import tkinter as tk

import math

root = tk.Tk()

root.title('Scientific Calculator')

root.configure(bg='#FFEBCD')

root.resizable(width=False, height=False)

ent\_field = tk.Entry(root, bg='#D3D3D3', fg='#000080', font=('Arial', 35),

borderwidth=5, justify="right")

ent\_field.grid(row=0, columnspan=10, padx=5, pady=5,

sticky='n'+'s'+'e'+'w')

ent\_field.insert(0, '0')

FONT = ('Arial', 20, 'bold')

class SC\_Calculator():

def \_init\_(self):

self.current = ''

self.inp\_value = True

self.result = False

def Entry(self, value):

ent\_field.delete(0, 'end')

ent\_field.insert(0, value)

def Enter\_Num(self, num):

self.result = False

firstnum = ent\_field.get()

secondnum = str(num)

if self.inp\_value:

self.current = secondnum

self.inp\_value = False

else:

self.current = firstnum+secondnum

self.Entry(self.current)

def Standard\_Ops(self, val):

temp\_str = ent\_field.get()

try:

if val == '=':

ans = str(eval(temp\_str))

self.result = True

self.Entry(ans)

else:

self.Entry(temp\_str+val)

self.inp\_value = False

except ValueError:

self.Entry('Error')

except SyntaxError:

self.Entry('Error')

def Clear\_Entry(self):

self.result = False

self.current = "0"

self.Entry(0)

self.inp\_value = True

def SQ\_Root(self):

try:

self.current = math.sqrt(float(ent\_field.get()))

self.Entry(self.current)

except ValueError:

self.Entry('Error')

except SyntaxError:

self.Entry('Error')

def Pi(self):

self.result = False

self.current = math.pi

self.Entry(self.current)

def E(self):

self.result = False

self.current = math.e

self.Entry(self.current)

def Deg(self):

try:

self.result = False

self.current = math.degrees(float(ent\_field.get()))

self.Entry(self.current)

except ValueError:

self.Entry('Error')

except SyntaxError:

self.Entry('Error')

def Rad(self):

try:

self.result = False

self.current = math.radians(float(ent\_field.get()))

self.Entry(self.current)

except ValueError:

self.Entry('Error')

except SyntaxError:

self.Entry('Error')

def Exp(self):

try:

self.result = False

self.current = math.exp(float(ent\_field.get()))

self.Entry(self.current)

except ValueError:

self.Entry('Error')

except SyntaxError:

self.Entry('Error')

def Fact(self):

try:

self.result = False

self.current = math.factorial(float(ent\_field.get()))

self.Entry(self.current)

except ValueError:

self.Entry('Error')

except SyntaxError:

self.Entry('Error')

def Sin(self):

try:

self.result = False

self.current = math.sin(math.radians(float(ent\_field.get())))

self.Entry(self.current)

except ValueError:

self.Entry('Error')

except SyntaxError:

self.Entry('Error')

def Cos(self):

try:

self.result = False

self.current = math.cos(math.radians(float(ent\_field.get())))

self.Entry(self.current)

except ValueError:

self.Entry('Error')

except SyntaxError:

self.Entry('Error')

def Tan(self):

try:

self.result = False

self.current = math.tan(math.radians(float(ent\_field.get())))

self.Entry(self.current)

except ValueError:

self.Entry('Error')

except SyntaxError:

self.Entry('Error')

def Sinh(self):

try:

self.result = False

self.current = math.sinh(math.radians(float(ent\_field.get())))

self.Entry(self.current)

except ValueError:

self.Entry('Error')

except SyntaxError:

self.Entry('Error')

def Cosh(self):

try:

self.result = False

self.current = math.cosh(math.radians(float(ent\_field.get())))

self.Entry(self.current)

except ValueError:

self.Entry('Error')

except SyntaxError:

self.Entry('Error')

def Tanh(self):

try:

self.result = False

self.current = math.tanh(math.radians(float(ent\_field.get())))

self.Entry(self.current)

except ValueError:

self.Entry('Error')

except SyntaxError:

self.Entry('Error')

def Ln(self):

try:

