Write the program for the following.

1

(A) Create a program that asks the user to enter their name and their age. Print out a message addressed to them that tells them the year that they will turn 100 years old.

```
name = input("What is your name: ")

age = int(input("How old are you: "))

year = str((2017 - age)+100)

print(name + " will be 100 years old in the year " + year)
```

OUTPUT

```
What is your name: Dhiraj kumar sinha

How old are you: 18

Dhiraj kumar sinha will be 100 years old in the year 2099
```

(B) Enter the number from the user and depending on whether the number is even or odd, print out an appropriate message to the user.

```
num = int(input("Enter a number: "))

mod = num % 2

if mod > 0:

   print("You picked an odd number.")

else:

   print("You picked an even number.")
```

OUTPUT

```
Enter a number: 12

You picked an even number.
```

(C) Write a program to generate the Fibonacci series.

```
nterms = int(input("How many terms? "))

# first two terms

n1 = 0

n2 = 1
```

```
count = 2
# check if the number of terms is valid
if nterms <= 0:
   print("Please enter a positive integer")
elif nterms == 1:
   print("Fibonacci sequence upto",nterms,":")
   print(n1)
else:
   print("Fibonacci sequence upto",nterms,":")
   print(n1,",",n2,end=', ')
   while count < nterms:
      nth = n1 + n2
      print(nth,end=' , ')
      # update values
      n1 = n2
      n2 = nth
      count += 1
```

OUTPUT

```
How many terms? 5
Fibonacci sequence upto 5 :
0 , 1, 1 , 2 , 3 ,
```

(d) Write a function that reverses the user defined value.

```
def rev():
    num = int(input("Enter a number: "))
    org=num
    rev=0
    while num!=0:
       rem=num%10
       num=num//10
       rev=rev*10+rem
```

```
    print("Reverse of number ",org,"is = ",rev)
```

OUTPUT

```
>>> rev()
Enter a number: 123

Reverse of number  123 is =  321
```

(E) Write a function to check the input value is Armstrong and also write the function for Palindrome.

```
print('!* To Find Palindrome and/or Armstrong Number')

def Palindrome_Number():

   n = int(input('Enter a Number to check for palindromee'))

   m=n

   a = 0

   while(m!=0):

      a = m % 10 + a * 10

      m = m // 10

   if( n == a):

      print(n,'is a palindrome number')

   else:

      print(n,'is not a palindrome number')

def Armstrong_Number():

    number = int(input('Enter a Number to check for Armstrong'))

    temp = number

    Sum = 0

     #loop till the quotient is 0

    while(temp != 0):

       rem  = temp % 10 #find reminder

       Sum  = Sum + (rem * rem * rem) #cube reminder and add it to the Sum

       temp = temp // 10 #find quotient, if 0 then loop again

         #if the entered number and the Sum value matches, it is an Armstrong number

    if(number == Sum):
```

```
        print (number,"is Armstrong Number")
    else:
        print (number,"is Not an Armstrong Number")
```

OUTPUT

```
!* To Find Palindrome and/or Armstrong Number
>>> Palindrome_Number()
Enter a Number to check for palindrome:12321
12321 is a palindrome number
>>> Armstrong_Number()
Enter a Number to check for Armstrong:153
153 is Armstrong Number
```

(f) Write a recursive function to print the factorial for a given number.

```
def recur_factorial(n):
  """Function to return the factorial
  of a number using recursion"""
  if n == 1:
     return n
  else:
     return n*recur_factorial(n-1)
#  take input from the user
num = int(input("Enter a number: "))
# check is the number is negative
if num < 0:
  print("Sorry, factorial does not exist for negative numbers")
elif num == 0:
  print("The factorial of 0 is 1")
else:
  print("The factorial of",num,"is",recur_factorial(num))
```

OUTPUT

```
Enter a number: -1
Sorry, factorial does not exist for negative numbers
Enter a number: 0
The factorial of 0 is 1
Enter a number: 5
The factorial of 5 is 120
```

2

(A) Write a function that takes a character (i.e. a string of length 1) and returns True if it is a vowel, False otherwise

```
def is_vowel(char):
    vowels = ('a', 'e', 'i', 'o', 'u')
    if char not in vowels:
        return False
    return True
c= input("Enter a character of length 1 to check whether it is vowel or not:")
if len(c)==1:
    print ("Result is",is_vowel(c))
else:
    print("You have not entered character of length 1")
```

