★★ INDEX ★★

IInd Semister

No.	Title	Page No.	Date	Staff Member's Signature
1)	Demonstrate the use of different file accessing mode attributes read method.	21	26/11/19	
2)	Iteration.	24	3/12/19	
3)	Exception	27	10/12/19	
4)	Regular Expression	30	17/12/19	
5(A)	GUI Component	33	7/01/2020	
5(B)	GUI Component (Radio Button) (Scrollbar & List Box)	35	14/1/2020	
5(c)	GUI ~~c, bx~~ list box, photo-image.	43	21/01/20	
5(d)	GUI Component of a) Spin box (b) panned window (c) Canvas	46	11/02/20	
6)	database connectivity	49	18/02/20	

Scanned with CamScanner

PRACTICAL-1.

Aim :- Demonstrate the use of different file accessing modes different attributes read methods.

Step 1 :- Create a file object using open method ~~meth~~ and use the write access mode followed by writing some content 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 content of variable.

Step 3 :- Now use the fileobject, for finding the name of the file, the file mode in which its opened wheather the file is still open or close and finally the output of the softspace attribute.

Scanned with CamScanner

```
fileobj = open("abc.txt","w") # file open(write mo
fileobj.write("I am a computer Science student\n
fileobj.write("and Studing in FY BSC\n") # file wri
fileobj.close() # file close.

fileobj = open("abc.txt","r") # read mode.
# read()
str1 = fileobj.read()
print("The output of read method : ", str1)
fileobj.close()
>>> ('The output of read method : ', 'I am a compu
    Student Student\n and Studing in FYBSC\n'

#readline()
fileobj = open("abc.txt","r")
str2 = fileobj.readline()
print("The output of readline method : ", str2)
fileobj.close()
>>> ('The output of readline method : ', I am
    computer Science Student\n')

#readlines()
fileobj = open("abc.txt","r")
str3 = fileobj.readlines()
print("The output of read lines method : ", str3)
fileobj.close()
>>> ('The output of readlines method : ', 'computer Scien
    Student\n and Studing in FY BSC\n')
# file attributes
a = file obj.name
print("name of file (name attribute) : ", a)
>>> ('name of file (name attribute) ', 'abc.txt')
b = fileobj.closed
print("(close) attributes", b)
>>> ("(close) attributes", 'True')
```

Scanned with CamScanner

Step 4 :- Now, open the fileobj in write mode write some another content close subsequently then again open the fileobj in 'w+' mode that is the update mode and write contents.

Step 5 :- Open fileobj in read mode display the update written content and close open again in 'r+' mode with parament passed and display the output subsequently

Step 6 :- Now, open file obj in append mode open write method write contents close the file obj again open the fileobj in read mode and display the appending output.

Scanned with CamScanner

```
c = fileobj.mode
Print("file mode", c)
>>> ('file mode', 'r')
d = fileobj.softspace
Print("softspace", d)
>>> ('softspace', 0)
```

w+ mode

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

r+ mode

```
fileobj = open("abc.txt","r+")
S1 = fileobj.read(6)
Print("output of r+", S1)
fileobj.close()
>>> ('output of r+', 'Adarsh')~~Pande~~
```

append mode

```
fileobj = open("abc.txt","a")
fileobj.write("data structure")
fileobj.close()
fileobj = open("abc.txt","r")
S3 = fileobj.read()
Print("output of append mode is", S3)
fileobj.close()
>>> ('output of append mode is', 'Adarsh Pandey data structure')
```

Write mode

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

read mode

```
fileobj = open("abc.txt","r")
S = fileobj.read()
Print("output of read method", S)
>>> ('output of read method', 'Adarsh Pandey')
```


Scanned with CamScanner

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

#seek()
fileobj = open("abc.txt","r")
st = fileobj.seek(0,0)
Print("seek(0,0) is : ", st)
fileobj.close()
>>>('seek(0,0) is : ', None)

file obj = open("abc.txt","r")
st1 = file obj.seek(0,1)
Print("seek(0,1) is :-", st1)
file obj.close()
>>>('seek(0,1) is :-' None)

file obj = open("abc.txt","r")
st2 = file obj.seek(0,2)
Print("seek(0,2) is :-", st2)
fileobj.close()
>>>('seek(0,2) is : ', None)

# finding length of different lines : list
within lines
file obj = open("abc.txt", "r")
st3 = fileobj.readlines()
Print("The output is : ", st3)
for line in stert:
    Print(len(line))
fileobj.close()
>>>(The output is :- [Adarsh Pandey Data structure])
30.
```

Scanned with CamScanner

Step 7 :- Open the fileobj in read mode declare a variable and perform fileobj dot tell method and Store the output consequently in variable.

Step 8 :- Use the seek method with the argument with opening the file obj in read mode and closing subsequently

Step 9 :- Open file obj 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 conditional Statement and display the length.

31/12/17

Scanned with CamScanner

PRACTICAL:- 02.

Aim :- Iterators.

Step1:- Create a tuple with elements that we needed to iterate using the iter and next method. The number of time we use the iter and next method we will get the next iterating element in the tuple. Display the same.

Step2:- The similar output can be obtained by using for conditional statement. The iterable variable is to be declared in for loop which will iterate.

Step3:- Define a function name square with a parameter which will obtained output of square value of the given number. In similar function declare value to get the value raised 3 and return the same.

Step4:- Call the declared function using function call.

Scanned with CamScanner

```
#itor() & next()
mylist1 = ("banana", "Orange", "apple")
myitor1 = itor(mylist1)
print(next(myitor1))
myitor2 = itor(mylist)
print(next(myitor2))
myitor3 = itor(mylist)
Print(next(myitor3))
>>> banana
    orange
    apple
# for loop
Mytuple1 = ("car", "Bike", "Ship")
for x in mytuple1:
    Print(x)
>>> car
    Bike
    Ship
# square and cube
def square(x):
    y = x * x
    return y
def cube(x):
    z = x * x * x
    return z
funct1 = [square, cube]
for r in range(5):
    value = list(map(lambda x: x(r), funct1))
    Print(value)
```

```
>>> [0, 0]
    [1, 1]
    [4, 8]
    [9, 27]
    [16, 64]
    [25, 216].
```

Scanned with CamScanner

map()

```
listnum = [0,4,8,2,9,10,12,15,25,19,30]
listnum = 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))
```

```
>>> [0,4,3,2,4,0,2,0,0,4,0]
EVEN
ODD
ODD
ODD
ODD
EVEN
ODD
EVEN
EVEN
ODD
EVEN
```

odd number

```
class odd:
    def __iter__(self):
        self.num=1
        return self
    def __next__(self):
        num.self=num
        num.self+=2
    def __next__(self):
        num=self.num
        self.num+=2
        return num
myobj = odd( )
Myiter = iter(myobj)
x= int(input("Enter anumber: "))
for i in myiter:
    if (i<x):
        Print(i).
```

Scanned with CamScanner

Step 5 :- using for conditional Statement Specifying the range use the list type-casting with map method declaring a 'lambda' i.e annonymus function and print the Same.

Step 6 :- Declare a listnum variable and declare Some elements then use the map method with help of lambda function give two argument display the output.

Step 7 :- Define a function even with a parameter then using conditional Statement do check whether the number is even and odd and return respectively.

Step 8 :- Define a class and within that define the iter() method which will initialize the first element within the container object

Step 9 :- Now, use the next() and define the logic for displaying odd value.

Scanned with CamScanner

Step 10:- Define an object of a class.

Step 11:- Accept an number from the user till which we want to display the odd number.

Scanned with CamScanner

>>> Enter a number : 25

1 9
3 12
5 15
7 19
9
11
13
15
17
19
21
23

10/12

Scanned with CamScanner

IO error

```
try:
    fo = open ("efg.txt", "w")
    fo.write ("python is an intendent language")
except IOError:
    Print ("Enter mode for file operation")
else:
    Print("operation is successfull")
```

>>> operation is successfull

R outputs

>>> Enter mode for file operation.

value Error

```
try:
    x = int(input("Enter a statement: "))
except Value Error:
    Print ("Arithmetic Error!")
else:
    print("operation is successfull")
```

>>> Enter a statement: abc
Arithmetic Error!

>>> Enter a statement: 123
Operation is successfull.

Scanned with CamScanner

Practical :- 03.

Topic :- Exception.

Step 1 :- In the try block open an file in the open mode with write mode. write some contents in file.

Step 2 :- In except (IO Error) block use the error in IO Error use the approviation message to display.

Step 3 :- Else display the operation is success-full!.

Step 4 :- In try block accept an input from the error.

Step 5 :- In except block use value error and print the same message.

Step 6 :- Else display the operation is success full.

Scanned with CamScanner

Step 7 :- Accept a integer value from the user. In the try use the division method

Step 8 :- For the exception to the raised use the except keyward (and) i.e Type Error print Incompatible values.

Step 9 :- ~~the~~ use except with error of zero division Error and print the message according that is if entered ~~the~~ number is zero not able to perform operation.

Step 10 :- Declare Static variable and Value

Step 11 :- For Multiline execution use the error type by seperation then with a comma.

Scanned with CamScanner

Type error, zero division error.