self.result = False

self.current = math.log(float(ent\_field.get()))

self.Entry(self.current)

except ValueError:

self.Entry('Error')

except SyntaxError:

self.Entry('Error')

def Log\_10(self):

try:

self.result = False

self.current = math.log10(float(ent\_field.get()))

self.Entry(self.current)

except ValueError:

self.Entry('Error')

except SyntaxError:

self.Entry('Error')

def Log\_2(self):

try:

self.result = False

self.current = math.log2(float(ent\_field.get()))

self.Entry(self.current)

except ValueError:

self.Entry('Error')

except SyntaxError:

self.Entry('Error')

def Pow\_2(self):

try:

self.result = False

self.current = int(ent\_field.get())\*\*2

self.Entry(self.current)

except ValueError:

self.Entry('Error')

except SyntaxError:

self.Entry('Error')

def Pow\_3(self):

try:

self.result = False

self.current = int(ent\_field.get())\*\*3

self.Entry(self.current)

except ValueError:

self.Entry('Error')

except SyntaxError:

self.Entry('Error')

def n\_floor(self):

try:

self.result = False

self.current = math.floor(float(ent\_field.get()))

self.Entry(self.current)

except ValueError:

self.Entry('Error')

except SyntaxError:

self.Entry('Error')

def n\_ceil(self):

try:

self.result = False

self.current = math.ceil(float(ent\_field.get()))

self.Entry(self.current)

except ValueError:

self.Entry('Error')

except SyntaxError:

self.Entry('Error')

def Abs(self):

try:

self.result = False

self.current = abs(float(ent\_field.get()))

self.Entry(self.current)

except ValueError:

self.Entry('Error')

except SyntaxError:

self.Entry('Error')

numberpad = "789456123"

i = 0

button = []

for j in range(2, 5):

for k in range(3):

button.append(tk.Button(root, text=numberpad[i], font=FONT,

fg="#000000", width=6, height=2,

highlightbackground='#ADD8E6', highlightthickness=2))

button[i].grid(row=j, column=k, sticky='n' +

's'+'e' + 'w', padx=3, pady=3)

button[i]["command"] = lambda x=numberpad[i]: sc\_app.Enter\_Num(x)

i += 1

btn\_CE = tk.Button(root, text='CE', command=lambda: sc\_app.Clear\_Entry(),

font=FONT, height=2, fg="#000000",

highlightbackground='#ADD8E6', highlightthickness=2)

btn\_CE.grid(row=1, column=0, columnspan=2,

sticky='n'+'s'+'e'+'w', padx=3, pady=3)

btn\_sqr = tk.Button(root, text='\u221A', command=lambda: sc\_app.SQ\_Root(),

font=FONT, width=6, height=2, fg="#000000",

highlightbackground='#ADD8E6', highlightthickness=2)

btn\_sqr.grid(row=1, column=2, sticky='n'+'s'+'e'+'w', padx=3, pady=3)

btn\_0 = tk.Button(root, text='0', command=lambda: sc\_app.Enter\_Num('0'),

font=FONT, width=6, height=2, fg="#000000",

highlightbackground='#ADD8E6', highlightthickness=2)

btn\_0.grid(row=5, column=0, columnspan=2,

sticky='n'+'s'+'e'+'w', padx=3, pady=3)

btn\_point = tk.Button(root, text='.', command=lambda: sc\_app.Standard\_Ops('.'),

font=FONT, width=6, height=2, fg="#000000",

highlightbackground='#ADD8E6', highlightthickness=2)

btn\_point.grid(row=5, column=2, sticky='n'+'s'+'e'+'w', padx=3, pady=3)

btn\_equal = tk.Button(root, text='=', command=lambda: sc\_app.Standard\_Ops('='),

font=FONT, width=6, height=2, fg="#000000",

highlightbackground='#ADD8E6', highlightthickness=2)

btn\_equal.grid(row=5, column=3, sticky='n'+'s'+'e'+'w', padx=3, pady=3)

btn\_add = tk.Button(root, text='+', command=lambda: sc\_app.Standard\_Ops('+'),

font=FONT, width=6, height=2, fg="#000000",

highlightbackground='#ADD8E6', highlightthickness=2)