OUTPUT

```
Enter a character of length 1 to check whether it is vowel or not: abcd
You have not entered character of length 1
Enter a character of length 1 to check whether it is vowel or not:e
Result is True
```

(B) Define a function that computes the length of a given list or string.

```
def cal_length(str):
    length= len(str)
    print(" length of the string \"",str,"\" is = ", length)
c= input("Enter the string   ")
```

```
cal_length(c)
```

OUTPUT

```
Enter the string   dhiraj
 length of the string " dhiraj " is =  6
```

(C) Define a procedure histogram() that takes a list of integers and prints a histogram to the screen. For example, histogram([4, 9, 7]) should print the following:

****

****

****

```
def histogram(inputList):
    for i in range(len(inputList)):
      print (inputList[i]*'*')
List = [4,9,7]
histogram(List)
```

OUTPUT

```
****
*********
*******
```

3

(A) A pangram is a sentence that contains all the letters of the English alphabet at least once, for example: The quick brown fox jumps over the lazy dog. Your task here is to write a function to check a sentence to see if it is a pangram or not.

```
import re
def isPangram(inputSentence):
   alphabetList = 'abcdefghijklmnopqrstuvwzyx'
   alphabetCount = 0
   if len(inputSentence) < 26:
```

```
        return False
    else:
        inputSentence = re.sub('[^a-zA-Z]','',inputSentence)
        inputSentence = inputSentence.lower()
        for i in range(len(alphabetList)):
            if alphabetList[i] in inputSentence:
                alphabetCount = alphabetCount+1
        if alphabetCount == 26:
            return True
        else:
            return False
print ("Enter a sentence : ")
 inputSentence = input()
if (isPangram(inputSentence)):
    print ("Input Sentence is a Pangram")
else:
    print ("Input Sentence is not a Pangram")
```

OUTPUT

```
Enter a sentence :
The quick brown fox jumps over the lazy dog
Input Sentence is a Pangram
Enter a sentence :
dhiraj kumar sinha
Input Sentence is not a Pangram
```

(B) Take a list, say for example this one:

a = [1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89]

and write a program that prints out all the elements of the list that are less than 5.

```
myList=[1, 1, 12, 3, 5, 8, 13, 21, 3, 55, 89]
```

```
myNewList=[]
#number=int(input("Enter a number"))
for i in myList:
    if(i<5):
        myNewList.append(i)

print("List containing numbers less than 5 is/are :", myNewList)
```

OUTPUT

```
List containing numbers less than 5 is/are : [1, 1, 3, 3]
```

4

(A) Write a program that takes two lists and returns True if they have at least one common member.

```
def common_data(list1, list2):
    result = False
    for x in list1:
        for y in list2:
            if x == y:
                result = True
    return result
print(common_data([1,2,3,4,5], [5,6,7,8,9]))
print(common_data([1,2,3,4,5], [6,7,8,9]))

print(common_data([1,2,10,4,5], [6,7,8,9]))
```

OUTPUT

```
True
False
False
```

(B) Write a Python program to print a specified list after removing the 0th, 2nd, 4th and 5th elements.

```
li = [12,24,35,70,88,120,155]



li = [x for (i,x) in enumerate(li) if i not in
        (0,2,4,5)]
print (li)
```

OUTPUT

```
[24, 70, 155]
```

(C) Write a Python program to clone or copy a list.

```
original_list = [10, 22, 44, 23, 4]
new_list = list(original_list)
print(original_list)
print(new_list)
```

OUTPUT

```
[10, 22, 44, 23, 4]
[10, 22, 44, 23, 4]
```

5

(A) Write a Python script to sort (ascending and descending) a dictionary by value.

```
import operator
d = {1: 2, 3: 4, 4: 3, 2: 1, 0: 0}
print('Original dictionary : ',d)
sorted_d = sorted(d.items(), key=operator.itemgetter(1))
print('Dictionary in ascending order by value : ',sorted_d)
sorted_d = sorted(d.items(), key=operator.itemgetter(1),reverse=True)
print('Dictionary in descending order by value : ',sorted_d)
```

OUTPUT

```
Original dictionary :  {1: 2, 3: 4, 4: 3, 2: 1, 0: 0}
```

Dictionary in ascending order by value : [(0, 0), (2, 1), (1, 2), (4, 3), (3, 4)]