```
a = 16
b = input("Enter the letter : ")
try :
    print (a/b)
except TypeError :
    print ("Incompatible value")
except ZeroDivisionError :
    print ("Denomination is zero so operation
            cannot be performed")
```

```
>>> Enter the letter : t
    Incompitable value.
>>> Enter the letter : 8
    2.
>>> Enter the letter : 0
    Denomination is zero so operation cannot
    be performed
```

Multiline exception.

```
a, b = 2/0
try :
    print (2/0)
    print ('20' + 30)
except (TypeError, ZeroDivisionError) :
    print ("Invalid Input")
```

```
>>> 2.0
>>> 2030
>>> Invalid Input
```

Scanned with CamScanner

#using except keyword.

```
try:
    a= open("efg.txt","w")
    a.write("language")
except IOError:-
        Print("Exception")
else:-
        Print("successfull")
def x():
    l=[ ]
    Print(len(l))
def y():
    list1=[3,2,4,1]
    Print(len(li))
Print(x())
Print(y())
```

Output:-

```
successfull
0
None
4
None
```

#raise Method:-

```
try:
    a= int(input("Enter a value:"))
    raise ValueError
except ValueError:
    Print("Please enter integer value!")
>>> Enter : XYZ
    Please enter integer value!
```

Scanned with CamScanner

Step 12 :- Use try block open a file in write mode and surrounding / subsequently enter value in the file.

Step 13 :- Use the IOError and display appropriate message.

Step 14 :- define a function with empty list and calculate the length of the list.

Step 15 :- define another function ~~x~~ initialize or declare some the element in list and calculate the length of the same and display the same.

Step 16 :- In try block accept input from the user and if the user enters character values raise an Error that is saying up Enter integer value.

24/12

Scanned with CamScanner

es

PRACTICAL-04.

TOPIC :- Regular expression.

Step 1:- Import re module declare pattern and declare sequence use match method with declare arguments if arguments matched than 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 argument and print the same.

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

Scanned with CamScanner

```
Import re
pattern = r"FYCS"
Sequence = "FYCS represent computer science"
if re.match (pattern, sequence):
    Print("Matched patterned found.")
else:
    Print("Not found")
>>> matched patterned found!
#numerical value (segregation)
Import re
Pattern = r'\d+'
Sequence = 'hello123', howdy789, 45 howru'
Output = re.findall (pattern, sequence)
Print (output)
>>> ['123', '789', '45'].
# Split( )
Import re
pattern = r'\d+'
String = 'hello123', howdy789, 45 Howru'.
Output = re.Split (pattern, String)
print (output)
>>> ['hello', 'howdy', 'Howru']
```

Scanned with CamScanner

```
#no_space.
import re
string = 'abc def ghi'
pattern = r'\s+'
replace = ''
output = re.sub(pattern, replace, string)
print(output)
>>> abcdefghi
#group()
import re
sequence = 'python is an interesting language'
v = re.search('\Apython', sequence)
print(v)
output = v.group()
print(output)
>>> <_sre.SRE_Match object at 0x0281DF00>
    python.
#verifying the given set of phone number.
import re
list1 = ['8009865764', '6876543210', '9765846235',
         '8975648533 5']
for value in list1:
    if re.match(r'[8-9]{1}[0-9]{9}', value
      or len(value) == (0):
        print("Criteria matched for all number
    else:
        print("Criteria failed")
>>> Criteria Not matched for all number
    Criteria Not matched for all number
    Criteria matched for all number
    Criteria matched for all number.
```

Scanned with CamScanner

Step 4 :- Import re module declare String and according 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 with dot operator gives memory location using 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 number are in range of 0 to 9 and check whether the entered number are equal to 10 if criteria matched print all number matches otherwise print failed.

Scanned with CamScanner

Step 7 :- Import re modulus declared a string use the modulus with find-all() 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 seperating 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 variable with initial value as zero use for condition and subsequently use the if condition check whether condition satisfy add up the or else increment value and display the value subsequently

Scanned with CamScanner

Vawels

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

host & 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

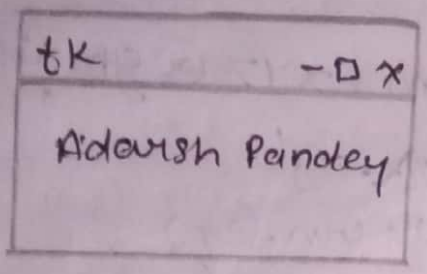
```
import re
s = 'Mr.a, Ms.b, Ms.c, Mr.t'
p = r'[Ms|Mr]+'
o = re.findall (p, s)
print (o)
M = 0
f = 0
for v in o:
    if (v == 'Ms'):
        f = f+1
    else:
        M = M+1
print ("No of Males is :", M)
print ("No. of female is :", f)
>>> ['Mr', 'Ms', 'Ms', 'Mr']
```

['No. of Male is 2)
(No of female is 2).

Q.1 # Creation of parent window

```
from Tkinter import *
root.tk()
l= Label(root, text="Adarsh Pandey")
l.pack()
root.mainloop()
```

OUTPUT :-



Q.2. ~~it~~

```
from Tkinter import *
root.tk()
l = Label(root, text="Adarsh Pandey")
l.pack()
l1 = label(root, text="FYBSC", bg="blue, fg="g
    font="10")
l1.pack(side=LEFT, padx="20")
~~l2 = pack(side=LEFT, pady="30")~~
~~l2 = pack()~~
l2 = Label(root, text="FYBSC", bg="green",
    fg="Yellow"; font="30")
l2.pack(side=LEFT; pady="30")
l3 = Label(root, text="FYBSC", bg="light blue",
    fg="black", font="20")
l3.pack(side=TOP, ipad="40")
```

Scanned with CamScanner

PRACTICAL :- 05 (A)

Topic :- Gui components.

Q.1

Step1 :- Use the tkinter libraries for importing the features of the text widget.

Step 2 :- Create an object using the tk().

Step3 :- Create a variable using the widget label and use the text method

Step4 :- Use the mainloop() for triggering of the corresponding above mention event

Q.2

Step1 :- Use the tkinter library for importing the feature of the text widget.

Step2 :- Create a variable from the text method and position it on the parent window

Step3 :- Use the pack() along with the object created from the text() and use the parameter.

1) side = LEFT, padx = 20
2) side = LEFT, pady = 30
3) side = TOP, padx = 40
4) side = TOP, pady = 50

Scanned with CamScanner

Step 4:- Use the mainloop() for the trigger of the corresponding event.

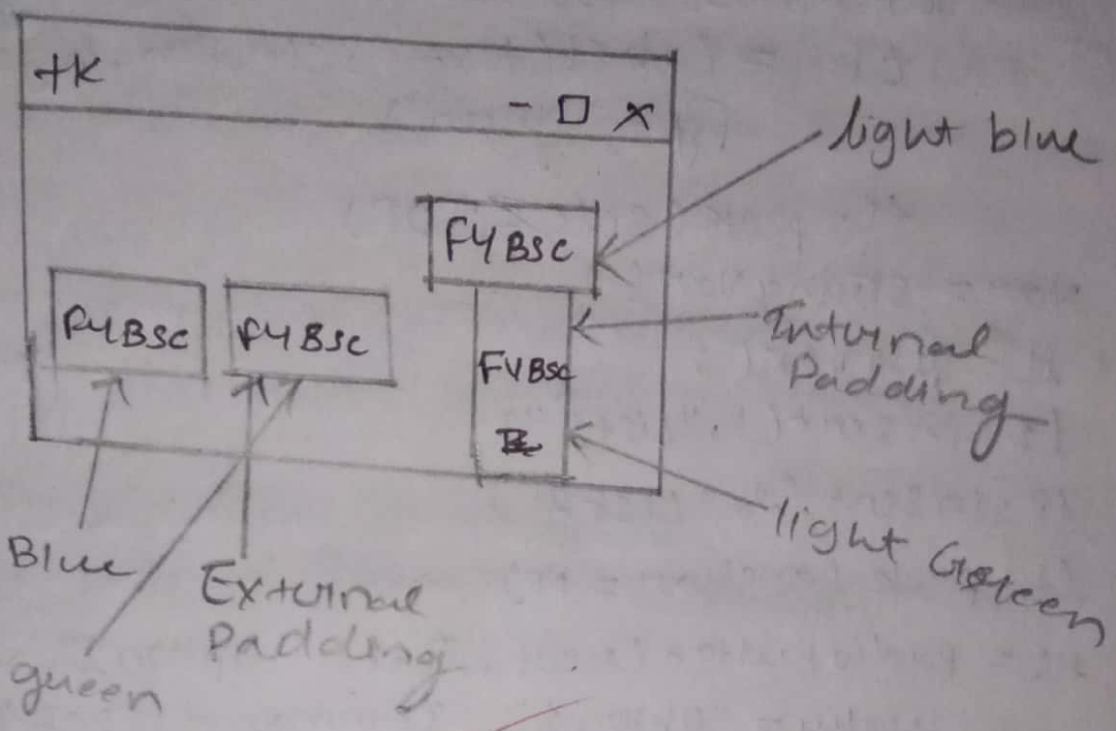
Step 5:- Now repeat above steps with the table() which takes the following arguments.

1) Name of the parent window
2) Text attribute which defines the string
3) The background color (bg)
4) The foreground fg and than use the pack() with a relevant padding attributes.

Scanned with CamScanner

