

```

No. 7
file obj = open("abc.txt", "w") → 6
file obj . write ("Python is intended language\n")
file obj . close ()
file obj = open("abc.txt", "r")
str1 = file obj . read ()
print ("the output of read method : ", str1)
file obj . close ()

No. 8
file obj = open("abc.txt", "r")
str2 = file obj . readline ()
print ("The output of readline method : ", str2)
output :-  

>>>
>>> the output of read method : Python is
intended language
>>> the output in readline method : /, ['Python is
intended language\n']

```

25

### Practical No. 1

- Aim : Demonstrate the ~~use of file~~ managing file accessing with different attribute
- Program
1. Create a file obj by use open Method can use a ~~written~~ a ~~new~~ mode followed by writing some content to file and then closing a file
  2. Now open a file in read mode and use read method or read line method and store the variable and finally display the content of variable
  3. now use the ~~file obj~~ for finding the main of file mode in which is being open, whether the file listed open or close & finally output of softspace attribute

- now open file object in write mode & write some another content & close subsequently & then again open obj in 'w+' mode that is update mode & write a content
- open file obj in read mode with parameter for update written content close & open again in 'r+' mode with parameter passed & display the output.
- now open file obj in append mode open corite method & corite content close the file obj again open file object in read mode & display the append output

```

a = file_obj.name
print("Name of file : " + a)
>>> Name of file :- abc.txt

b = file_obj.closed
print("closed attribute : " + b)
>>> closed attribute : -true

c = file_obj.mode()
print("file mode : " + c)
>>> file mode : r

d = file_obj.softspace
print("softspace : " + d)
>>> softspace : 0

# w+ mode
file_obj = open("abc.txt", "w+")
file_obj.write("Python is case Sensitive")
file_obj.close()

file_obj = open("abc.txt", "r+")
str1 = file_obj.read()
print("the output of r+ : " + str1)
file_obj.close()

>>> (the output of r+ : the Python is case sensitive)

```

```

file obj = open("abc.txt", "r+")
strg = file_obj.read(10)
print(strg)
pos = file_obj.tell()
print(pos)
strg = file_obj.seek(0, 0)
print(strg)
>>> the Python is
>>> 10
>>> node
.
.
.
with open("abc.txt", "r+") as g:
    s = g.readline()
    c = g.read(1)
    while len(c) > 0:
        print(c, end="*")
        c = g.read(1)
>>> p*V*H*Dn*i*S*x*inten*te*kl*on*ng*re
>>> fileObj = open("abc.txt", "r")
    c = fileObj.read(1)
    str1 = fileObj.read(10)
    for item in str1:
        print(item)

```

- open a file `obj` in read mode, declare a variable `strg` to perform a `file_obj.read()` method & store the value
- Use `tell` method to find position of pointer in file then display the position by `seek` method with arg by mention in syntax
- Using `file` operation for reading individual statement & inserting some character of content file
- Open a file by using `with` in beginning & give variable as `g`
- use while loop with length of file & display the ans

## Practical No. 2

Aim: The Uses of Iteration & Iterable Method

Program :-

To know how iter & next()

work

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

Create a tuple & use iter & next method to display first & value

for accessing the individual value for given obj. we can use for loop

\* WAP using Iterable method to display set of first 20 numbers

→ define the iter method with arg & initialize to first value

→ for extracting next element from container use next method with arg & compare no of element required in container by conditional Statement

x program:-

• Mytuple = ("banana", "orange", "apple")

myiter = iter(mytuple)

Print(next(mytuple))

⇒ banana

• For i in mytuple:

Print(i)

⇒ banana

⇒ orange

⇒ apple

\* Program -

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)

- now create an object from given class & pass this object as arg to other method
- now use conditional display all value from Container
- + write a program to return weather no. is odd or even from given set
- Map function takes two argument one is a function and another sequence which is iterable
- the output of map function return the list data type
- first take the enter the set of no. random
- now use map function which is modulus divide by 5
- then def even function which return the even number using conditional statement modular division
- the divide by it with 2  
else it is odd number