Dictionary in descending order by value : [(3, 4), (4, 3), (1, 2), (2, 1), (0, 0)]

(B) Write a Python script to concatenate following dictionaries to create a new one. Sample Dictionary :

```
dic1={1:10, 2:20}
dic2={3:30, 4:40}
dic3={5:50,6:60}
```

Expected Result : {1: 10, 2: 20, 3: 30, 4: 40, 5: 50, 6: 60}

```
dic1={1:10, 2:20}
dic2={3:30, 4:40}
dic3={5:50,6:60}
dic4 = {}
for d in (dic1, dic2, dic3):
   dic4.update(d)
print(dic4)
```

OUTPUT

```
{1: 10, 2: 20, 3: 30, 4: 40, 5: 50, 6: 60}
```

(C) Write a Python program to sum all the items in a dictionary.

```
dict1={'data1':10,'data2':54,'data3':24}
print('dictionary of addition is',sum(dict1.values()))
```

OUTPUT

```
dictionary of addition is 88
```

6

(A) Write a Python program to read an entire text file.

```
f=open('test.txt')
text=f.read()
print(text)
```

test.txt

HELLO WORLD

OUTPUT

HELLO WORLD

(B) Write a Python program to append text to a file and display the text.

```
file=open('text2.txt','a')
file.write('Hello World\n')
file.write('This is our new text file\n')
file.write('and this is another line.\n')
file.write('Why? Because we can.\n')
file.close()
f2=open('text2.txt')
print(f2.read())
```

text2.txt

this is text2 file.

OUTPUT

this is text2 file.

Hello World

This is our new text file

and this is another line.

Why? Because we can.

(c) Write a Python program to read last n lines of a file.

```
import sys
import os
def file_read_from_tail(fname,lines):
    bufsize = 8192
    fsize = os.stat(fname).st_size
    iter = 0
    with open(fname) as f:
        if bufsize > fsize:
```

```
        bufsize = fsize-1
        data = []
        while True:
            iter +=1
            f.seek(fsize-bufsize*iter)
            data.extend(f.readlines())
            if len(data) >= lines or f.tell() == 0:
                print(''.join(data[-lines:]))
                break


file_read_from_tail('text3.txt',2)
```

test3.txt

this is text2 file.

Hello World

This is our new text file

and this is another line.

Why? Because we can.

OUTPUT

and this is another line.

Why? Because we can.

7.

(A) Design a class that store the information of student and display the same.

```
class student:
    'common base class for all employee'
    studentcount=0


    def __init__(self,name,age):
        self.name=name
        self.age=age
```

```
        student.studentcount+=1


    def displaycount(self):
        print('total student',student.studentcount)


    def displaystudent(self):
        print('name:',self.name,'\n','Age:',self.age)


stud1=student('Dhiraj kumar',19)
stud2=student('Sinha',19)
stud1.displaystudent()
stud2.displaystudent()
print('total student are',student.studentcount)
```

OUTPUT

```
name: Dhiraj kumar
 Age: 19
name: Sinha
 Age: 19
total student are 2
```

(B) Implement the concept of inheritance using python.

```
class Polygon:
  def __init__(self, no_of_sides):
    self.n = no_of_sides
    self.sides = [0 for i in range(no_of_sides)]


  def inputSides(self):
    self.sides = [float(input("Enter side "+str(i+1)+" : ")) for i in range(self.n)]


  def dispSides(self):
```

```
    for i in range(self.n):
        print("Side",i+1,"is",self.sides[i])


class Triangle(Polygon):
  def __init__(self):
    Polygon.__init__(self,3)


  def findArea(self):
    a, b, c = self.sides
    # calculate the semi-perimeter
    s = (a + b + c) / 2
    area = (s*(s-a)*(s-b)*(s-c)) ** 0.5
    print('The area of the triangle is %0.2f',area)
```

OUTPUT

```
>>> t = Triangle()
>>> t.inputSides()
Enter side 1 : 3
Enter side 2 : 4
Enter side 3 : 5
>>> t.dispSides()
Side 1 is 3.0
Side 2 is 4.0
Side 3 is 5.0
>>> t.findArea()
The area of the triangle is %0.2f 6.0
```

(C) Create a class called Numbers, which has a single class attribute called MULTIPLIER, and a constructor which takes the parameters x and y (these should all be numbers).

i. **Write a method called add which returns the sum of the attributes x and y.**

ii. Write a class method called multiply, which takes a single number parameter a and returns the product of a and MULTIPLIER.

