
```

Q) Import re

```
string = "python is important"  
result 1 = re.findall(r"\w+", string)  
print(result 1)
```

```
result 2 = re.findall(r"\w+", string)  
print(result 2)
```

Output

```
>>> [python, is, is, important]  
['python', 'is', 'important']
```

Q) import re

```
string = "python is important"
```

```
result 1 = re.findall(r"\w+", string)  
print(result 1)
```

```
result 2 = re.findall(r"\w+\$+", string)
```

```
print(result 2)
```

```
Output >>> [python]  
[important]
```

No. 1

```
code
1) import re
string = "hello(123 4 abc)345"
pattern = "(\d+)(\w+)"
result = re.findall(pattern, string)
result[0][0]
print(result)
print(result)

output
[123, '345']
['hello', 'abc']
```

2) code

```
import re
string = "python is interpreted language"
result = re.search('xyz', string)
if result:
    print("Match found")
else:
    print("Not found")

output
Match not found
```

033

Practical :-

Topic: Regular Expressions:-

Step 1:- Import re module declare pattern and declare sequence
use match method with define expression if arguments
matched then print th otherwise print pattern not
found

Step 2:- Import re module declare pattern with value
use it find all() with arguments and print th
same

Step 3:- Import re module declare pattern with metacharacter
use a split() and print output

4) write a regular expression for finding all th method string at
th string at th beginning of th given frequency

To achieve th above objective we will be using search()
To check th condition we will use if else conditional
statement

180

Q) write a program using an assert to check if all list elements are empty

Step 1:- Define a function which accepts an argument and check using the assert statement whether a given list is empty list and accordingly with the message

Step 2:- close the function and in the body of programming and define certain element in list and take some appropriate action

Q) write a program to check in range of the age of the student in given class and if the age do not fall in given range else in value error exception otherwise return the valid number

Step 1:- Define a function which will accept the age of the student from the standard input

Step 2:- use the if condition to check whether the input age falls in the range and to return the age else use in value error exception

Step 3:- Define a while loop to check accept age of the student and terminate the loop when the age

accept with value in range and print the message

valid range

032

#program :-

```
def assert_(n)
    assert(len(n) == 0)
    print("list is empty")
var l=[ ]
print(assert_(var))
```

Output

list is empty

#program :-

```
def acceptage():
    age = int(input("Enter age"))
    if age > 30 or age < 6 :
        raise ValueError
    return age
```

valid = False

while not valid

try:

ages = acceptage()

valid = True

except ValueError:

print("Not a valid age")

Output

Enter age : 4

Not a valid age

Enter age : 18

No.

```
080  
#include <iostream>  
using namespace std;  
int main()  
{  
    int class1;  
    cout << "Enter class" << endl;  
    cin >> class1;  
    if(class1 < 0 || class1 > 100)  
    {  
        cout << "Enter numeric value" << endl;  
        return 1;  
    }  
    cout << "Enter class" << endl;  
}
```

#include

```
try  
{  
    f = fopen("abc.txt", "w");  
    f.write("unit sonaware");  
}  
except (ferror)  
{  
    cout << "Error writing on file" << endl;  
}  
else  
{  
    cout << "Operation carried out successfully" << endl;  
}  
f.close();
```

Output

Operation carried out successfully

No.

Practical no.-3

081

Aim: Program to understand exception handling

1. write a program using the exception method of the native arithmetic error

Step 1: use the try block and except the input using the raw input method and convert it into the integer datatype and subsequently terminate the block

Step 2: use the except block with the exception name as value error and display the appropriate message if the suspicious code is part of the try block

2. write a program for accepting the file in a given mode and use fstream::invalid_error as an exception of the given input

Step 1: within the try block open the file using the write mode and write some content onto file:

Step 2: use the except block with the error and display the message regarding missing of the file or incompatibility of the mode use the else block to display a message that the operation is carried out successfully.

020

No.	
10	
11	
12	
13	

Output:
=> f=fac()
=>x=citer()
=>x=next()
11=1
=>x.next()
11=2
=>x.next()
11=6

030

Jath
J.M. 16/07

880

No.

myobj = odd()
myfiver = five(myobj)
X = int(input("Enter a number:"))

For immixer :

```
if (i < X):  
    print(i)
```

>>> Enter a number: 15

1
3
5
7
9
11
13
15
17
19
21
23
25
27
29
31
33
35
37
39
41
43
45
47
49
51
53
55
57
59
61
63
65
67
69
71
73
75
77
79
81
83
85
87
89
91
93
95
97
99

code:

```
class fact:  
    def __init__(self):  
        self.f = 1  
        self.n = 1  
    def __next__(self):  
        if self.f <= 10:  
            num = self.f  
            self.f += 1  
            return num  
        else:  
            raise StopIteration
```

580

Step 1: Define a function even with 0 parameters
elements can use the map method with help of lambda function
and two arguments as they are same

Step 2: Define a function even with 0 parameters
elements can use the map method with help of lambda function
and add two arguments as they are same

Step 3: Define a function even with 0 parameters
elements can use the map method with help of lambda function
and add two arguments as they are same

Step 4: Define a class and what define
steps. Define a class and within and what define
the whole class which will initialize the first element
within the container object

Step 5: Now use the next () and define the logic for
displaying odd value.