\* program

```

list num = [0, 5, 7, 8, 9, 11, 15, 20, 29]
list num = list(map(lambda x: x%5, list num))

def even(x):
    if x % 2 == 0:
        return "Even"
    else:
        return "Odd"

list(map(even, list num))

```

```

>>> ['Even', 'Odd', 'Even', 'Odd', 'Even', 'Odd',
     'Even', 'Even', 'Even']

```

- write a program to print square & cube of given set of no. using method
- defining a function square which return the value in square.
- defining a function cube which return the cube value.
- function return value like one or will be square of no. & other such cube define a function one as list square, comma cube both in square bracket
- defining a range where you want square & cube of numbers
- value out is equal to map function with range now print the function f by list

Program :-

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

```
>>> [1, 1]
[2, 8]
[3, 27]
[4, 64]
[5, 125]
```

## Program :-

Class odd :

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

```
def next(self):
```

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

```
num = self.num
```

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

```
return num
```

```
my_obj = odd()
```

```
x = int(input("Enter the number"))
```

```
my_iter = iter(my_obj)
```

```
for i in my_iter:
```

```
    if i < num:
```

```
        print(i)
```

```
>> Enter the number: 6
```

```
1  
3  
5
```

WAP to display set of odd no.

→ defining a class & within it define the iter method which initialized the first element, which contain object

→ now use the next method & define logic for displaying the odd

The code will go an exception, i.e. object are not able to traverse to define ex. elements

→ take the input where of range to display set of odd no.

→ use for loop conditional statement if should be greater than num in input

To find the square of numbers without map function

Define a list which contains a value

Define an empty list

Use for loop followed by append method to append the result into list

Print the value of list

list = [1, 2, 8, 9, 10]

empty = []

for i in list:

\* empty.append(i \*\* 2)

print(empty)

Output

» output

empty = [1, 4, 64, 81, 100]

J-16/12

No. 1

F G

Except IOError:

Print("there is environment error")

Else:

Print("operation is successful")

Operation is successful



35

### Practical No.3

\* write a program using exception block related to environment error

- i use try block to define the normal course of action for example :- to define a file obj & open a file in write mode write mode & write some content in file
- ii use except block with I/O error & environment & convey the appropriate message to user else display operation carry out successfully

- 23
- write a program to demonstrate multiple operator with IO Error & value error

step 1: accept the value from the user if it is valid  
value display the entered value & terminate the  
condition by using break statement

step 2: define a except block with value errors as  
keyword & display the appropriate message

step 3: we can define multiple exception using  
except block & finding the different category of  
errors

try:

```
x = int(input("Enter the number"))
print(x, "value is correct")
break
```

except ValueError:
 print("The value is invalid")
except IOError:
 print("There is an environment error")

### output

Enter the number: abc  
The value is invalid

Just

### \* Programs:

```
Import re  
Pattern = r'FYCS'  
Sequence = 'FYCS represent the Computer Science Stream'  
if re.match(Pattern, Sequence):  
    print("match Pattern")  
else:  
    print("not found")
```

Output : Match Pattern

### \* Programs

### Practical No. 4

37

a) WAP to demonstrate the match method

Step 1: import the re library ~~match~~ method for the common usage of regular expression which involve the match, search or etc.

Step 2: Enter the pattern which you want by putting & in forward in pattern which you want

Step 3: Enter the sequence or string  
Step 4: use if loop statement to use the match method that commonly take a argument from user. If given pattern is matched with sequence then print the appropriate message that match found else display match not founded

- WAP using / script program with split method
- & display the appropriate output to user

Q. write a regular expression for segregate the alpha beta value from numeric values in given string

Step 1: Import re method for a common usage of regular expression which involve the.findall, match or replace

Step 2: write the regular expression you want as enter the both 3 characters as well numeric value

Step 3: Enter the pattern as '\d+' & it will only return only numeric value

Step 4: use.findall method with \d+ in front with re will take two argument as Pattern & Sequence & display the output

- Now, using same program using re.split method & display output to user.