```
lu = label(root, text = "FY BSC", bg = "Light Green",
      fg = "white", font = "10")
lu.pack(side = TOP, pady = "50")
root.mainloop( )
```


OUTPUT :-



Tylor

Radio buttons

```
From tkinter import *
root = Tk()
root.geometry("100 x 100")
def sel():
    selection = "You Select" + str(var.get())
    t1 = Label(text = selection, bg = "white",
               fg = "green")
    t1.pack(side = TOP)
var = StringVar()
l1 = Listbox()
l1.insert(1, "List1")
l2.insert(2, "List2")
l1.pack(anchor = N)
r1 = RadioButton(root, text = "Option 1", variable = var
    value = "Option 1", command = sel)
r1.pack(anchor = N)
r2 = RadioButton(root, text = "Option 2", variable = var
    value = "Option 2", command = sel).
r2 = pack(anchor = N)
root.mainloop()
```

Scanned with CamScanner

PRACTICAL :- 5 (B).

AIM :- Gui Component

Q.1.

Step1 :- Import the relevent method from the tkinter library create an object with the parent window.

Step2 :- Use the parent window object along with the geometry() declaring specific pixel size of the parent window.

Step3 :- Now define a function which tells the user about the given selection made from multiple option available.

Step4 :- Now, define the parent window and define the object with control variable

Step5 : Use the listbox() and insert option on the parent window along with the pack() with specifying anchor attribute.

Step6 : Create an object from radio button which will take following argument & parent window object text variable which will take the values option no 1, 2, 3, ... variable argument corresponding value & trigger the function declared.

Scanned with CamScanner

Step 7: Now call the pack() for radio obj
so created and specify the argument wi
anchor attribute.

Step 8: Initially make use of the mainloop()
along with parent object.

Q.2

Step 1: Import relevant method from the
tkinter library.

Step 2: Create a parent ~~window~~ object corre
ponding to the parent window.

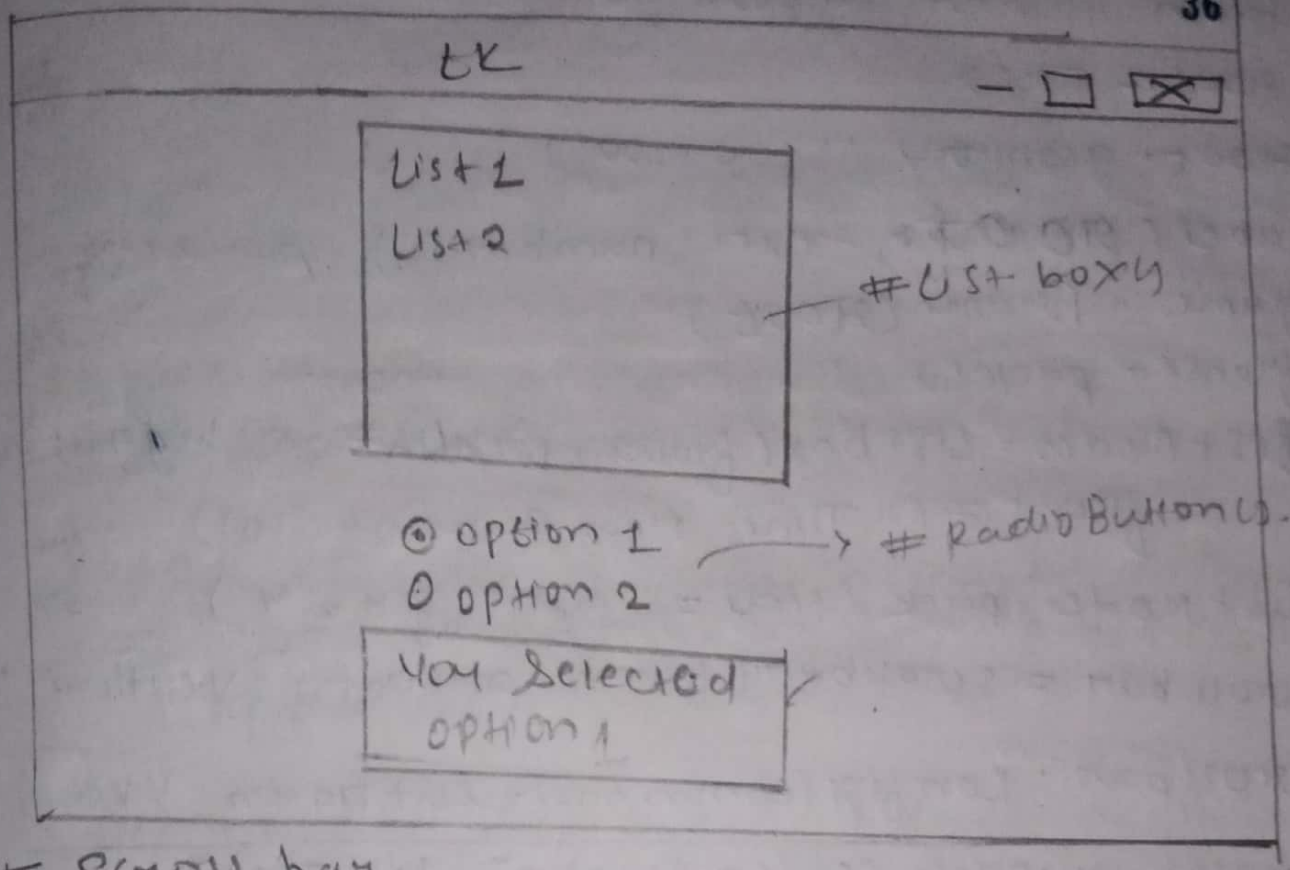
Step 3: Use the geometry() for laying of the
window

Step 4: Create an object and use the scroll bar

Step 5: Use the pack() along with the scro
bar() object with side and fill attribute

Step 6: Use the mainloop() with the parent
window object

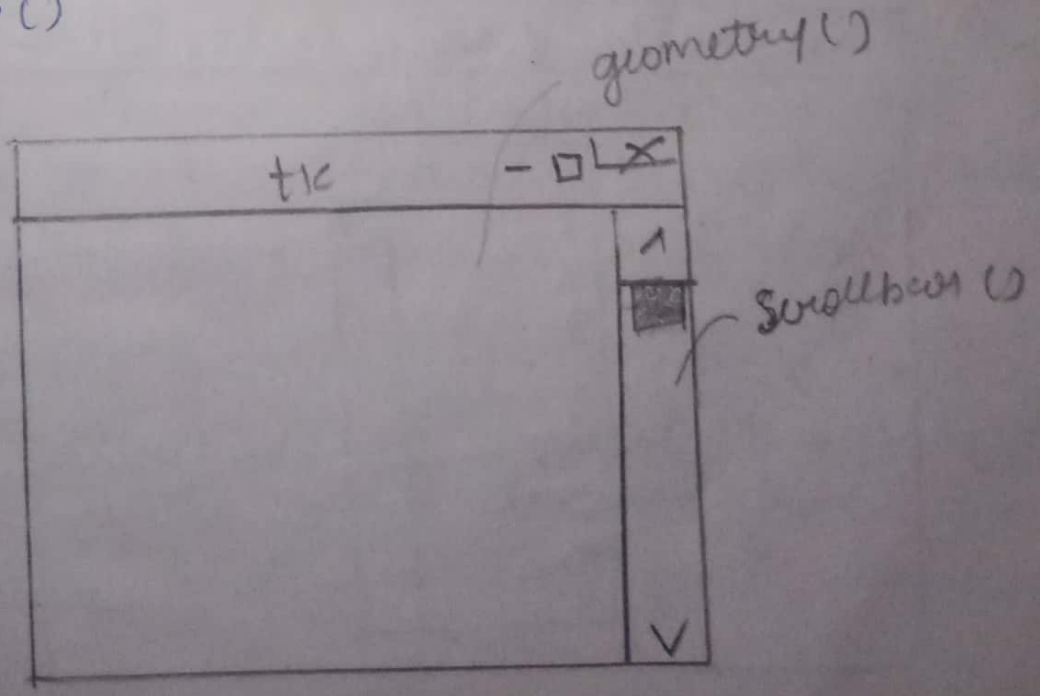
Scanned with CamScanner



Scroll bar

