```
#from this import s
from tkinter import *
import pygame
pygame.mixer.init()
pygame.mixer.music.load("C:\\Users\\srija\\Downloads\\for-future-bass-
159125.mp3")
pygame.mixer.music.play()
window = Tk()
window.title('SCIENTIFIC CALCULATOR')
window.geometry('900x500+400+150')
bg = PhotoImage(
file="C:\\Users\\srija\\OneDrive\\Pictures\\welcome1.png" )
my_label = Label(window , image = bg)
my_label.place(x = 0 ,y = 0 ,relwidth = 1 ,relheight = 1)


def second_window():
    window.withdraw()
    import math
    import tkinter.messagebox
    import pygame

    pygame.mixer.init()
    pygame.init()
    sound = pygame.mixer.Sound("C:\\Users\\srija\\Downloads\\interface-
124464.mp3")

    root = Tk()
    root.title("Scientific Calculator")
    root.configure(background='red')
    root.resizable(width=False, height=False)
    root.geometry("480x568+450+90")

    calc = Frame(root)
    calc.grid()


    class Calc:
        def __init__(self):
            self.total = 0
            self.current = ''
            self.input_value = True
            self.check_sum = False
            self.op = ''
            self.result = False

        def numberEnter(self, num):
            self.result = False
            firstnum = txtDisplay.get()
            secondnum = str(num)
            if self.input_value:
                self.current = secondnum
                self.input_value = False
            else:
```

```
            if secondnum == '.':
                if secondnum in firstnum:
                    return
            self.current = firstnum + secondnum
        self.display(self.current)
        sound.play()

    def sum_of_total(self):
        self.result = True
        self.current = float(self.current)
        if self.check_sum:
            self.valid_function()
        else:
            self.total = float(txtDisplay.get())

    def display(self, value):
        txtDisplay.delete(0, END)
        txtDisplay.insert(0, value)

    def valid_function(self):
        if self.op == "add":
            self.total += self.current
        if self.op == "sub":
            self.total -= self.current
        if self.op == "multi":
            self.total *= self.current
        if self.op == "divide":
            self.total /= self.current
        if self.op == "mod":
            self.total %= self.current
        self.input_value = True
        self.check_sum = False
        self.display(self.total)
        sound.play()

    def operation(self, op):
        self.current = float(self.current)
        if self.check_sum:
            self.valid_function()
        elif not self.result:
            self.total = self.current
            self.input_value = True
        self.check_sum = True
        self.op = op
        self.result = False
        sound.play()

    def Clear_Entry(self):
        self.result = False
        self.current = "0"
        self.display(0)
        self.input_value = True
        sound.play()
```

```
        def All_Clear_Entry(self):
            self.Clear_Entry()
            self.total = 0
            sound.play()

        def pi(self):
            self.result = False
            self.current = math.pi
            self.display(self.current)
            sound.play()

        def tau(self):
            self.result = False
            self.current = math.tau
            self.display(self.current)
            sound.play()

        def e(self):
            self.result = False
            self.current = math.e
            self.display(self.current)
            sound.play()

        def backspace(self):
            self.result = False
            self.current = txtDisplay.get()
            self.current = self.current[:-1]  # Remove the last character
            self.display(self.current)
            sound.play()

        def squared(self):
            self.result = False
            self.current = math.sqrt(float(txtDisplay.get()))
            self.display(self.current)
            sound.play()

        def cos(self):
            self.result = False
            self.current =
math.cos(math.radians(float(txtDisplay.get())))
            self.display(self.current)
            sound.play()

        def cosh(self):
            self.result = False
            self.current =
math.cosh(math.radians(float(txtDisplay.get())))
            self.display(self.current)
            sound.play()

        def tan(self):
            self.result = False
            self.current =
math.tan(math.radians(float(txtDisplay.get())))
```

```
            self.display(self.current)
            sound.play()

        def tanh(self):
            self.result = False
            self.current =
math.tanh(math.radians(float(txtDisplay.get())))
            self.display(self.current)
            sound.play()

        def sin(self):
            self.result = False
            self.current =
math.sin(math.radians(float(txtDisplay.get())))
            self.display(self.current)