#### \* program

```
import re
sequence = 'hello123, howdy789, hhowwo'
pattern = r'\d+'
output = re.findall(pattern, sequence)
print(output)
```

Output  
 >>> ['123', '789', '5']

#### \* Program

```
import re
seq = 'hello123, howdy789, h5howwo'
output = re.split(r'\d+', seq)
print(output)
```

Output

>>> 'hello', 'howdy', 'h5howwo'

& write a regular expression for removing white spaces from String

- Step 1: import re method for a common usage of replace & sub methods
- Step 2: enter the string with blank spaces
- Step 3: Enter the pattern as \s+ in front of \ black slash S+ it mean that it take a look on blank spaces
- Step 4: ~~then~~ take replace as variable with no blank spaces
- Step 5: now, use this sub method that will take replace all blank spaces to no spaces in a given string & display output to user

```
No. 18
import re
sequence = 'abc def ghi'
pattern = r'\s+'
replace = ''
v1 = re.sub(pattern, replace, sequence)
print(v1)
```

Output  
abcdefghi

- No.
- Ques.
- write a regular expression for a host & domain name
  - Import 're' method for common usage of regular expression which involve the find all, replace, match method
  - Enter the sequence as to segregate host name to domain name
  - the pattern to be used include the space with `\w` it represent the match with alpha numeric character appropriate
  - Display the output to user

\* Program

```
import re
sequence = "Please connect to : abc@gmail.com, de
def@gmail.com"
pattern = '[\w\.-]+@[ \w\.-]+'
output = re.findall(pattern, sequence)
print(output)
```

`['abc@gmail.com', 'def@gmail.com']`

- i write a regular expression finding string given sequence using search method
- ii import re method for common usage of regular expression which invoke the `re.findall`, `re.replace`, `re.search` method
- iii enter the sequence which you want
- iv using pattern `\A` which you want to search in string
- v use the `re.search` method for searching the string and display this output to user

```
import re
str1 = ['Python is intended language']
output = re.findall(r'\b[aeiou]\w+', str1)
print(output)
```

[is', 'intended']

Ans

- Q write a regular expression to find a string in a given paragraph with words which start with vowel
- import re method for common usage of regular expression which involve the find all, replace, & search method.
- Enter the string which u want
- use find all method with first word pattern along with
- display the output to user otherwise

```

Import re
list1 = ['8004564189', '8282649485', '7865432109']
for i in list1:
    for value in i:
        if re.match(r'[5-9]{1}[0-9]{9}', value or len(value) == 10):
            print("no. matched")
        else:
            print("no. not matched")

```

» no. not matched

- i) Import re method for common usage  
→ findallString a method, search & match method.
- ii) to enter the list of phone no. which you want
- iii) use for loop for loop to see the value in list & inside for loop use if statement & match the given criteria by using regular expression by match method
- iv) if no. is matched display the output to user else display no. is not matched

## Practical No. 5

- \* Topic :- Basic command in GUI
- \* Basic Body & commands that you always use in GUI
- i Tkinter :- import the Tkinter library  
to import this we use `from Tkinter import *`  
→ from Tkinter import \*
- Tkinter module needed to build Tkinter widget
- You have to import Tkinter module in each program
- ii `root = Tk()`  
• by `Tk()` we generate parent window  
• for each program, we have to create only one parent window
- iii `root.mainloop()` command is used to launch the window & start the window event loop.

Basic source code

```
from Tkinter import *
root = Tk()
body of gui
root.mainloop()
```

### Algorithm

```
from Tkinter import *
root = Tk()
T1 = Text(root)
T1.insert(END, "The Python is intended language")
T1.pack(side=TOP, padx=20, pady=20, ipadx=40
       ipady=50)
L = Label(root, text="label", bg="white"
          fg="blue")
L.pack(side=LEFT, padx=10, ipadx=20, ipady=30)
root.mainloop()
```

45

To make use of GUI application along with basic pack method

- algorithm
  - Use Tkinter library for importing the feature of text object.
  - Create a variable from text variable & position onto parent window
  - use of pack method along with object created from text method & use parameter
  - use the main loop method to triggering the event by use of parameter
    - i] Side = top padx=20 ipadx=40
    - ii] ipady=50
  - Now repeat above step with label method which takes following argument
    - i] name of parent window
    - ii] The text attribute which define string
    - iii] The background colour (bg)
    - iv] The foreground colour fg