btn\_add.grid(row=1, column=3, sticky='n'+'s'+'e'+'w', padx=3, pady=3)

btn\_sub = tk.Button(root, text='-', command=lambda: sc\_app.Standard\_Ops('-'),

font=FONT, width=6, height=2, fg="#000000",

highlightbackground='#ADD8E6', highlightthickness=2)

btn\_sub.grid(row=2, column=3, sticky='n'+'s'+'e'+'w', padx=3, pady=3)

btn\_mul = tk.Button(root, text='\*', command=lambda: sc\_app.Standard\_Ops('\*'),

font=FONT, width=6, height=2, fg="#000000",

highlightbackground='#ADD8E6', highlightthickness=2)

btn\_mul.grid(row=3, column=3, sticky='n'+'s'+'e'+'w', padx=3, pady=3)

btn\_div = tk.Button(root, text='/', command=lambda: sc\_app.Standard\_Ops('/'),

font=FONT, width=6, height=2, fg="#000000",

highlightbackground='#ADD8E6', highlightthickness=2)

btn\_div.grid(row=4, column=3, sticky='n'+'s'+'e'+'w', padx=3, pady=3)

btn\_pi = tk.Button(root, text='\u03C0', command=lambda: sc\_app.Pi(),

font=FONT, width=5, height=2, fg="#000000",

highlightbackground='#ADD8E6', highlightthickness=2)

btn\_pi.grid(row=1, column=4, sticky='n'+'s'+'e'+'w', padx=3, pady=3)

btn\_e = tk.Button(root, text='e', command=lambda: sc\_app.E(),

font=FONT, width=5, height=2, fg="#000000",

highlightbackground='#ADD8E6', highlightthickness=2)

btn\_e.grid(row=1, column=5, sticky='n'+'s'+'e'+'w', padx=3, pady=3)

btn\_deg = tk.Button(root, text='Deg', command=lambda: sc\_app.Deg(),

font=FONT, width=5, height=2, fg="#000000",

highlightbackground='#ADD8E6', highlightthickness=2)

btn\_deg.grid(row=1, column=6, sticky='n'+'s'+'e'+'w', padx=3, pady=3)

btn\_exp = tk.Button(root, text='Exp', command=lambda: sc\_app.Exp(),

font=FONT, width=5, height=2, fg="#000000",

highlightbackground='#ADD8E6', highlightthickness=2)

btn\_exp.grid(row=2, column=4, sticky='n'+'s'+'e'+'w', padx=3, pady=3)

btn\_fact = tk.Button(root, text='x!', command=lambda: sc\_app.Fact(),

font=FONT, width=5, height=2, fg="#000000",

highlightbackground='#ADD8E6', highlightthickness=2)

btn\_fact.grid(row=2, column=5, sticky='n'+'s'+'e'+'w', padx=3, pady=3)

btn\_rad = tk.Button(root, text='Rad', command=lambda: sc\_app.Rad(),

font=FONT, width=5, height=2, fg="#000000",

highlightbackground='#ADD8E6', highlightthickness=2)

btn\_rad.grid(row=2, column=6, sticky='n'+'s'+'e'+'w', padx=3, pady=3)

btn\_sin = tk.Button(root, text='sin', command=lambda: sc\_app.Sin(),

font=FONT, width=5, height=2, fg="#000000",

highlightbackground='#ADD8E6', highlightthickness=2)

btn\_sin.grid(row=3, column=4, sticky='n'+'s'+'e'+'w', padx=3, pady=3)

btn\_cos = tk.Button(root, text='cos', command=lambda: sc\_app.Cos(),

font=FONT, width=5, height=2, fg="#000000",

highlightbackground='#ADD8E6', highlightthickness=2)

btn\_cos.grid(row=3, column=5, sticky='n'+'s'+'e'+'w', padx=3, pady=3)

btn\_tan = tk.Button(root, text='tan', command=lambda: sc\_app.Tan(),

font=FONT, width=5, height=2, fg="#000000",

highlightbackground='#ADD8E6', highlightthickness=2)