```
from tkinter import *
root = Tk()
root.geometry("100x100")
s = Scrollbar()
s.pack(side="right", fill="y")
root.mainloop()
```

Output:-

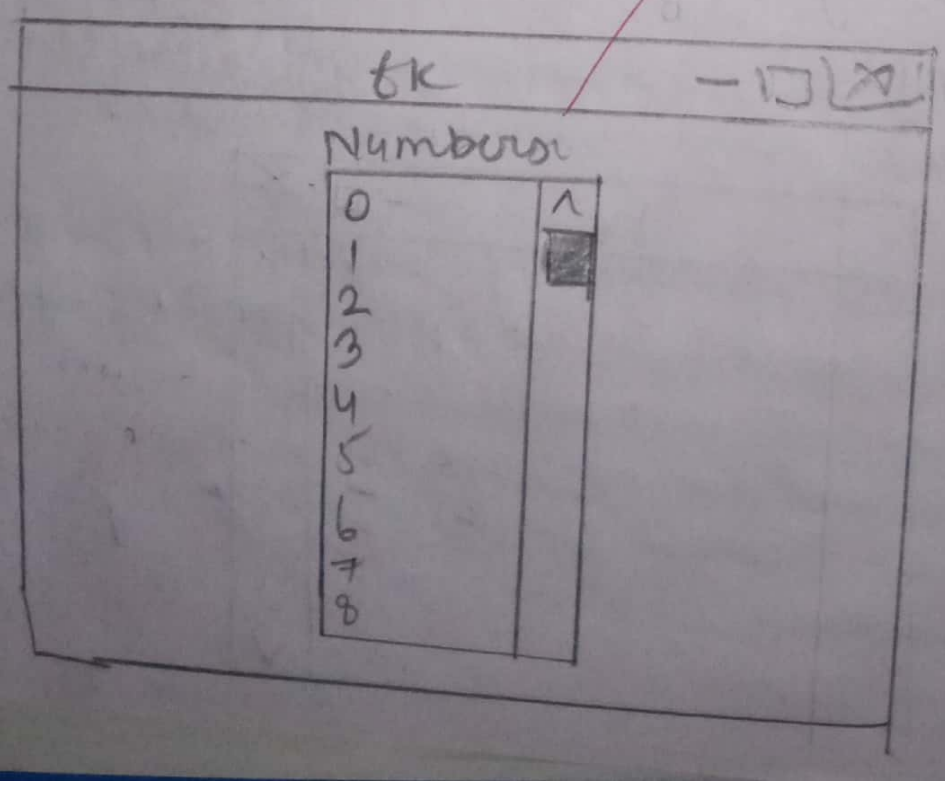


Scanned with CamScanner

#3 Using frame widget.

```
from tkinter import *
root = Tk()
root.geometry ("100x100")
label( root, text="number" ).pack()
frame = Frame(root)
frame.pack()
listNodes = Listbox( frame, width = 20, height = 22,
        font = ("Times New Roman", 10))
listNodes.pack (side="left", fill='y')
scrollbar = scrollbar( frame, orient = "Vertical")
scrollbar.config(command=listnodes.yview)
scrollbar.pack (side="right", fill="y")
for x in range(100):
    listnodes.insert( END, str(x))
window.mainloop()
```

Output:



Q.3

Step 1 :- Import the relevant libraries from the tkinter method.

Step 2 :- Create an corresponding object of the parent window.

Step 3 :- Use the geometry manager with pixel size (680 x 500) or any other suitable pixel value.

Step 4 :- Use the label widget along with the parent object created and subsequentially use the pack method.

Step 5 :- Use the frame widget ~~along~~ along with the parent object created and use the pack ~~pa~~ method.

Step 6 :- Use the listbox method along with the attributes like width, height, font. To create a listbox method object use pack() for the same.

Step 7 :- Use the scrollbar () with a object use the attribute of vertical then configure the same with object created from the scrollbar () and use pack().

Scanned with CamScanner

Q.4

Step 1 : Import relevent method from tkinter library.

Step 2 : Define the object corresponding the window and define the size of parent window in term of no. of pixels

Step 3 : Now, define the frame object from the method and place it on to the parent window.

Step 4 : Create another frame object same as the left frame and put it on the parent window on its LEFT side

Step 5 : Similarly define the RIGHT frame and subsequently define the following button object placed on to the given frame with the attribute as text, active background and foreground

Step 6 : Now, use the pack() along with the side attribute

Step 7 : Similarly create the button object corresponding to the MODIFY operator put it on to the frame object on side = right

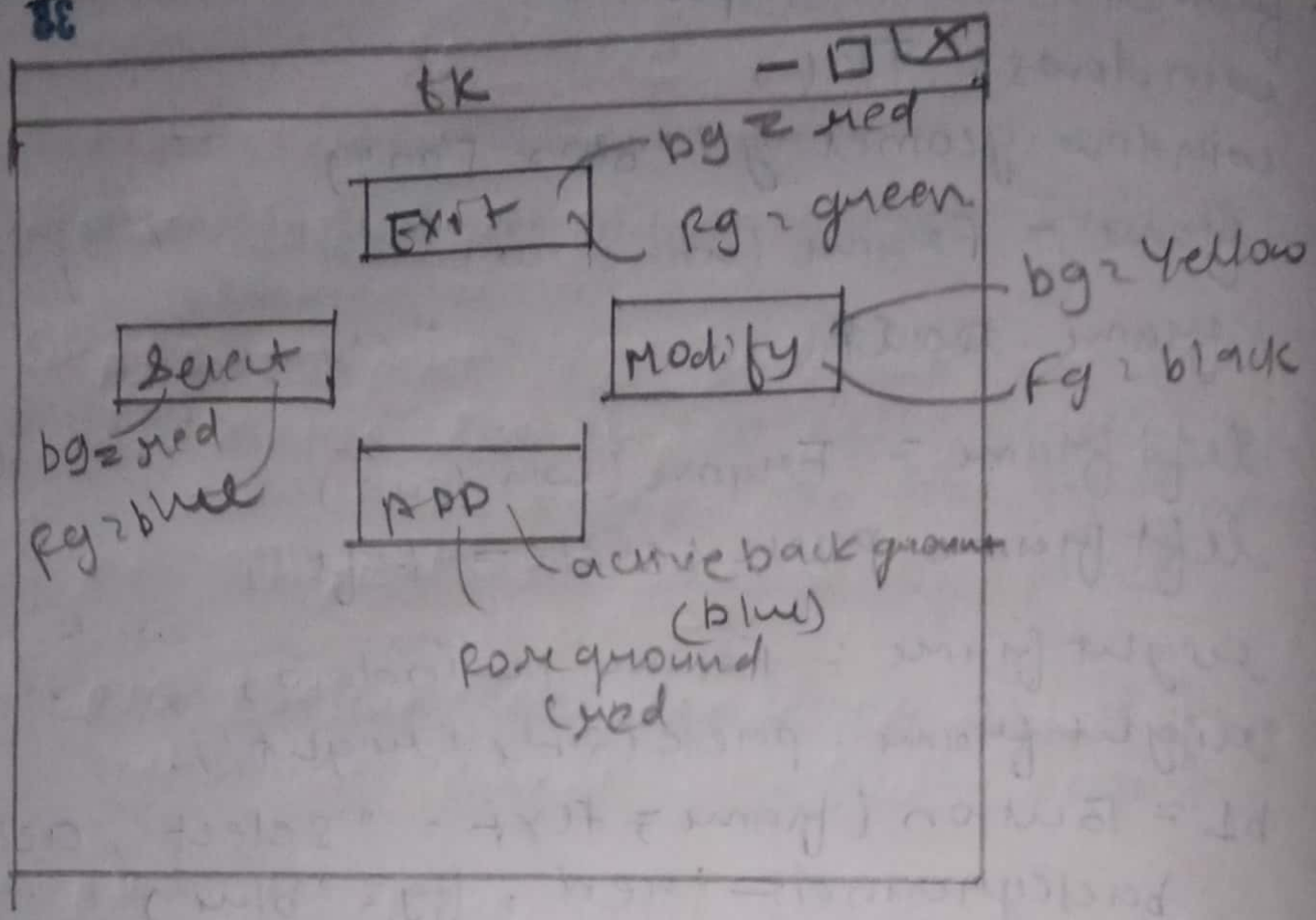
Scanned with CamScanner

```
from tkinter import *
windows = Tk()
window.geometry("680x500")
frame = Frame(window)
frame.pack().