JRT

2 To make use of RadioButton widget for selection one of multiple option

Step1: Use Tkinter library to import relevant method

Step2: Define a function which tell user about given Selection from multiple option available

Step3: Use config method along with label method & call function as arg within method

Step4: now def a parent window & def option using control = variable

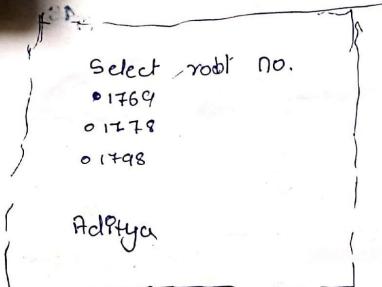
Step5 now Create a obj of Radio button will take follow arg  
i) Partition parent window  
ii) Text variable  
iii) Def val arg  
iv) Corresponding value will triggered

\$ also call pack method for radio obj

18

```
from Tkinter import *
def Sel1():
    Selection = "Aditya"
    Label.config(text=selection)
def Sel2():
    Selection = "John"
    Label.config(text=selection)
def Sel3():
    Selection = "Sachin"
    Label.config(text=selection)
root = Tk()
var = IntVar()
L1 = Label(root, text="Select any roll no.")
L1.pack(side=TOP)
R1 = Radiobutton(root, text="1764", variable=var, value=1, command=Sel1)
R1.pack(anchor=N)
R2 = Radiobutton(root, text="1761", variable=var, value=2, command=Sel2)
R2.pack(anchor=N)
R3 = Radiobutton(root, text="1798", variable=var, value=3, command=Sel3)
R3.pack(anchor=N)
Label = Label(root)
Label.pack()
root.mainloop()
```

Jan 12



- 47
- a) Write a program to make use of scroll bar widget of GUI application
    - Import tkinter library to use scroll bar widget
  - Create an object corresponding to scroll parent window & create an object from scroll bar & place it onto parent window so created
  - Create an object of label method to provide a heading & place it on parent window
  - Use ~~parent~~ <sup>Pack</sup> window along with object & use argument side & fill
  - Create an object of list box method and place it onto parent window
  - Step Use for loop to insert value in object of list box by using insert method

- Use config method along with scroll bar object & use command attribute
- Finally use main loop method

```

from Tkinter import *
root = Tk()
root.geometry("400x400")
l = Label(root, text="B batch Roll No.", bg="white",
          fg="blue")
l.pack()
s = Scrollbar(root)
s.pack(side=RIGHT, fill=Y)
mylist = Listbox(root, yscrollcommand=s.set,
                 fg="lightblue")
for num in range(1741, 1780):
    mylist.insert(END, "* Roll No.: " + str(num))
mylist.pack(side=LEFT, fill=BOTH)
scroll.config(command=mylist.yview)
root.mainloop()

```

Output

The image shows a window titled "B Batch Roll NO." containing a listbox with the following items:

- \* Roll No. 1741
- \* Roll No. 1742
- \* Roll No. 1743
- \* Roll No. 1744
- \* Roll No. 1745
- \* Roll No. 1746
- \* Roll No. 1747
- \* Roll No. 1748
- \* Roll No. 1749
- \* Roll No. 1750
- \* Roll No. 1751
- \* Roll No. 1752
- \* Roll No. 1753
- \* Roll No. 1754
- \* Roll No. 1755
- \* Roll No. 1756
- \* Roll No. 1757
- \* Roll No. 1758
- \* Roll No. 1759
- \* Roll No. 1760
- \* Roll No. 1761
- \* Roll No. 1762
- \* Roll No. 1763
- \* Roll No. 1764
- \* Roll No. 1765
- \* Roll No. 1766
- \* Roll No. 1767
- \* Roll No. 1768
- \* Roll No. 1769
- \* Roll No. 1770
- \* Roll No. 1771
- \* Roll No. 1772
- \* Roll No. 1773
- \* Roll No. 1774
- \* Roll No. 1775
- \* Roll No. 1776
- \* Roll No. 1777
- \* Roll No. 1778
- \* Roll No. 1779
- \* Roll No. 1780

```

From tkinter import *
root = Tk()
root.config(bg="grey")
root.geometry("500x500")
def finish():
    messagebox.askokcancel("warning this will end program")
def quit():
    quit()
def info():
    list1 = Listbox()
    list1.insert(1, "Name: apple")
    list1.insert(2, "Producer: iPhone")
    list1.insert(3, "language: Swift")
    list1.insert(4, "OS: iOS")
    list1.grid()
def aboutus():
    list2 = Label(text="About us")
    list2.grid()
    list3 = Label(text="Steve Jobs owner of iPhone")
    list3.grid()

