from tkinter import \*  
import math  
import parser  
import tkinter.messagebox  
  
root = Tk()  
root.title("Scientific Calculator")  
root.configure(background="light green")  
root.resizable(width=False, height=False)  
root.geometry("480x624+20+20")  
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)  
  
 def sum\_of\_total(self):  
 self.result = True  
 self.current = float(self.current)  
 if self.check\_sum == True:  
 self.valid\_function()  
 else:  
 self.total = float(txtDisplay.get())  
  
 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  
 if self.op == "inv":  
 self.total = 1 / self.current  
 self.input\_value = True  
 self.check\_sum = False  
 self.display(self.total)  
  
 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  
  
 def Clear\_Entry(self):  
 self.result = False  
 self.current = "0"  
 self.display(0)  
 self.input\_value = True  
  
 def all\_Clear\_Entry(self):  
 self.Clear\_Entry()  
 self.total = 0  
  
 def tanh(self):  
 self.reult = False  
 self.current = math.tanh(math.radians(float(txtDisplay.get())))  
 self.display(self.current)  
  
 def tan(self):  
 self.reult = False  
 self.current = math.tan(math.radians(float(txtDisplay.get())))  
 self.display(self.current)  
  
 def sinh(self):  
 self.reult = False  
 self.current = math.sinh(math.radians(float(txtDisplay.get())))  
 self.display(self.current)  
  
 def sin(self):  
 self.reult = False  
 self.current = math.sin(math.radians(float(txtDisplay.get())))  
 self.display(self.current)  
  
 def log(self):  
 self.reult = False  
 self.current = math.log(float(txtDisplay.get()))  
 self.display(self.current)  
  
 def exp(self):  
 self.reult = False  
 self.current = math.exp(float(txtDisplay.get()))  
 self.display(self.current)  
  
 def mathsPM(self):  
 self.reult = False  
 self.current = -(float(txtDisplay.get()))  
 self.display(self.current)  
  
 def squared(self):  
 self.reult = False  
 self.current = math.sqrt(float(txtDisplay.get()))  
 self.display(self.current)  
  
 def cos(self):  
 self.reult = False  
 self.current = math.cos(math.radians(float(txtDisplay.get())))  
 self.display(self.current)  
  
 def cosh(self):  
 self.reult = False  
 self.current = math.cosh(math.radians(float(txtDisplay.get())))  
 self.display(self.current)  
  
 def display(self, value):  
 txtDisplay.delete(0, END)  
 txtDisplay.insert(0, value)  
  
 def pi(self):  
 self.reult = False  
 self.current = math.pi  
 self.display(self.current)  
  
 def tau(self):  
 self.reult = False  
 self.current = math.tau  
 self.display(self.current)  
  
 def e(self):  
 self.reult = False  
 self.current = math.e  
 self.display(self.current)  
  
 def acosh(self):  
 self.result = False  
 self.current = math.acosh(float(txtDisplay.get()))  
 self.display(self.current)  
  
 def asinh(self):  
 self.result = False  
 self.current = math.asinh(float(txtDisplay.get()))  
 self.display(self.current)  
  
 def expm1(self):  
 self.result = False  
 self.current = math.expm1(float(txtDisplay.get()))  
 self.display(self.current)  
  
 def lgamma(self):  
 self.result = False  
 self.current = math.lgamma(float(txtDisplay.get()))  
 self.display(self.current)  
  
 def degrees(self):  
 self.result = False  
 self.current = math.degrees(float(txtDisplay.get()))  
 self.display(self.current)  
  
 def log2(self):  
 self.result = False  
 self.current = math.log2(float(txtDisplay.get()))  
 self.display(self.current)  
  
 def log10(self):  
 self.result = False  
 self.current = math.log10(float(txtDisplay.get()))  
 self.display(self.current)  
  
 def log1p(self):  
 self.result = False  
 self.current = math.log1p(float(txtDisplay.get()))  
 self.display(self.current)  
  