            sound.play()

        def sinh(self):
            self.result = False
            self.current =
math.sinh(math.radians(float(txtDisplay.get())))
            self.display(self.current)
            sound.play()

        def log(self):
            self.result = False
            self.current = math.log(float(txtDisplay.get()))
            self.display(self.current)
            sound.play()

        def exp(self):
            self.result = False
            self.current = math.exp(float(txtDisplay.get()))
            self.display(self.current)
            sound.play()

        def acosh(self):
            self.result = False
            self.current = math.acosh(float(txtDisplay.get()))
            self.display(self.current)
            sound.play()

        def asinh(self):
            self.result = False
            self.current = math.asinh(float(txtDisplay.get()))
            self.display(self.current)
            sound.play()


        def expm1(self):
            self.result = False
            self.current = math.expm1(float(txtDisplay.get()))
            self.display(self.current)
            sound.play()
```

```
        def lgamma(self):
            self.result = False
            self.current = math.lgamma(float(txtDisplay.get()))
            self.display(self.current)
            sound.play()

        def degrees(self):
            self.result = False
            self.current = math.degrees(float(txtDisplay.get()))
            self.display(self.current)
            sound.play()

        def log2(self):
            self.result = False
            self.current = math.log2(float(txtDisplay.get()))
            self.display(self.current)
            sound.play()

        def log10(self):
            self.result = False
            self.current = math.log10(float(txtDisplay.get()))
            self.display(self.current)
            sound.play()

        def log1p(self):
            self.result = False
            self.current = math.log1p(float(txtDisplay.get()))
            self.display(self.current)
            sound.play()


    added_value = Calc()

    txtDisplay = Entry(calc, font=('Helvetica', 20, 'bold'),
                       bg='black', fg='red',
                       bd=30, width=28, justify=LEFT)
    txtDisplay.grid(row=0, column=0, columnspan=4, pady=1)
    txtDisplay.insert(0, "0")

    numberpad = "789456123"
    i = 0
    btn = []
    for j in range(2,5):
        for k in range(3):
            btn.append(Button(calc, width=6, height=2,
                              bg='black', fg='red',
                              font=('Helvetica', 20, 'bold'),
                              bd=4, text=numberpad[i]))
            btn[i].grid(row=j, column=k, pady=1)
            btn[i]["command"] = lambda x=numberpad[i]:
added_value.numberEnter(x)
            i += 1
```

```
btnClear = Button(calc, text=chr(67), width=6,
              height=2, bg='powder blue',
              font=('Helvetica', 20, 'bold'),
              bd=4, command=added_value.Clear_Entry)
btnClear.grid(row=1, column=0, pady=1)

btnAllClear = Button(calc, text=chr(67) + chr(69),
                 width=6, height=2,
                 bg='powder blue',
                 font=('Helvetica', 20, 'bold'),
                 bd=4,
                 command=added_value.All_Clear_Entry)
btnAllClear.grid(row=1, column=1, pady=1)

btnsq = Button(calc, text="\u221A", width=6, height=2,
           bg='powder blue', font=('Helvetica',
                                   20, 'bold'),
           bd=4, command=added_value.squared)
btnsq.grid(row=1, column=2, pady=1)

btnAdd = Button(calc, text="+", width=6, height=2,
            bg='powder blue',
            font=('Helvetica', 20, 'bold'),
            bd=4, command=lambda: added_value.operation("add"))
btnAdd.grid(row=1, column=3, pady=1)

btnSub = Button(calc, text="-", width=6,
            height=2, bg='powder blue',
            font=('Helvetica', 20, 'bold'),
                bd=4, command=lambda: added_value.operation("sub"))
btnSub.grid(row=2, column=3, pady=1)

btnMul = Button(calc, text="x", width=6,
            height=2, bg='powder blue',
            font=('Helvetica', 20, 'bold'),
            bd=4, command=lambda: added_value.operation("multi"))
btnMul.grid(row=3, column=3, pady=1)

btnDiv = Button(calc, text="/", width=6,
            height=2, bg='powder blue',
            font=('Helvetica', 20, 'bold'),
            bd=4, command=lambda: added_value.operation("divide"))
btnDiv.grid(row=4, column=3, pady=1)

btnZero = Button(calc, text="0", width=6,
             height=2, bg='black', fg='red',
             font=('Helvetica', 20, 'bold'),
             bd=4, command=lambda: added_value.numberEnter(0))
btnZero.grid(row=5, column=0, pady=1)

btnDot = Button(calc, text=".", width=6,
            height=2, bg='powder blue',
            font=('Helvetica', 20, 'bold'),
            bd=4, command=lambda: added_value.numberEnter("."))
