```
from tkinter import *
from PIL import Image,ImageTk       #external library to attach image pip install pillow
#fields is to take the input from the user
fields = ('Basic Pay','Provident fund',"special pay",'stagnation pay','graduation pay','number of years of work','age at next birthday','commutation percentage')

#entries is a dictonary which is used to store the inputs given by the user

def pension(entries):          #pension function is used to calculate pension
    x_years=33-float(entries['number of years of work'].get())      #x_years is used to store how many years less than maximum years.
    if float(entries['number of years of work'].get())<10:
        #entries[].get() will take the input given by the on the gui window
        pension=0                                       #store the pension calculated from the given inputs
        print('pension:',int(pension))
    elif float(entries['number of years of work'].get())>=33:
        pension=(float(entries['Basic Pay'].get())+float(entries['Provident fund'].get())+float(entries['special pay'].get())+float(entries['stagnation pay'].get())+float(entries['graduation pay'].get()))/2
        print('pension:',int(pension))
    else:
        pension=(float(entries['Basic Pay'].get())+float(entries['Provident fund'].get())+float(entries['special pay'].get())+float(entries['stagnation pay'].get())+float(entries['graduation pay'].get()))/2
        pension=pension-(x_years*3.03035)
        print('pension:',int(pension))          #explict convertion of pension
    label001=Label(root,text="pension is:"+str(pension),font=(25))      #label001 is to print the  output on the gu window
    label001.place(relx=0.0,rely=1.0,anchor='sw')
def commutation(entries):          #commutation function is used to calculate commuation
    x_age=float(entries['age at next birthday'].get())-61                #x_age is used to store the age of the user
    x_years=33-float(entries['number of years of work'].get())             #x_years is used to store the numbe of years the user worked
    if float(entries['number of years of work'].get())<10:
        pension=0
    elif float(entries['number of years of work'].get())>=33:
        pension=(float(entries['Basic Pay'].get())+float(entries['Provident fund'].get())+float(entries['special pay'].get())+float(entries['stagnation pay'].get())+float(entries['graduation pay'].get()))/2
    else:
        pension=(float(entries['Basic Pay'].get())+float(entries['Provident fund'].get())+float(entries['special pay'].get())+float(entries['stagnation pay'].get())+float(entries['graduation pay'].get()))/2
        pension=pension-(x_years*3.03035)
    commuatation_percentage=entries['commutation percentage'].get()     #store the commuation percentage entered by the user in the gui window
    commuatation_percentage=float(commuatation_percentage)
    if commuatation_percentage>=33.33:
        commutation_amount= float(pension*33.33)/100
    else:
        commutation_amount= float(pension*commuatation_percentage)/100                          #commutation_amount is used to store the amount to deducted from the pension of the user
    commutation_factor=9.81                                     #commutation_factor store the maximum value of factor
    final_commutation_factor=commutation_factor-(x_age*0.3297)             #final_commutation_factor is the commutation_factor based on the age of the user
```

```
    commutation_value =commutation_amount*final_commutation_factor*12       #commutation_value st
ores the amount to be paid to user
    commutation_amount=int(commutation_amount)
    commutation_value=int(commutation_value)
    reduced_pension=pension-
commutation_amount                          #reduced_pension is the amount paid to the user after commuatat
ion
    reduced_pension=int(reduced_pension)
    print(commutation_value,'\n',commutation_amount,'\n',final_commutation_factor,'\n',reduced_pension)
    label002=Label(root,text="pension:"+str(pension)+'\n'+"commutation_amount:"+str(commutation_am
ount)+'\ncommutation_factor:'+str(final_commutation_factor)+'\ncommutation_value:'+str(commutation_
value)+'\nreduced_pension:'+str(reduced_pension)+'\n',font=(25))
    label002.place(relx=0.0,rely=1.0,anchor='sw')                       #label002 is to print the commuatation d
etails on the gui window
def pensionform(root, fields):              #pensionform is function to design the input taking things
  entries = {}
  for field in fields:
    row = Frame(root)                    #row is to initilise the frame function from the tkinter library
    lab = Label(row, width=22, text=field+": ", anchor='w',font=("arial",12))            #lab is a label to take
 input from the user
    ent = Entry(row,font=(8))                              #ent is entry to fill the input values by user
    ent.insert(0,"0")
    row.pack(fill = X, padx = 5 , pady = 5)
    lab.pack(side = LEFT,pady=5,padx=5)
    ent.pack(side = RIGHT, expand = YES, fill = X)
    entries[field] = ent
  return entries
if __name__ == '__main__':
  root = Tk()              #root is to initilise the tkinter library
  root.title('Calculator')
  root.geometry("700x800+60+600")
  lab=Label(root,text='Calculator for pension and commutaion of bank employees',bg='red',font=('arial',1
6,'bold'))
  lab.pack()
  imge=Image.open("calci.jpg")                #imge and photo are used to open adn display the photo on gui
window
  imge=imge.resize((200,200),Image.ANTIALIAS)
  photo=ImageTk.PhotoImage(imge)
  label007=Label(root,image=photo)            #label007 is to attach the photo to th gui window
  label007.pack(side=TOP)
  ents = pensionform(root, fields)
  root.bind('<Return>', (lambda event, e = ents: fetch(e)))
  b1 = Button(root, text = 'Quit',fg="white",bg="red",font=(10),command=root.quit)
  b1.pack(side = RIGHT, padx = 50, pady = 40)
  b2 = Button(root, text='commutation',fg="blue",bg="yellow",font=(10),command=(lambda e=ents:com
mutation(e)))
  b2.pack(side = RIGHT, padx = 50, pady = 40)
  b3 = Button(root, text = 'pension',fg="white",bg="green",font=(10),command=(lambda e=ents:pension(
e)))
  b3.pack(side = RIGHT, padx = 50, pady = 40)
  root.mainloop()
```