  
added\_value = Calc()  
txtDisplay = Entry(calc, relief=SUNKEN, font=('grotesque', 20, 'bold'), bg="light green", bd=30, width=28, justify=RIGHT)  
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, font=('grotesque', 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, font=('grotesque', 20, 'bold'), bd=4, bg="light green",  
 command=added\_value.Clear\_Entry).grid(row=1, column=0, pady=1)  
btnAllClear = Button(calc, text=chr(67) + chr(69), width=6, height=2, font=('grotesque', 20, 'bold'), bd=4,  
 bg="light green", command=added\_value.all\_Clear\_Entry).grid(row=1, column=1, pady=1)  
  
btnSq = Button(calc, text="√", width=6, height=2, font=('grotesque', 20, 'bold'), bd=4, bg="light green",  
 command=added\_value.squared).grid(row=1, column=2, pady=1)  
btnAdd = Button(calc, text="+", width=6, height=2, font=('grotesque', 20, 'bold'), bd=4, bg="light green",  
 command=lambda: added\_value.operation("add")).grid(row=1, column=3, pady=1)  
  
btnSub = Button(calc, text="-", width=6, height=2, font=('grotesque', 20, 'bold'), bd=4, bg="light green",  
 command=lambda: added\_value.operation("sub")).grid(row=2, column=3, pady=1)  
btnMult = Button(calc, text="×", width=6, height=2, font=('grotesque', 20, 'bold'), bd=4, bg="light green",  
 command=lambda: added\_value.operation("multi")).grid(row=3, column=3, pady=1)  
  
btnDiv = Button(calc, text=chr(247), width=6, height=2, font=('grotesque', 20, 'bold'), bd=4, bg="light green",  
 command=lambda: added\_value.operation("divide")).grid(row=4, column=3, pady=1)  
btnZero = Button(calc, text="0", width=6, height=2, font=('grotesque', 20, 'bold'), bd=4, bg="light green",  
 command=lambda: added\_value.numberEnter(0)).grid(row=5, column=0, pady=1)  
  
btnDot = Button(calc, text=".", width=6, height=2, font=('grotesque', 20, 'bold'), bd=4, bg="light green",  
 command=lambda: added\_value.numberEnter(".")).grid(row=5, column=1, pady=1)  
btnPM = Button(calc, text=chr(177), width=6, height=2, font=('grotesque', 20, 'bold'), bd=4, bg="light green",  
 command=added\_value.mathsPM).grid(row=5, column=2, pady=1)  
  
btnEquals = Button(calc, text="=", width=6, height=2, font=('grotesque', 20, 'bold'), bd=4, bg="light green",  
 command=added\_value.sum\_of\_total).grid(row=5, column=3, pady=1)  
  
btnPi = Button(calc, text='π', width=6, height=2, font=('grotesque', 20, 'bold'), bd=4, bg="Green",  
 command=added\_value.pi).grid(row=1, column=4, pady=1)  
btnCos = Button(calc, text="cos", width=6, height=2, font=('grotesque', 20, 'bold'), bd=4, bg="Green",  
 command=added\_value.cos).grid(row=1, column=5, pady=1)  
  
btnTan = Button(calc, text="tan", width=6, height=2, font=('grotesque', 20, 'bold'), bd=4, bg="Green",  
 command=added\_value.tan).grid(row=1, column=6, pady=1)  
btnSin = Button(calc, text="sin", width=6, height=2, font=('grotesque', 20, 'bold'), bd=4, bg="Green",  
 command=added\_value.sin).grid(row=1, column=7, pady=1)  
  
btn2Pi = Button(calc, text='2π', width=6, height=2, font=('grotesque', 20, 'bold'), bd=4, bg="Green",  
 command=added\_value.tau).grid(row=2, column=4, pady=1)  
btnCosh = Button(calc, text="cosh", width=6, height=2, font=('grotesque', 20, 'bold'), bd=4, bg="gray",  
 command=added\_value.cosh).grid(row=2, column=5, pady=1)  
  