iii. Write a static method called subtract, which takes two number parameters, b and c, and returns b - c.

iv. Write a method called value which returns a tuple containing the values of x and y. Make this method into a property, and write a setter and a deleter for manipulating the values of x and y.

```
class numbers:
    Multiplier=3.5

    def __init__(self,x,y):
        self.x=x
        self.y=y

    def add(self):
        return self.x+self.y

    @classmethod
    def multiply(cls,a):
        return cls.MULTIPLIER*a

    @staticmethod
    def subtract(b,c):
        return b-c

    @property
    def value(self):
        return(self.x,self.y)

    @value.setter
    def value(self,xy_tuple):
        self.x,self.y=xy_tuple

    @value.deleter
    def value(self):
        del self.x
        del self.y
```

OUTPUT

```
>>>
```

8.

(A) Open a new file in IDLE ("New Window" in the "File" menu) and save it as geometry.py in the directory where you keep the files you create for this course. Then copy the functions you wrote for calculating volumes and areas in the "Control Flow and Functions" exercise into this file and save it.

Now open a new file and save it in the same directory. You should now be able to import your own module like this:

import geometry
Try and add print dir(geometry) to the file and run it.

Now write a function pointyShapeVolume(x, y, squareBase) that calculates the volume of a square pyramid if squareBase is True and of a right circular cone if squareBase is False. x is the length of an edge on a square if squareBase is True and the radius of a circle when squareBase is False. y is the height of the object. First use squareBase to distinguish the cases. Use the circleArea and squareArea from the geometry module to calculate the base areas.

```
import geometry
def pointyShapevolume(x,h,square):
    if square:
        base=geometry.square(x)
    else:
        base=geometry.circlearea(x)
    return h*base/3.0
print(pointyShapevolume(4,2.6,True))
print(pointyShapevolume(4,2.6, False))
```

Geometry.py

```
def square(x):
    return ((x**2)*3.14)
def circlearea(base):
    return (base*base)
```

OUTPUT

```
43.54133333333334
13.866666666666667
```

(B) Write a program to implement exception handling.

```
import sys

randomList = ['a', 0, 2]

for entry in randomList:
  try:
    print("The entry is", entry)
    r = 1/int(entry)
    break
  except:
    print("Oops!",sys.exc_info()[0],"occured.")
    print("Next entry.")
    print()
print("The reciprocal of",entry,"is",r)
```

OUTPUT

```
The entry is a
```

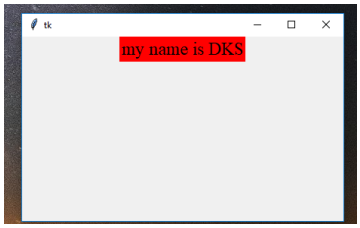
```
Oops! <class 'ValueError'> occured.
Next entry.

The entry is 0
Oops! <class 'ZeroDivisionError'> occured.
Next entry.

The entry is 2
The reciprocal of 2 is 0.5
```

9.

(A) Try to configure the widget with various options like: bg=”red”, family=”times”, size=18

```
from tkinter import *
root=Tk()
w=Label(root,text='my name is DKS',bg='red',font='Times 18')
w.pack()
root.mainloop()
```