leftframe = Frame(windows)
leftframe.pack(side="left")
rightframe = Frame(window)
rightframe.pack(side="right").
b1 = Button(frame, text = "select", active
   background="red", bg="blue")
b2 = Button(frame, text="modify", bg=
   "yellow", bg="black")
b3 = Button(frame, text = "ADD", bg="blue",
   Fg="red")
b4 = Button(frame, text="EXIT", bg =
   "red", fg="green").
b1.pack(side="LEFT", padx=20)
b2.pack(side="RIGHT", padx=30)
b3.pack(side="bottom", pady=20)
b4.pack(side="top").
```

Scanned with CamScanner

Output:



Scanned with CamScanner

Step8 :- Create another button object & place it ~~buttone~~ button as RIGHT frame & label the button as ADD.

Step9 :- Add another button and put it on the top of frame and label is as EXIST.

Step10 :- Use the pack() simultenously for a object & finally use the main loop()

Mr 21/1

PRACTICAL :- 5(C).

Aim :- GUI component.

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

Step 2 :- Import messagebox

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

Step 4 :- define a function which will use the message box with showinfo attribute

Step 5 :- Declare a button with pattern window object along with the command attribute

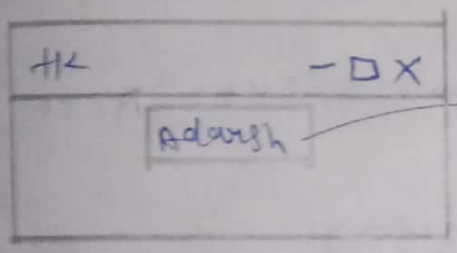
Step 6 :- place the button widget onto the parent window and finally call mainloop() for (triggering) of the event called above.

Scanned with CamScanner

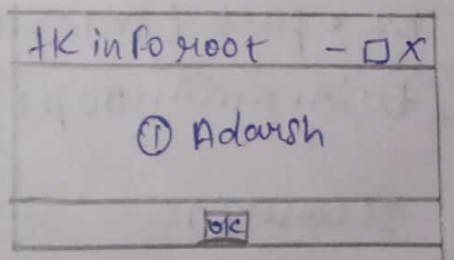
message box.

```
from Tkinter import *
import tkmessageBox
root = TK()
def function():
    tkmessageBox.showinfo("info root","python")
    b1 = Button(root, text = "Adarsh", command=
        function)
    b1.pack()
    root.mainloop()
```

Output :



click than this window will popup.



Scanned with CamScanner

#Multiple window

05 #Different button (relief)

```
From Tkinter import *
root = TK()
root.minisize(300, 300)
def main()
    win = TK()
    win.config(bg="PINK")
    win.title("Home")
    win.minisize(300, 300)
    l = Label(win, text="DIVINE \n Providence
        \n Golden coin")
    l.pack()
    b1 = Button(loop win, text="Next", command
        = second)
    b1.pack(side = RIGHT)
    b2 = Button(win, text="exit", command =
        terminate)
    b2.pack(side = LEFT)
    twin.mainloop();

def second():
    wbp = TK()
    win.config(bg = "orange")
    win2.title("Above us")
    win2.minisize(300, 300)
    l2 = Label(wbp2, text = "Created by : Adarsh
        for more detailed, contact to our
        official account")
```

Scanned with CamScanner

Step 1:- Import the relevant method from the library tkinter library along with parent window object declared.

Step 2:- Use parentwindow object along with minisize function for window size.

Step 3:- Define a function main, declare parent window object and use config() tittle(), minisize(), label() as well as button() and use pack() & mainloop() simultaneously.

Step 4:- Similarly define the function second and use the attribute accordingly.

Step 5:- Declare another function button along with parent object and declare button with attribute like FLAT, RIDGE, GROOVE, RAISED, SUNKEN along with the relief widget.

Step 6:- Finally added the mainloop(), for event driven programming.

Scanned with CamScanner

```
l.pack()
b3 = button (text = "prev", command =
  mein)
b3 . pack (side = LEFT)
b2 = button (text = "exit", command =
  terminate)
b2 . pack (side = RIGHT)
.mainloop()

def button():
    win = Tk():
    win3 . geometry ("300x300")
    b1 = button win3, text = " Flat button",
        relief = FLAT)
    b1. pack()
    b2 = button (win3, text = " groove button",
      relief = GROOVE)
    b2. pack ()
    b3 = Button (win3, text = "raised button",
      relief = RAISED)
    b3. pack()
    b4 = Button (win3, text = " sunken Button",
      relief = SUNKEN).
    b4 . pack()
```

Scanned with CamScanner

```
b5 = Button (win, text = "ridge button",
    = RIDGE)
top.mainloop()
def maximize():
    quit()
b5 = Button (root, text = "TOUR DETAILS",
    command = main)
b5.pack()
b6 = button (root, text = "Button Detail",
    command = buttons
b6.pack()
root.mainloop()
```

27/7

AIM :- GUI component

Step 1 :- Import relevant method from the tk inter library.

Step 2 :- Create parent window object and use the config method along with background color attribute specified.

Step 3 :- define a function finish with the message-box widget which will display a message ie a warning message and subsequently terminate the program.

Step 4 :- define a function info use a listbox widget along with the some use the listbox object ~~of the~~ along with input method and insert the same and finally use the grid() with external padding attribute

Step 5 :- Define a function about us ~~give~~ with label widget and text attribute and subsequently use grid us.

Scanned with CamScanner

Step6:- use photoimage widget file & filename with .gif attribute

Step7:- Create a frame object along with Frame() along with parent window Object height & width specified and subsequently use the grid method with row and column attribute specified

Step8:- Similarly create another frame object as declared debat.

Step9:- Create another object & use the sub sample (-3,4).

Scanned with CamScanner

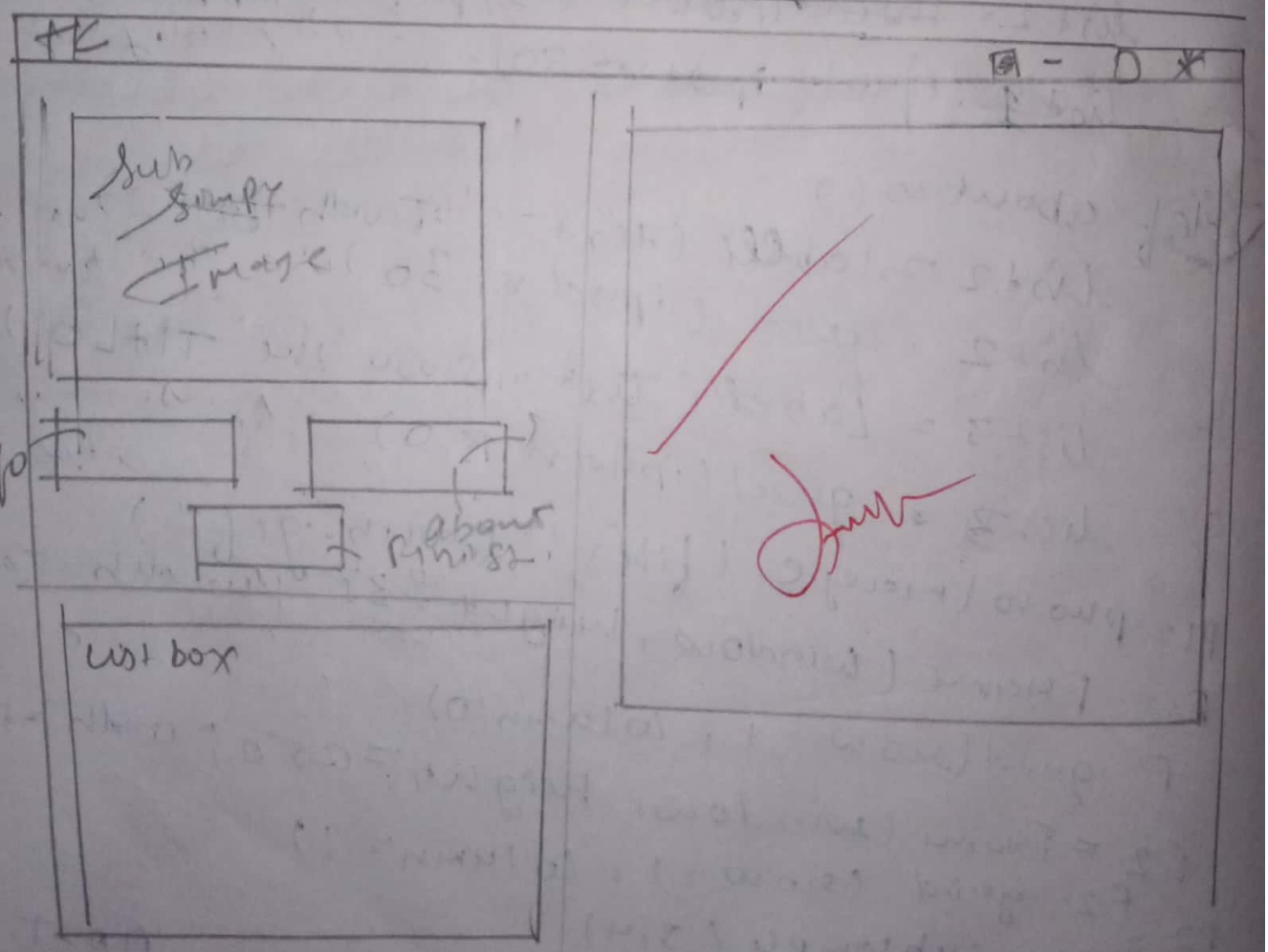
```
From tkinter import *
window = TK()
window.config(bg="green")
def finist():
    messagebox.askcancel("warning",
        "This will End the program
    quit();
def inf(0)
    list1 = listbox()
    list1.insert(1,"KULCHA")
    list1.insert(2,"MIX veges")
    list1.insert(3,"NAAN")
    list1.grid(ipadx=30).