```

Q Write a program to demonstrate GUI component

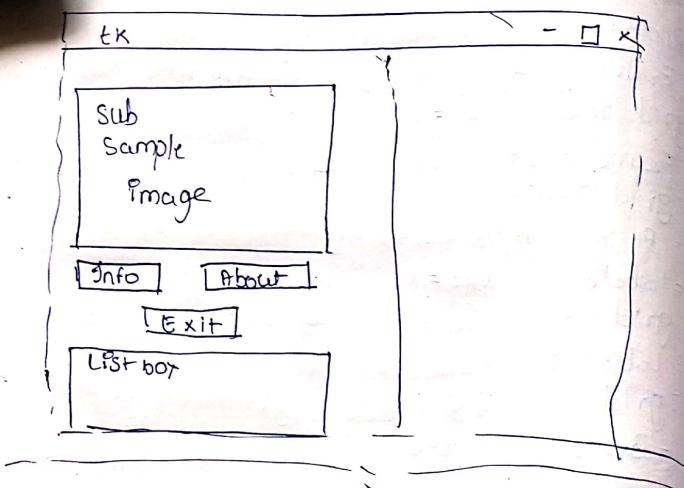
- 1] Import the relevant method from tkinter library
- 2] Create a parent window obj and use the config method along with bg color & attribute specified
- 3] define a function finish with message box widget which will display a message i.e warning message & subsequently terminate program
- 4] define a function info use a list widget along with object of the same. Use the listbox widget along with insert method & insert the same & finally use grid method with ipadx attribute
- 5] Define a function about us with value widget & text attribute. Subsequently use grid method ()

- 6 Use photo img widget with file & filename with gif attribute
- 7 Create a frame object along with frame along with parent object & height & width specified
- 8 Simplay Create another frame obj along with parent window & height & width specified
- 9 Create another obj & use subSample & use label widget with frame object & subsequently use grid()
- 10 & use main loop method

PIFH

```

p1 = PhotoImage(file="download.gif")
f1 = Frame(root, height=35, width=5)
f1.grid(row=0, col=0)
f2 = Frame(root, height=25, width=50)
f2.grid(row=1, col=1)
p2 = f1.SubSample(5, 4)
l1 = Label(f1, image=p2, relief=FLAT)
l1.grid(row=0, col=0, padx=20, pady=15)
l2 = Label(f2, image=p1, relief=SUNKEN)
l2.grid(padx=25, pady=10)
b1 = Button(f1, text="file", relief=SUNKEN,
            command=info)
b1.grid(row=1, col=0)
b2 = Button(f1, text="About us", relief=SUNKEN,
            command>Aboutus)
b3 = Button(f1, text="EXIT", relief=Raised, command=
            root.quit)
b2.grid(row=1, col=1)
    
```



```
From Tkinter import *
root = Tk()
s1 = Spinbox(root, from_=0, to=10)
s1.pack(anchor=S)
l = Label(s1, text="From 1 to 10")
l.pack()
root.mainloop()
```



write a program for spinbox

- > Step 1: Create an object from Tk method & subsequently create an object from Spinbox
- > Make the obj so created onto parent window & triggered corresponding event
- use pack method to provide a direction using anchor method
- use mainloop method