OUTPUT

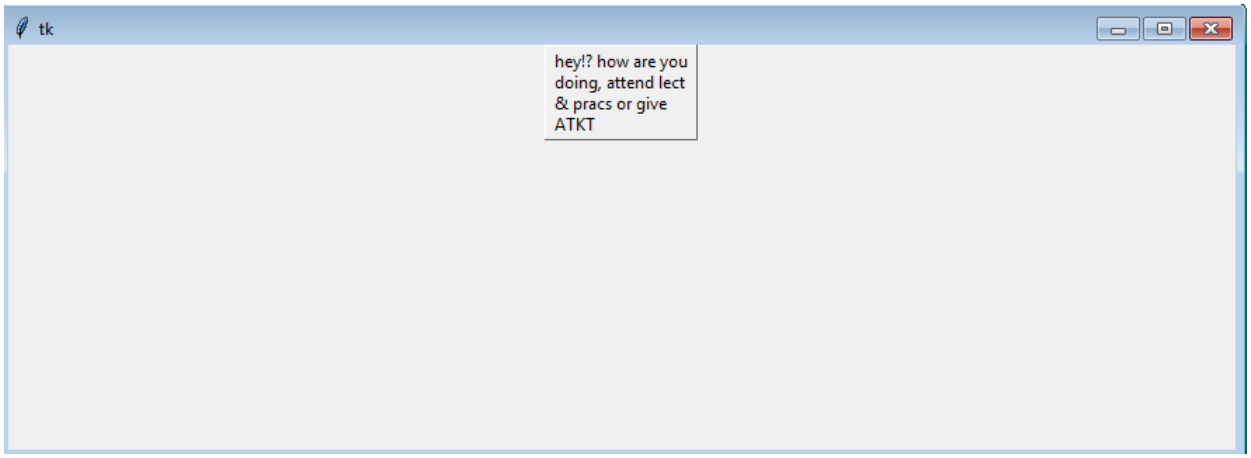


(B) Try to change the widget type and configuration options to experiment with other widget types like Message, Button, Entry, Checkbutton, Radiobutton, Scale etc.

#Meassage.py

```
import tkinter
from tkinter import *
root=Tk()
var=StringVar()
label=Message(root,textvariable=var,relief=RAISED)
var.set("hey!? how are you doing, attend lect&pracs or give ATKT")
label.pack()
root.mainloop()
```

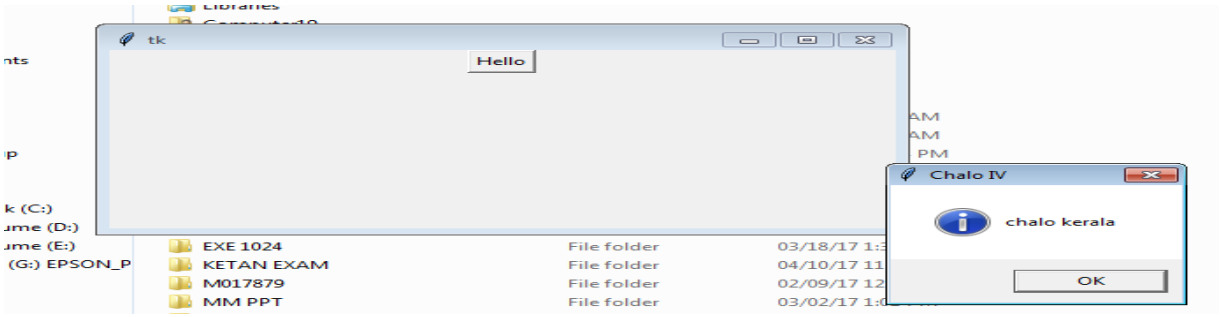
OUTPUT



#Button.py

```
import tkinter
from tkinter import *
top=tkinter.Tk()
def helloCallBack():
    tkinter.messagebox.showinfo("Chalo IV","chalo kerala")
B=tkinter.Button(top,text="Hello",command=helloCallBack)
B.pack()
top.mainloop()
```

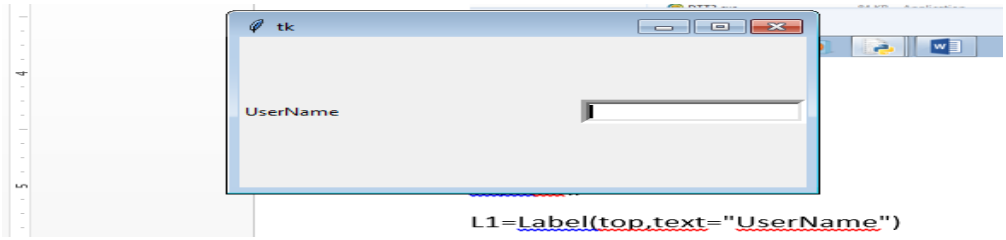
OUTPUT



#Entry.py

```
import tkinter
from tkinter import *
top=Tk()
L1=Label(top,text="UserName")
L1.pack(side=LEFT)
E1=Entry(top,bd=5)
E1.pack(side=RIGHT)
top.mainloop()
```