def aboutus()
    list2 = label(text="Earth")
    list2.grid(ipadx=30)
    list3 = Label(text="Store the TITLE")
    list3.grid(ipadx=30)
p1=photoImage(file="Earth.gif")
F = Frame(window, hight=350, width=500).
F.grid(row=1, column=0)
F2 = Frame(window, hight=250, width=500)
F2.grid(row=1, column=1)
P2 = p1.subsample(3,4)
l1 = Label(f1, image=p2, relief=FLAT.
```

```
L2 = Label( F2, image = P2, relief = SUNK GM)
L2.grid (padx = 25, pady = 10)

b1 = Button( F1, text = "Informs me", relief = S
    KG M), command = info ()).
b1.grid (row1 column = 0

B2 = Button (f1, text "About", relief = Sunk
    command = aboutus)
b2.grid (row = 1, colum, padx = 5)
B3 = Button (F1, text = "QUIT", Relief = DISG
    command = finish)
B3.grid (row = 2, column = 1, Pdax = 15
window.mainloop ().
```



PRACTICAL - 5(D)

AIM:- GUI component - Spinbox(), canvas(), and panedwindow.

Spinbox:-

Step 1:- Import relevant method from tkinter library

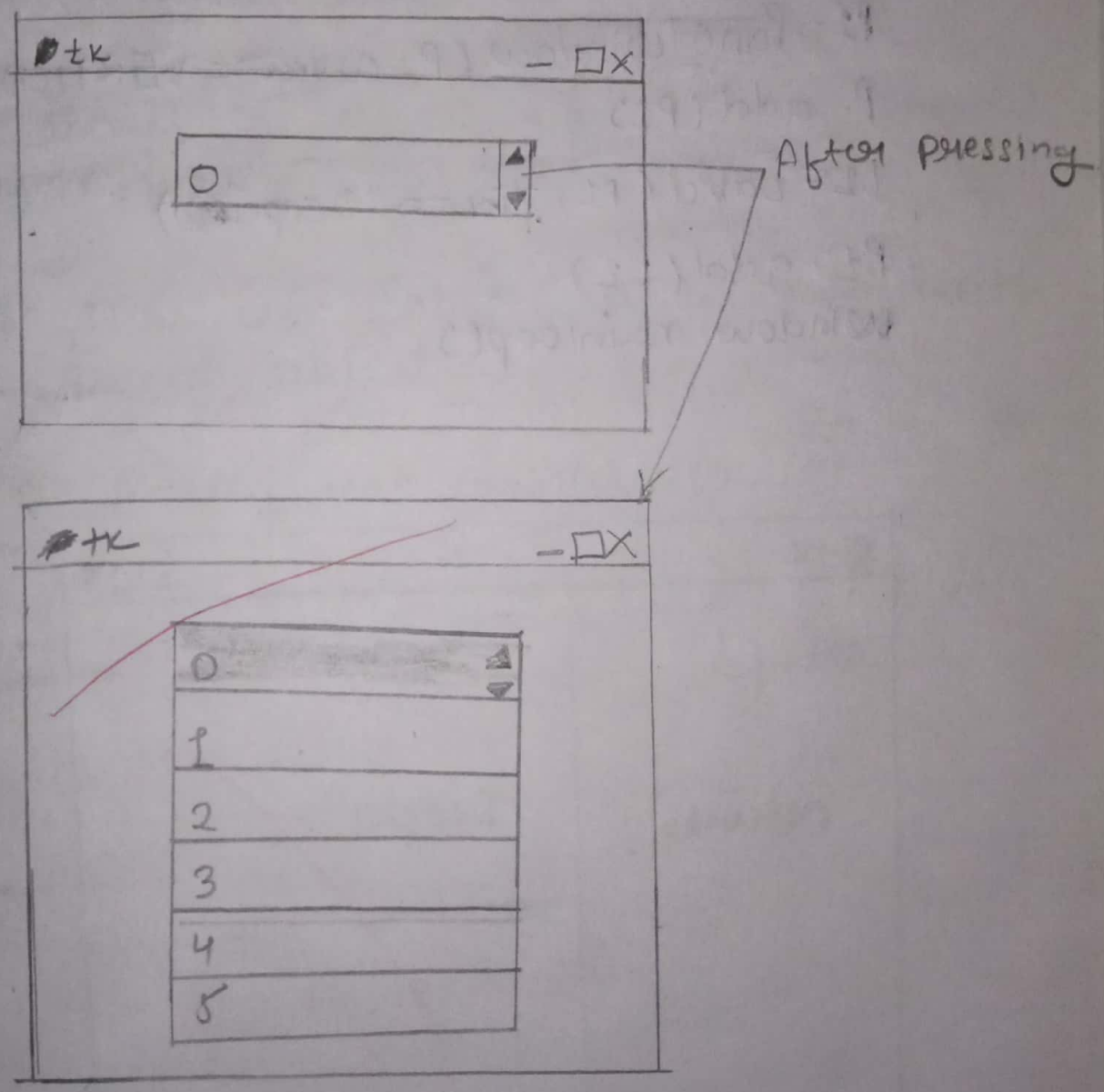
Step 2:- Create parent window object along with Tk().

Step 3:- Create an object from Spinbox() and use attribute parent window, from and to.

Step 4:- Subsequently use pack() along with Spinbox object and use the mainloop() at the end

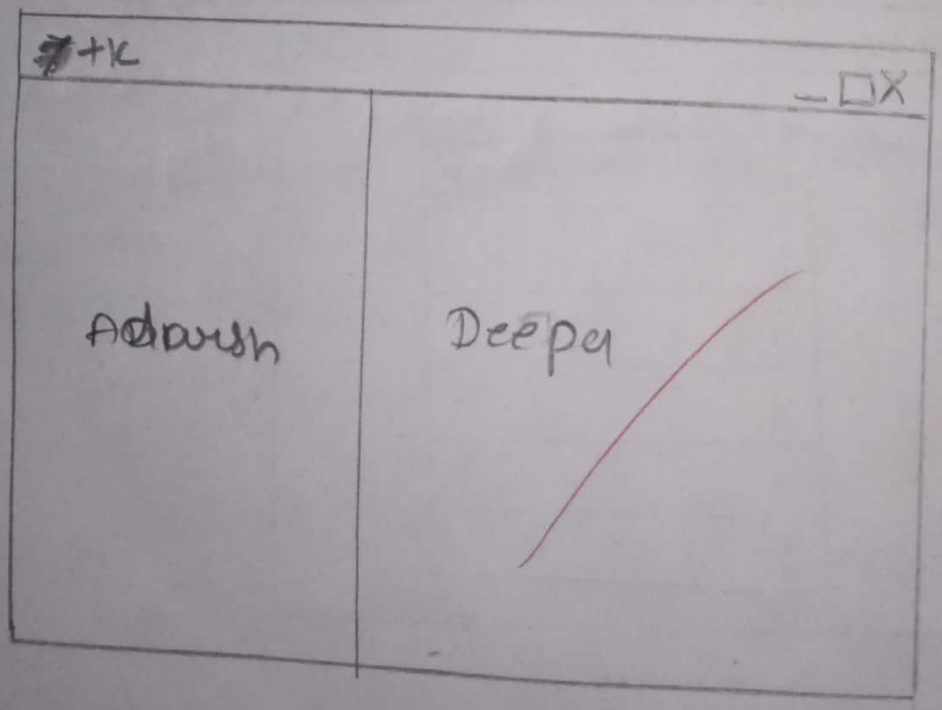
Scanned with CamScanner

~~import~~
```
From tkinter import *.
master = TK()
S= Spinbox (master, from=0 to 5)
S.pack()
Master.mainloop().
```



Scanned with CamScanner