```
def even(x):  
    return "Even"  
else:  
    return "Odd"
```

```
# odd numbers  
class odd:
```

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

```
        self.num = 1
```

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

```
        num = self.num
```

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

```
        dict = next__(self)
```

```
        num = self.num
```

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

```
        return num
```

Q. 7 in range (3) when list(map(lambda x: x**2, [1, 2, 3])) 028
when list (value)
=> [0, 1]
[1, 4]
[4, 9]

def even(x):

```
listnum = [0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25]
```

```
listnum = list(map(lambda x: x**2, listnum))
```

```
print(listnum)
```

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

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

```
    return "Even"
```

```
else:
```

```
    return "Odd"
```

```
# odd numbers
```

```
class odd:
```

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

```
        self.num = 1
```

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

```
        num = self.num
```

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

```
        dict = next__(self)
```

```
        num = self.num
```

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

```
        return num
```

new class next()

mytuple = ("banana", "orange", "apple")

myiter1 = iter(mytuple)

print(next(myiter1))

myiter2 = iter(mytuple)

print(next(myiter2))

myiter3 = iter(mytuple)

print(next(myiter3))

>>> banana

orange

apple

for loop

mytuple1 = ("Anil", "Anil", "Sunawane")

for x in mytuple1:

print(x)

>>> ~~Anil~~ Anil

Sunawane

square and cube.

def square(x):

y = x * x

return y

def cube(x):

z = x * x * x

return z

Step1 = [square, cube]

Step 4: open file obj in read mode declare a variable and return the object by tell method and store its output consequently in variable.

Steps: use file seek method with its arguments with opening the file obj in read mode and it uses subsequently

step 4 open file with read mode also use file readlines method and start its outputs. consequently end print it some for counting its length use the for conditional statement and display its length.

12

```
# 411:
file obj = open("abc.txt", "r")
s1 = file obj.read(10)
print ("seek(0,0) is:", s1)
file obj.close()
>>> ("seek(0,0) is:", None)
>>> file obj.seek(0,0)
read(10) open('abc.txt','r')
s1 = file obj.read(10)
print ("seek(0,0) is:", s1)
print ("seek(0,1) is:", None)
file obj.close()
>>> ("seek(0,1) is:", None)
file obj = open('abc.txt')
s1 = file obj.seek(0,1)
print ("seek(0,2) is:", s1)
print ("file obj close()")
>>> ("seek(0,2) is:", None)
```

26

finding length of different lines exists within lines:
file obj = open("abc.txt", "r")
s1 = file obj.readlines()
print ("output is:", s1)
for line in s1:
 print (len(line))
file obj.close()
>>> output: [[lourik sir data structure]]

↓
↓
↓
↓

last

No	Code
1.	<code>file obj = fopen("file.txt", "r")</code>
2.	<code>file obj = fopen("file.txt", "w")</code>
3.	<code>file obj = fopen("file.txt", "a")</code>
4.	<code>file obj = fopen("file.txt", "r+");</code>
5.	<code>file obj = fopen("file.txt", "w+");</code>
6.	<code>file obj = fopen("file.txt", "a+");</code>
7.	<code>file obj = fopen("file.txt", "rb");</code>
8.	<code>file obj = fopen("file.txt", "wb");</code>
9.	<code>file obj = fopen("file.txt", "ab");</code>
10.	<code>file obj = fopen("file.txt", "rb+");</code>
11.	<code>file obj = fopen("file.txt", "wb+");</code>
12.	<code>file obj = fopen("file.txt", "ab+");</code>

write mode:

```

file obj = fopen("file.txt", "w")
file obj.write("Hello World")
file obj.close()

```

append mode:

```

file obj = fopen("file.txt", "a")
file obj.write("Hello World")
file obj.close()

```

read mode:

```

file obj = fopen("file.txt", "r")
file obj.read()
file obj.close()

```

append + read mode:

```

file obj = fopen("file.txt", "r+")
file obj.write("Hello World")
file obj.read()
file obj.close()

```

read + append mode:

```

file obj = fopen("file.txt", "a+")
file obj.read()
file obj.write("Hello World")
file obj.close()

```

objective: Demonstrate the use of different file accessing modes, different attributes read methods.

Step1: Create a file object using open method and use the write method and use the write-access mode followed by writing some contents onto the file and then closing the file

Step2: Now open the file in read mode and then use read(), readline() and readlines() and store the output in variable and finally display the contents of variable.

Steps: Now use the file object for finding the name of the file mode in which its opened whether the file is still open or close and finally the output of the soft space attribute

```

fileobj = open("abc.txt", "w") # file open (write mode)
fileobj.write("computer science subjects\n")
fileobj.close()

# read()
fileobj = open("abc.txt", "r")
str1 = fileobj.read()
print("The output of read method : ", str1)
fileobj.close()

# readline()
fileobj = open("abc.txt", "r")
str2 = fileobj.readline()
print("The output of readline method : ", str2)
fileobj.close()

# readlines()
fileobj = open("abc.txt", "r")
str3 = fileobj.readlines()
print("The output of readlines method : ", str3)
fileobj.close()

>>> ['The output of readlines method : ', 'computer science subjects\n', 'DBMS\n', 'Python\n', 'DSn']
```

file attributes:

```

a = fileobj.name
print("name of file (name attribute) : ", a)
>>> ('name of file (name attribute)', 'abc.txt')
```