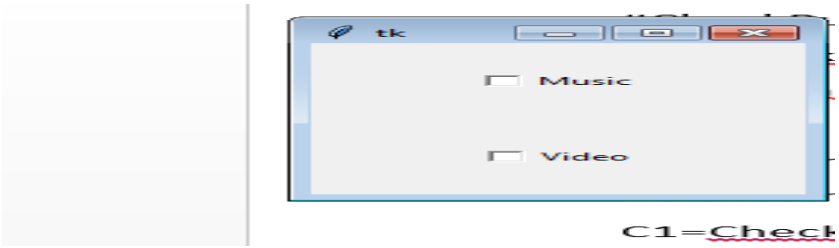
OUTPUT



#CheckButton.py

```
import tkinter
from tkinter import *
top=Tk()
CheckVar1=IntVar()
CheckVar2=IntVar()
C1=Checkbutton(top,text="Music",variable=CheckVar1,
onvalue=1,offvalue=0,height=5,width=20)
C2=Checkbutton(top,text="Video",variable=CheckVar2,
onvalue=1,offvalue=0,height=5,width=20)
C1.pack()
C2.pack()
top.mainloop()
```

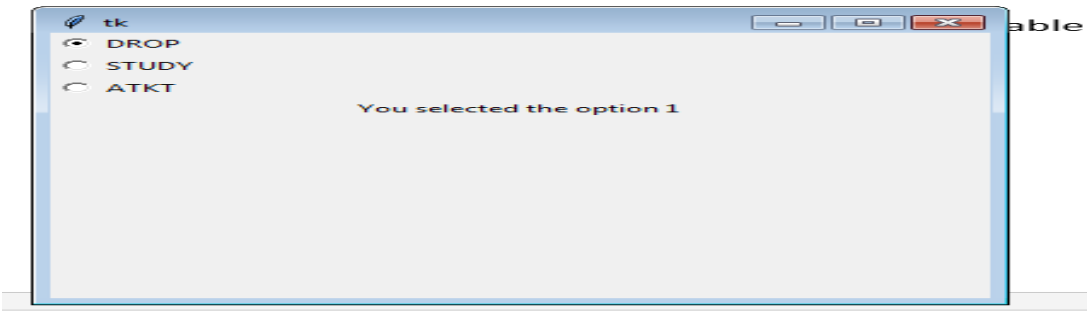
OUTPUT



#RadioButton.py

```
import tkinter
from tkinter import *
def sel():
    selection="You selected the option " + str(var.get())
    Label.config(text=selection)
root=Tk()
var=IntVar()
R1=Radiobutton(root,text="DROP",variable=var,value=1,command=sel)
R1.pack(anchor=W)
R2=Radiobutton(root,text="STUDY",variable=var,value=2,command=sel)
R2.pack(anchor=W)
R3=Radiobutton(root,text="ATKT",variable=var,value=3,command=sel)
R3.pack(anchor=W)
label=Label(root)
label.pack()
root.mainloop()
```

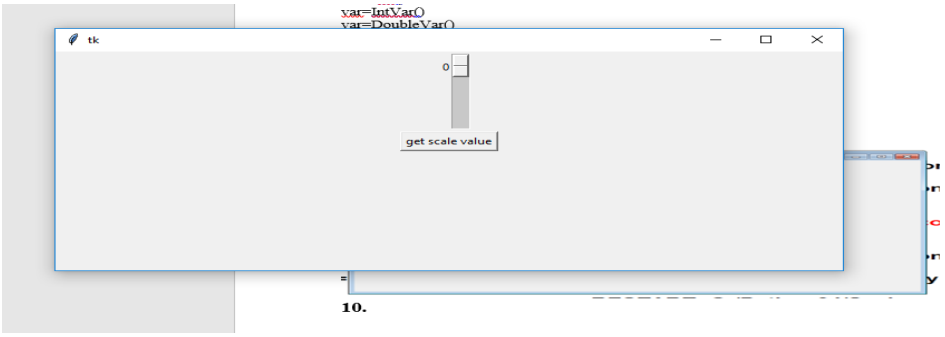
OUTPUT



#Scale.py

```
import tkinter
from tkinter import *
def sel():
    selection="Value =" + str(var.get())
    label.config(text=selection)
root=Tk()
var=IntVar()
var=DoubleVar()
scale=Scale(root,variable=var)
scale.pack(anchor=CENTER)
button=Button(root,text="get scale value",command=sel)
button.pack(anchor=CENTER)
label=Label(root)
label.pack()
root.mainloop()
```

OUTPUT



10.

(A) Design a simple database application that stores the records and retrieve the same.

OUTPUT

(B) Design a database application to search the specified record from the database.

OUTPUT

(C) Design a database application to that allows the user to add, delete and modify the records.

OUTPUT