```
import * from tkinter.
window= TK()
P= Panedwindow()
P.pack (fill= BOTH, expand=10)
L= Label(P, text = "Adarsh")
P.add(L)
P1= Panedwindow (P, orient= VERTICAL, bg="
P.add(P1)
L1= Label( P1, text= "Deepa )
P1.add(L1)
window.mainloop()
```



Panedwindow

Step 1 :- Import relevant Method from tkinter library.

Step 2 :- Create an object of parent window.

Step 3 :- Create an object of panedwindow() and subsequently use pack().

Step 4 :- Now, use label() and add with panned object.

Step 5 :- Finally use mainloop().

Amin

Canvas:

Step 1 :- Import relevant method from tkinter library.

Step 2 :- Create an object of parent window using TK()

Step 3 :- Create an object of Canvas().

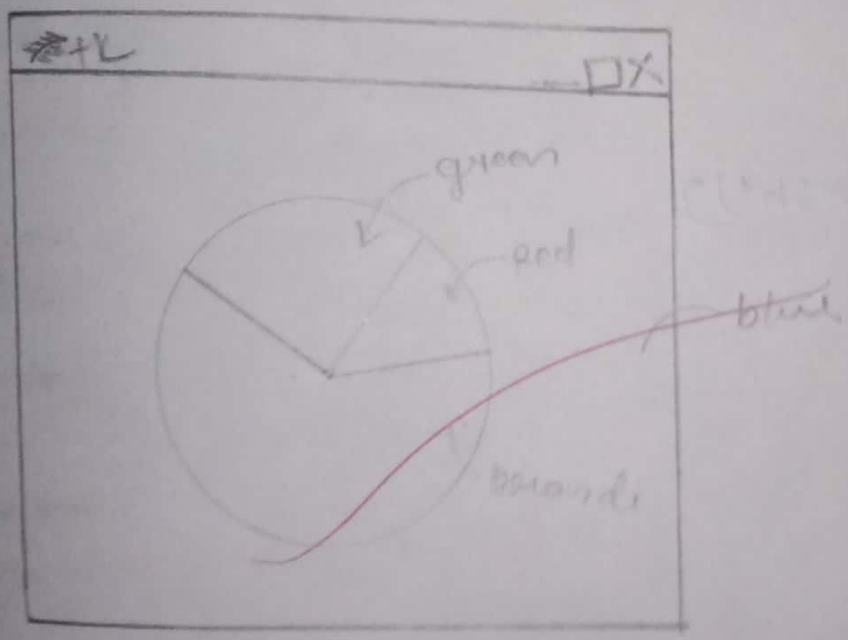
Step 4 :- Create 2 co-ordination

Step 5 :- Use create_arc() with the object of Canvas().

Step 6 :- Subsequently use pack() and mainloop()

Scanned with CamScanner

```
from tkinter import *
window = Tk()
c = canvas (window, bg "blue", height=250, width=300)
coord = 10, 50, 240, 210
arc = c.create_arc (coord, start=0, extent=100,
        fill="red")
arc = c.create_arc (coord, start=150, extent=210,
        fill="orange")
arc = c.create_arc (coord, start=50, extent=100,
        fill="green")
c.pack()
window.mainloop()
```



Jn 2112

Source code

```
import os, sqlite3.
con = sqlite3.connect("student.db")
c = con.cursor
c.execute("create table student(role INTEGER,
                name TEXT)")
c.execute("INSERT INTO student value (1739,
                "Adarsh"),
            (1736, "ROWA"),(1703, "Nilesh")")
c.execute("SELECT * from student WHERE rol
                1739")

Print(c.fetchall())
con.commit()
c.close
```

output

```
[(1739, 'Adarsh')]
```

Scanned with CamScanner

PRATICAL 6

AIM :- Database connectivity.

Step1 :- Import corresponding library to make data base connection, os and sqlite.

Step2 :- Now, create the connection object using sqlite3 library and the connect() for creating new database.

Step3 :- Now, create cursor object using the cursor() and from the connection object created.

Step4 :- Now, use the execute() for creating the table with the column name and respective data type.

Step5 :- Now with cursor object use the insert statement for entering the value corresponding to different field, corresponding to the data type.

Step6 :- Use the commit() to complete the transaction using the connection object.

Step7 :- Use the execute statement along with cursor object for accessing the value from data base using the select from where clause.

Scanned with CamScanner

Step 8:- Finally use fetch() or fetchall() and print the value from the table using

Step 9:- Finally close() the cursor object

```
from tkinter import *
import time
import random
import sqlite3

root=Tk()
root.geometry("1600x800")
root.title("Restaurant Management System")

text_Input=StringVar()
operator=" "

f1=Frame(root,width=1600,height=50,bg='powder blue',relief=SUNKEN)
f1.pack(side=TOP)
f2=Frame(root,width=800,height=700,relief=SUNKEN)
f2.pack(side=LEFT)
f3=Frame(root,width=300,height=700,relief=SUNKEN)
f3.pack(side=RIGHT)

#======================TIME==================================================
=#
localtime=time.asctime(time.localtime(time.time()))
#======================INFORMATION=========================================
#

lbl=Label(f1, font=("arial",50,"bold"), text="Restaurant Management
System", fg="green", bd=5, anchor='w')
lbl.grid(row=0,column=0)
lbl=Label(f1, font=("arial",20,"bold"), text=localtime, fg="steel blue",
bd=5, anchor='w')
lbl.grid(row=1,column=0)

E1=Entry(f3,font=("arial",20,'bold'),textvariable=text_Input, bd=30,
insertwidth=4,bg='powder blue',justify='right')
E1.grid(columnspan=4)

#========================CALCULATOR=====================================#

def btnClick(number):
    global operator
    operator+=str(number)
    text_Input.set(operator)

def Clearbutton():
    global operator
    operator+=""
    text_Input.set("")
def Equaltobtn():
    global operator
    sumup=str(eval(operator))
    text_Input.set(sumup)
    operator+=sumup
```

Scanned with CamScanner

```
btn7=Button(f3,padx=16,pady=16, fg='black',bg='powder
blue',font=("arial",20,"bold"),
            text='7',command= lambda: btnClick(7)) .grid(row=2,column=0)
btn8=Button(f3,padx=16,pady=16, fg='black',bg='powder
blue',font=("arial",20,"bold"),
            text='8',command= lambda: btnClick(8)) .grid(row=2,column=1)
btn9=Button(f3,padx=16,pady=16, fg='black',bg='powder
blue',font=("arial",20,"bold"),
            text='9',command= lambda: btnClick(9)) .grid(row=2,column=2)
Addition=Button(f3,padx=16,pady=16, fg='black',bg='powder
blue',font=("arial",20,"bold"),
            text='+',command= lambda:
btnClick("+")) .grid(row=2,column=3)
btn4=Button(f3,padx=16,pady=16, fg='black',bg='powder
blue',font=("arial",20,"bold"),
            text='4',command= lambda: btnClick(4)) .grid(row=3,column=0)
btn5=Button(f3,padx=16,pady=16, fg='black',bg='powder
blue',font=("arial",20,"bold"),
            text='5',command= lambda: btnClick(5)) .grid(row=3,column=1)
btn6=Button(f3,padx=16,pady=16, fg='black',bg='powder
blue',font=("arial",20,"bold"),
            text='6',command= lambda: btnClick(6)) .grid(row=3,column=2)
substraction=Button(f3,padx=16,pady=16, fg='black',bg='powder
blue',font=("arial",20,"bold"),
            text='-',command= lambda: btnClick('-
')) .grid(row=3,column=3)
btn1=Button(f3,padx=16,pady=16, fg='black',bg='powder
blue',font=("arial",20,"bold"),
            text='1',command= lambda: btnClick(1)) .grid(row=4,column=0)
btn2=Button(f3,padx=16,pady=16, fg='black',bg='powder
blue',font=("arial",20,"bold"),
            text='2',command= lambda: btnClick(2)) .grid(row=4,column=1)
btn3=Button(f3,padx=16,pady=16, fg='black',bg='powder
blue',font=("arial",20,"bold"),
            text='3',command= lambda: btnClick(3)) .grid(row=4,column=2)
multiplication=Button(f3,padx=16,pady=16, fg='black',bg='powder
blue',font=("arial",20,"bold"),
            text='*',command= lambda:
btnClick('*')) .grid(row=4,column=3)
btn0=Button(f3,padx=16,pady=16, fg='black',bg='powder
blue',font=("arial",20,"bold"),
            text='0',command= lambda: btnClick(0)) .grid(row=5,column=0)
equalto=Button(f3,padx=16,pady=16, fg='black',bg='powder
blue',font=("arial",20,"bold"),
            text='=',command= Equaltobtn) .grid(row=5,column=1)
clearbtn=Button(f3,padx=16,pady=16, fg='black',bg='powder
blue',font=("arial",20,"bold"),
            text='C',command= Clearbutton) .grid(row=5,column=2)
division=Button(f3,padx=16,pady=16, fg='black',bg='powder
blue',font=("arial",20,"bold"),
            text='/',command= lambda:
btnClick('/')) .grid(row=5,column=3)
```

Scanned with CamScanner

```
#-------------------------------------------------------------------------------
---------------------------------------------------------------------------------
---------------------
reference=StringVar()


lb2=Label(f2,text="Reference: ",fg="black",font=("arial",16,"bold"),
bd=20, anchor='w') .grid(row=0,column=0)
E2=Entry(f2,font=("arial",16,'bold'),textvariable=reference, bd=16,
insertwidth=2,bg='powder blue', justify= 'right')
E2.grid(row=0,column=1)