btn\_tan.grid(row=3, column=6, sticky='n'+'s'+'e'+'w', padx=3, pady=3)

btn\_sinh = tk.Button(root, text='sinh', command=lambda: sc\_app.Sinh(),

font=FONT, width=5, height=2, fg="#000000",

highlightbackground='#ADD8E6', highlightthickness=2)

btn\_sinh.grid(row=4, column=4, sticky='n'+'s'+'e'+'w', padx=3, pady=3)

btn\_cosh = tk.Button(root, text='cosh', command=lambda: sc\_app.Cosh(),

font=FONT, width=5, height=2, fg="#000000",

highlightbackground='#ADD8E6', highlightthickness=2)

btn\_cosh.grid(row=4, column=5, sticky='n'+'s'+'e'+'w', padx=3, pady=3)

btn\_tanh = tk.Button(root, text='tanh', command=lambda: sc\_app.Tanh(),

font=FONT, width=5, height=2, fg="#000000",

highlightbackground='#ADD8E6', highlightthickness=2)

btn\_tanh.grid(row=4, column=6, sticky='n'+'s'+'e'+'w', padx=3, pady=3)

btn\_ln = tk.Button(root, text='ln', command=lambda: sc\_app.Ln(),

font=FONT, width=5, height=2, fg="#000000",

highlightbackground='#ADD8E6', highlightthickness=2)

btn\_ln.grid(row=5, column=4, sticky='n'+'s'+'e'+'w', padx=3, pady=3)

btn\_log2 = tk.Button(root, text='log2', command=lambda: sc\_app.Log\_2(),

font=FONT, width=5, height=2, fg="#000000",

highlightbackground='#ADD8E6', highlightthickness=2)

btn\_log2.grid(row=5, column=5, sticky='n'+'s'+'e'+'w', padx=3, pady=3)

btn\_log10 = tk.Button(root, text='log10', command=lambda: sc\_app.Log\_10(),

font=FONT, width=5, height=2, fg="#000000",

highlightbackground='#ADD8E6', highlightthickness=2)

btn\_log10.grid(row=5, column=6, sticky='n'+'s'+'e'+'w', padx=3, pady=3)

btn\_x\_pow2 = tk.Button(root, text='x\u00B2', command=lambda: sc\_app.Pow\_2(),

font=FONT, width=5, height=2, fg="#000000",

highlightbackground='#ADD8E6', highlightthickness=2)

btn\_x\_pow2.grid(row=1, column=7, sticky='n'+'s'+'e'+'w', padx=3, pady=3)

btn\_x\_pow3 = tk.Button(root, text='x\u00B3', command=lambda: sc\_app.Pow\_3(),

font=FONT, width=5, height=2, fg="#000000",

highlightbackground='#ADD8E6', highlightthickness=2)

btn\_x\_pow3.grid(row=2, column=7, sticky='n'+'s'+'e'+'w', padx=3, pady=3)

btn\_x\_floor= tk.Button(root, text='floor', command=lambda: sc\_app.n\_floor(),

font=FONT, width=5, height=2, fg="#000000",

highlightbackground='#ADD8E6', highlightthickness=2)

btn\_x\_floor.grid(row=3, column=7, sticky='n'+'s'+'e'+'w', padx=3, pady=3)

btn\_x\_ceil = tk.Button(root, text='ceil', command=lambda: sc\_app.n\_ceil(),

font=FONT, width=5, height=2, fg="#000000",

highlightbackground='#ADD8E6', highlightthickness=2)

btn\_x\_ceil.grid(row=4, column=7, sticky='n'+'s'+'e'+'w', padx=3, pady=3)

btn\_abs = tk.Button(root, text='Abs', command=lambda: sc\_app.Abs(),

font=FONT, width=5, height=2, fg="#000000",

highlightbackground='#ADD8E6', highlightthickness=2)

btn\_abs.grid(row=5, column=7, sticky='n'+'s'+'e'+'w', padx=3, pady=3)

if \_\_name\_\_ == '\_\_main\_\_':

sc\_app = SC\_Calculator()

root.mainloop()