btnTanh = Button(calc, text="tanh", width=6, height=2, font=('grotesque', 20, 'bold'), bd=4, bg="gray",  
 command=added\_value.tanh).grid(row=2, column=6, pady=1)  
btnSinh = Button(calc, text="sinh", width=6, height=2, font=('grotesque', 20, 'bold'), bd=4, bg="gray",  
 command=added\_value.sinh).grid(row=2, column=7, pady=1)  
  
btnLog = Button(calc, text='log', width=6, height=2, font=('grotesque', 20, 'bold'), bd=4, bg="Green",  
 command=added\_value.log).grid(row=3, column=4, pady=1)  
btninv = Button(calc, text="Inv", width=6, height=2, font=('grotesque', 20, 'bold'), bd=4, bg="gray",  
 command=lambda: added\_value.operation("inv")).grid(row=3, column=5, pady=1)  
  
btnMod = Button(calc, text="Mod", width=6, height=2, font=('grotesque', 20, 'bold'), bd=4,  
 command=lambda: added\_value.operation("mod")).grid(row=3, column=6, pady=1)  
btnE = Button(calc, text="e", width=6, height=2, font=('grotesque', 20, 'bold'), bd=4, bg="gray",  
 command=added\_value.e).grid(row=3, column=7, pady=1)  
  
btnLog2 = Button(calc, text='log2', width=6, height=2, font=('grotesque', 20, 'bold'), bd=4, bg="Green",  
 command=added\_value.log2).grid(row=4, column=4, pady=1)  
btnDeg = Button(calc, text="deg", width=6, height=2, font=('grotesque', 20, 'bold'), bd=4, bg="gray",  
 command=added\_value.degrees).grid(row=4, column=5, pady=1)  
  
btnAcosh = Button(calc, text="acosh", width=6, height=2, font=('grotesque', 20, 'bold'), bd=4, bg="gray",  
 command=added\_value.acosh).grid(row=4, column=6, pady=1)  
btnAsinh = Button(calc, text="asinh", width=6, height=2, font=('grotesque', 20, 'bold'), bd=4, bg="gray",  
 command=added\_value.asinh).grid(row=4, column=7, pady=1)  
  
btnLog10 = Button(calc, text='log10', width=6, height=2, font=('grotesque', 20, 'bold'), bd=4, bg="Green",  
 command=added\_value.log10).grid(row=5, column=4, pady=1)  
btnLog1p = Button(calc, text="log1p", width=6, height=2, font=('grotesque', 20, 'bold'), bd=4, bg="Green",  
 command=added\_value.log1p).grid(row=5, column=5, pady=1)  
  
btnExpm1 = Button(calc, text="expm1", width=6, height=2, font=('grotesque', 20, 'bold'), bd=4, bg="Green",  
 command=added\_value.expm1).grid(row=5, column=6, pady=1)  
btnLgamma = Button(calc, text="lgamma", width=6, height=2, font=('grotesque', 20, 'bold'), bd=4, bg="Green",  
 command=added\_value.lgamma).grid(row=5, column=7, pady=1)  
  
lblDisplay = Label(calc, text="Scientific Calculator", font=('grotesque', 30, 'bold'), justify=CENTER)  
lblDisplay.grid(row=0, column=4, columnspan=4)  
  
lblDisplay = Label(calc, text="NkC Calculator", font=('grotesque', 30, 'bold'), justify=CENTER)  
lblDisplay.grid(row=6, column=0, columnspan=4)  
  
  
def iExit():  
 iExit = tkinter.messagebox.askyesno("Scientific Calculator - NkC Configured", "Confirm if you want to exit")  
 if iExit > 0:  
 root.destroy()  
 return  
  
  
def Scientific():  
 root.resizable(width=False, height=False)  
 root.geometry("944x624+20+20")  
  
  
def Standard():  
 root.resizable(width=False, height=False)  
 root.geometry("480x624+20+20")  
  
  
menubar = Menu(calc)  
  
filemenu = Menu(menubar, tearoff=0)  
menubar.add\_cascade(label="File", menu=filemenu)  
filemenu.add\_command(label="Standadrd", 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()