Fries=StringVar()
lb3=Label(f2,text="Fries",fg="black",font=("arial",16,"bold"), bd=20,
anchor='w') .grid(row=1,column=0)
E3=Entry(f2,font=("arial",16,'bold'),textvariable=Fries, bd=16,
insertwidth=2,bg='powder blue', justify= 'right')
E3.grid(row=1,column=1)

Burger=StringVar()
lb4=Label(f2,text="Burger Meal",fg="black",font=("arial",16,"bold"),
bd=20, anchor='w') .grid(row=2,column=0)
E4=Entry(f2,font=("arial",16,'bold'),textvariable=Burger, bd=16,
insertwidth=2,bg='powder blue', justify= 'right')
E4.grid(row=2,column=1)

Filet=StringVar()
lb5=Label(f2,text="Filet_o_Meal",fg="black",font=("arial",16,"bold"),
bd=20, anchor='w') .grid(row=3,column=0)
E5=Entry(f2,font=("arial",16,'bold'),textvariable=Filet, bd=16,
insertwidth=2,bg='powder blue', justify= 'right')
E5.grid(row=3,column=1)

Chicken_Burger=StringVar()

lb6=Label(f2,text="Chicken Meal",fg="black",font=("arial",16,"bold"),
bd=20, anchor='w') .grid(row=4,column=0)
E6=Entry(f2,font=("arial",16,'bold'),textvariable=Chicken_Burger, bd=16,
insertwidth=2,bg='powder blue', justify= 'right')
E6.grid(row=4,column=1)

Cheese_Burger=StringVar()

lb8=Label(f2,text="Cheese Meal",fg="black",font=("arial",16,"bold"),
bd=20, anchor='w') .grid(row=5,column=0)
E8=Entry(f2,font=("arial",16,'bold'),textvariable=Cheese_Burger, bd=16,
insertwidth=2,bg='powder blue', justify= 'right')
E8.grid(row=5,column=1)

#-----------------------------------------------------Restaurant Info-2-----
---------------------------------------------------------------------------------
----------------------------

Drinks=StringVar()
```

Scanned with CamScanner

```
lb9=Label(f2,text="Drinks",fg="black",font=("arial",16,"bold"), bd=20,
anchor='w') .grid(row=0,column=2)
E9=Entry(f2,font=("arial",16,'bold'),textvariable=Drinks, bd=16,
insertwidth=2,bg='white', justify= 'right')
E9.grid(row=0,column=3)

Cost=StringVar()
lb10=Label(f2,text="Cost of Meal",fg="black",font=("arial",16,"bold"),
bd=20, anchor='w') .grid(row=1,column=2)
E10=Entry(f2,font=("arial",16,'bold'),textvariable=Cost, bd=16,
insertwidth=2,bg='white', justify= 'right')
E10.grid(row=1,column=3)

Service_Charge=StringVar()
lb11=Label(f2,text="Services Charge",fg="black",font=("arial",16,"bold"),
bd=20, anchor='w') .grid(row=2,column=2)
E11=Entry(f2,font=("arial",16,'bold'),textvariable=Service_Charge, bd=16,
insertwidth=2,bg='white', justify= 'right')
E11.grid(row=2,column=3)

Tax=StringVar()
lb12=Label(f2,text="State Tax",fg="black",font=("arial",16,"bold"),
bd=20, anchor='w') .grid(row=3,column=2)
E12=Entry(f2,font=("arial",16,'bold'),textvariable=Tax, bd=16,
insertwidth=2,bg='white', justify= 'right')
E12.grid(row=3,column=3)

Sub_Total=StringVar()

lb13=Label(f2,text="Sub Total",fg="black",font=("arial",16,"bold"),
bd=20, anchor='w') .grid(row=4,column=2)
E13=Entry(f2,font=("arial",16,'bold'),textvariable=Sub_Total, bd=16,
insertwidth=2,bg='white', justify= 'right')
E13.grid(row=4,column=3)

Total=StringVar()

lb15=Label(f2,text="Total Cost",fg="black",font=("arial",16,"bold"),
bd=20, anchor='w') .grid(row=5,column=2)
E15=Entry(f2,font=("arial",16,'bold'),textvariable=Total, bd=16,
insertwidth=2,bg='white', justify= 'right')
E15.grid(row=5,column=3)

#==========================Buttons==================================#
def ref():
    x=random.randint(213451,512637)
    randreference=str(x)
    reference.set(randreference)

    CoF=float(Fries.get())
    CoD=float(Drinks.get())
    CoFilet=float(Filet.get())
    CoBur=float(Burger.get())
```

Scanned with CamScanner

```
    CoC=float(Cheese_Burger.get())
    CoChicken=float(Chicken_Burger.get())

    CostofFries= CoF * 120.00
    CostofDrinks= CoD * 70.00
    CostoFilet= CoFilet * 199.00
    Costofburger= CoBur * 69.00
    Costof_cheese_burger= CoC * 99.00
    Costof_Chicken_burger= CoChicken * 149.00

    CostofMeal= "Rs",str( "%.2f" % (CostofFries+CostofDrinks+
CostoFilet+Costofburger+Costof_cheese_burger+Costof_Chicken_burger))

    PayTax= ((CostofFries+CostofDrinks+
CostoFilet+Costofburger+Costof_cheese_burger+Costof_Chicken_burger)* 0.2)

    TotalCost=(CostofFries+CostofDrinks+
CostoFilet+Costofburger+Costof_cheese_burger+Costof_Chicken_burger)

    Ser_Charge= ((CostofFries+CostofDrinks+
CostoFilet+Costofburger+Costof_cheese_burger+Costof_Chicken_burger) /99)

    Service= "Rs", str('%.2f' % Ser_Charge)
    OverAllCost=("Rs",str( "%.2f " % (PayTax + TotalCost + Ser_Charge)))
    PaidTax="Rs",str( "%.2f" % PayTax)
    Service_Charge.set(Service)
    Cost.set(CostofMeal)
    Tax.set(PaidTax)
    Sub_Total.set(CostofMeal)
    Total.set(OverAllCost)
def quit():
    root.destroy()

def reset():
    reference.set("")
    Fries.set("")
    Drinks.set("")
    Chicken_Burger.set("")
    Cheese_Burger.set("")
    Filet.set("")
    Sub_Total.set("")
    Total.set("")
    Service_Charge.set("")
    Tax.set("")
    Cost.set("")
    Burger.set("")


Totalbtn=Button(f2,text="
Total",font=("arial",16,"bold"),height=2,width=14,bd=16,anchor='w',bg="po
wder blue",command=ref) .grid(row=7,column=0)
restbtn=Button(f2,text="
Reset",font=("arial",16,"bold"),height=2,width=14,bd=16,anchor='w',bg="po
wder blue",command=reset) .grid(row=7,column=1)
```

Scanned with CamScanner

```
quitbtn=Button(f2,text="
Quit",font=("arial",16,"bold"),height=2,width=14,bd=16,anchor='w',bg="pow
der blue",command=quit) .grid(row=7,column=2)
#-------------=-----------------------------------------------
Receipt(database)-------------------------------------------------------
--------------------------------------


def print_bill():
     refer= reference.get()
     fries=Fries.get()
     drink=Drinks.get()
     chicken_burger=Chicken_Burger.get()
     filet=Filet.get()
     sub_total=Sub_Total.get()
     total=Total.get()
     service_charge=Service_Charge.get()
     tax=Tax.get()
     cost=Cost.get()
     burger=Burger.get()
     conn= sqlite3.connect('restaurant.db')
     with conn:
         cursor=conn.cursor()
    #cursor.execute('CREATE TABLE IF NOT EXISTS Bill (reference INT,
Fries INT, Drinks INT, Chicken_Burger INT, Filet INT, Sub_Total INT,
Total INT, Service_Charge INT, Tax INT, Cost INT, Burger INT)' )
     cursor.execute('INSERT INTO Bill (reference, Fries, Drinks,
Chicken_Burger,Filet, Sub_Total, Total, Service_Charge, Tax, Cost,
Burger) VALUES (?,?,?,?,?,?,?,?,?,?,?)' ,

(refer,fries,drink,chicken_burger,filet,sub_total,total,service_charge,
tax,cost,  burger))
     conn.commit()
     conn.close()

printbtn=Button(f2,text="          Print
Bill",font=("arial",16,"bold"),height=2,width=14,bd=16,anchor='w',bg="pow
der blue",command=print_bill) .grid(row=7,column=3)



root.mainloop()
```

Scanned with CamScanner

Scanned with CamScanner

```
Python 3.7.5 (tags/v3.7.5:5c02a39a0b, Oct 15 2019, 00:11:34) [MSC v.1916 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
================== RESTART: C:\Users\Admin\Desktop\restaurant.py ==================
491943   1          5          8          4          ('Rs', '4051.00')     ('Rs', '4902.12 ')     ('Rs', '40.92')        ('Rs', '810.20')       ('Rs', '40
51.00')   3
```

Scanned with CamScanner