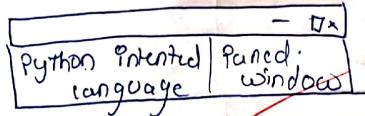
\* Paned windows

1. Create an obj from paned window  
\$ use pack method with attribute fill & expand
2. Create an obj from label method  
\$ but it info's paned window with  
text attribute \$ use add method to  
embed new obj
3. Similarly create a second paned window  
obj & add it to label / &  
paned with orientation specified
4. Now Create another label obj &  
place it onto 2 paned window
- Now use main loop method to  
trigger event

```

from tkinter import *
root = Tk()
p = PanedWindow(bg='red')
p.pack(fill=Both, expand=1)
l1 = Label(p, text="Python intended language", bg="yellow")
p.add(l1)
p1 = PanedWindow(p, orient=vertical, bg='blue')
p.add(p1)
l2 = Label(p1, text="Paned window", bg='gray')
p1.add(l2)
root.mainloop()

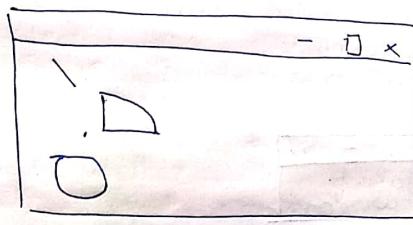
```



```

from Tkinter import *
root = Tk()
c1 = canvas(root, height=400, width=400,
            bg="blue")
oval = c1.create_oval(20, 140, 150, 250, fill="white")
line = c1.create_line(20, 30, 40, 60, fill="green")
arc = c1.create_arc(20, 30, 150, 60, fill="red")
root.mainloop()

```



program to demonstrate canvas

use Tkinter library to import relevant method

Create an obj from canvas obj & use attributes height, width, bg color & parent window obj

use method to create oval, line, arc along with canvas object so create own coordinate value &

now use mainloop method to trigger event

Jan 17/12

### Practical No. 6

Topic : database connectivity

→ Import DBm library & use open method for creating database by specifying name of data along with corresponding flag

- The flag attribute assume 3 string value c/n/w . Now use obj so created for assunning the given website & Corresponding regular name for website

- Check weather the given url address with regular name of page is not equal to display not found & Particular url found else display not found

1 Now import corresponding library which os & sqlite3

Now Create connection object by using Sqlite3 library & connect method for creating new database

```
import dbm
db=dbm.open("database",flag="c")
if db["database"]=="2"
    if db["www.g.com"]==database:
        print("not found")
else:
    print("Found")
```

```
Found
>>> import os,sqlite3
>>> connection=sqlite3.connect("Employee.db")
>>> cursor1=connection.cursor()
>>> cursor.execute("create table Employee
                    Name varchar(20) Primary Key,
                    Emp_id smallint(6),
                    DOJ date,
                    Salary decimal(5,2))")
>>> cursor.execute("insert into employee values('Aditya',100,12-11-10,
                    10000),
                    ('Raj',13,12-12-20,15000),
                    ('Dhaval',201,12-10-20,16000)")
>>> cursor.connection.commit()
>>> cursor1.execute("select * from employee")
>>> cursor1.fetchall()
```

now cursor obj using cursor method from connection obj created in earlier step

now use execute method for create table column name & respective data type

now use cursor obj use insert statement for entering corresponding different field of corresponding data type

use commit method to complete transaction by using connection obj

use execute statement along with cursor obj for accessing value of database by use select with from clause