```

```
    btnDot.grid(row=5, column=1, pady=1)

    btnPM = Button(calc, text="Bs", width=6,
                height=2, bg='powder blue', font=('Helvetica', 20,
'bold'),
                bd=4, command=added_value.backspace)
    btnPM.grid(row=5, column=2, pady=1)

    btnEquals = Button(calc, text="=", width=6,
                    height=2, bg='powder blue',
                    font=('Helvetica', 20, 'bold'),
                    bd=4, command=added_value.sum_of_total)
    btnEquals.grid(row=5, column=3, pady=1)

    btnPi = Button(calc, text="pi", width=6,
                height=2, bg='light green', fg='black',
                font=('Helvetica', 20, 'bold'),
                bd=4, command=added_value.pi)
    btnPi.grid(row=1, column=4, pady=1)

    btnCos = Button(calc, text="Cos", width=6,
                 height=2, bg='light green', fg='black',
                 font=('Helvetica', 20, 'bold'),
                 bd=4, command=added_value.cos)
    btnCos.grid(row=1, column=5, pady=1)

    btntan = Button(calc, text="tan", width=6,
                 height=2, bg='light green', fg='black',
                 font=('Helvetica', 20, 'bold'),
                 bd=4, command=added_value.tan)
    btntan.grid(row=1, column=6, pady=1)

    btnsin = Button(calc, text="sin", width=6,
                 height=2, bg='light green', fg='black',
                 font=('Helvetica', 20, 'bold'),
                 bd=4, command=added_value.sin)
    btnsin.grid(row=1, column=7, pady=1)

    btn2Pi = Button(calc, text="2pi", width=6,
                 height=2, bg='light green', fg='black',
                 font=('Helvetica', 20, 'bold'),
                 bd=4, command=added_value.tau)
    btn2Pi.grid(row=2, column=4, pady=1)

    btnCosh = Button(calc, text="Cosh", width=6,
                  height=2, bg='light green', fg='black',
                  font=('Helvetica', 20, 'bold'),
                  bd=4, command=added_value.cosh)
    btnCosh.grid(row=2, column=5, pady=1)

    btntanh = Button(calc, text="tanh", width=6,
                  height=2, bg='light green', fg='black',
                  font=('Helvetica', 20, 'bold'),
                  bd=4, command=added_value.tanh)
```

```
btntanh.grid(row=2, column=6, pady=1)

btnsinh = Button(calc, text="sinh", width=6,
             height=2, bg='light green', fg='black',

             font=('Helvetica', 20, 'bold'),
             bd=4, command=added_value.sinh)
btnsinh.grid(row=2, column=7, pady=1)

btnlog = Button(calc, text="log", width=6,
            height=2, bg='light green', fg='black',
            font=('Helvetica', 20, 'bold'),
                bd=4, command=added_value.log)
btnlog.grid(row=3, column=4, pady=1)

btnExp = Button(calc, text="exp", width=6, height=2,
            bg='light green', fg='black',
            font=('Helvetica', 20, 'bold'),
            bd=4, command=added_value.exp)
btnExp.grid(row=3, column=5, pady=1)

btnMod = Button(calc, text="Mod", width=6,
            height=2, bg='light green', fg='black',
            font=('Helvetica', 20, 'bold'),
            bd=4, command=lambda: added_value.operation("mod"))
btnMod.grid(row=3, column=6, pady=1)

btnE = Button(calc, text="e", width=6,