finally use fetch all method for displaying value of table by cursor obj

use execute method & drop table syntax for terminating database

# Python Project

## \* Program

```
from tkinter import *
root = Tk()
root.title("Python Project")
P = PhotoImage(file = "Taj-hotel-mumbai-colabq-962x392.gif")
label = Label(image = P)
label.image = P # keep a reference
label.grid(row=1, column=0)
l = Label(root, text = "Welcome to taj restaurant",
          fg = "red", font = 20).grid(row = 0, column = 0)
ll = Label(root, text = "Enter your name", font = 10)
ll.grid(row = 2, column = 0)
e = Entry(root, font = 10).grid(row = 0, column = 10)
l2 = Label(root, text = "How many people you are",
           font = 10).grid(row = 3, column = 0)
s = Spinbox(root, from_ = 1, to = 10).grid(row = 3,
                                             column = 1)
r = Radiobutton(root, text = "Air Condition room",
                value = 0).grid(row = 4, column = 0)
r1 = Radiobutton(root, text = "Non Air Condition", value = 1)
r1.grid(row = 4, column = 3)
```

```

def sub():
    top = Tk()
    top.geometry("400x400")
    def s():
        t = Tk()
        P5 = Label(t, text="your order has been recorded in it will place after 5min")
        b = Button(t, text="exit", command=t.destroy)
        b.grid(row=0, column=0)
        l3 = Label(t, text="what do you like to have", font=10).grid(row=1, column=0)
        l5 = Label(t, text="item", font=10).grid(row=4, column=0)
        l1a = Label(t, text="South Indian dishes", font=10)
        l1a.grid(row=0, column=1)
        c1 = Checkbutton(t, text="Idli", font=10)
        c1.grid(row=3, column=1)
        c2 = Checkbutton(t, text="dosa", font=10)
        c2.grid(row=2, column=1)
        c3 = Checkbutton(t, text="masala dosa", font=10)
        c3.grid(row=4, column=1)
        c4 = Checkbutton(t, text="Ovala dosa", font=10)
        c4.grid(row=5, column=1)
        c5 = Checkbutton(t, text="Chinese dosa", font=10)
        c5.grid(row=6, column=1)
    s()
    top.mainloop()

```

l6 = Label (top, text = "Price", font = 10).

l6.grid (row = 1, column = 10)

l8 = Label (top, text = '50', font = 10).grid (row = 3, column = 10)

l9 = Label (top, text = '100', font = 10).grid (row = 4, column = 10)

l10 = Label (top, text = "150", font = 10).grid (row = 5, column = 10)

l11 = Label (top, text = "200", font = 10).grid (row = 6, column = 10)

l12 = Label (top, text = '180', font = 10).grid (row = 7, column = 10)

l7 = Label (top, text = "Quantity", font = 10).grid (row = 1, column = 10)

s1 = Spinbox (top, from = 0, to = 3).grid (row = 3, column = 10)

s2 = Spinbox (top, from = 0, to = 3).grid (row = 4, column = 10)

s3 = Spinbox (top, from = 0, to = 3).grid (row = 5, column = 10)

s4 = Spinbox (top, from = 0, to = 3).grid (row = 6, column = 10)

s5 = Spinbox (top, from = 0, to = 3).grid (row = 7, column = 10)

a = Label (top, text = "Our Speciality", font = 10)

a.grid (row = 8, column = 10)

c5 = Checkbutton (top, text = "Pav bhaji", font = 10)

c5.grid (row = 9, column = 10)

c6 = Checkbutton (top, text = "bhel puri", font = 10)

c6.grid (row = 10, column = 5)

c7 = Checkbutton (top, text = "Paratha", font = 10)

c7.grid (row = 11, column = 5)

c8 = Checkbutton (top, text = "Pulav", font = 10)

c8.grid (row = 12, column = 5)

lalb = Label (top, text = "150", font = 10).grid (row = 9, column = 10)

lc = Label (top, text = "180", font = 10).grid (row = 10, column = 10)

ld = Label (top, text = "200", font = 10).grid (row = 11, column = 10)

lc = Label (top, text = "200", font = 10).grid (row = 12, column = 10)

- \* S6 = Spinbox (top, from=0, to=3).grid(row=9, column=10)
- \* S7 = Spinbox (top, from=0, to=3).grid(row=10, column=10)
- \* S8 = Spinbox (top, from=0, to=3).grid(row=11, column=10)
- \* S9 = Spinbox (top, from=0, to=3).grid(row=12, column=10)
  
- b = Button (top, text="Submit" command=s)
  - b.grid(row=13, column=10)
  - top.mainloop()
  
- b = Button (root, text="Submit", command=sub)
  - ~~b.grid(row=5, column=1)~~

not