          height=2, bg='light green', fg='black',
          font=('Helvetica', 20, 'bold'),
          bd=4, command=added_value.e)
btnE.grid(row=3, column=7, pady=1)

btnlog10 = Button(calc, text="log10", width=6,
              height=2, bg='light green', fg='black',
              font=('Helvetica', 20, 'bold'),
              bd=4, command=added_value.log10)
btnlog10.grid(row=4, column=4, pady=1)

btncos = Button(calc, text="log1p", width=6,
            height=2, bg='light green', fg='black',
            font=('Helvetica', 20, 'bold'),
            bd=4, command=added_value.log1p)
btncos.grid(row=4, column=5, pady=1)

btnexpm1 = Button(calc, text="expm1", width=6,
              height=2, bg='light green', fg='black',
              font=('Helvetica', 20, 'bold'),
              bd=4, command=added_value.expm1)
btnexpm1.grid(row=4, column=6, pady=1)

btngamma = Button(calc, text="gamma", width=6,
              height=2, bg='light green', fg='black',
              font=('Helvetica', 20, 'bold'),
```

```
              bd=4, command=added_value.lgamma)
btngamma.grid(row=4, column=7, pady=1)

btnlog2 = Button(calc, text="log2", width=6,
             height=2, bg='light green', fg='black',
             font=('Helvetica', 20, 'bold'),
             bd=4, command=added_value.log2)
btnlog2.grid(row=5, column=4, pady=1)

btndeg = Button(calc, text="deg", width=6,
            height=2, bg='light green', fg='black',
            font=('Helvetica', 20, 'bold'),
            bd=4, command=added_value.degrees)
btndeg.grid(row=5, column=5, pady=1)

btnacosh = Button(calc, text="acosh", width=6,
              height=2, bg='light green', fg='black',
              font=('Helvetica', 20, 'bold'),
                  bd=4, command=added_value.acosh)
btnacosh.grid(row=5, column=6, pady=1)

btnasinh = Button(calc, text="asinh", width=6,
              height=2, bg='light green', fg='black',
              font=('Helvetica', 20, 'bold'),
              bd=4, command=added_value.asinh)
btnasinh.grid(row=5, column=7, pady=1)

lblDisplay = Label(calc, text="Scientific Calculator",
               font=('Helvetica', 30, 'bold'),
                   bg='powder blue', fg='black', justify=CENTER)
lblDisplay.grid(row=0, column=4, columnspan=4)


def iExit():
    iExit = tkinter.messagebox.askyesno("Scientific Calculator",
                                        "Do you want to exit?")
    if iExit > 0:
        root.destroy()
        return


def Scientific():
    root.resizable(width=False, height=False)
    root.geometry("944x568+0+0")


def Standard():
    root.resizable(width=False, height=False)
    root.geometry("480x568+0+0")


menubar = Menu(calc)

filemenu = Menu(menubar, tearoff=0)
```

```
    menubar.add_cascade(label='File', menu=filemenu)
    filemenu.add_command(label="Standard", command=Standard)
    filemenu.add_command(label="Scientific", command=Scientific)
    filemenu.add_separator()
    filemenu.add_command(label="Exit", command=iExit)



    root.config(menu=menubar)

    root.mainloop()
button1 = Button(window, text = "open scientific calculator" ,
font=("helvetica",28),bg="powder blue",command=second_window)
button1.pack(pady = 20)
window.mainloop()
```