**Code**

#XOR operation - Uday

from tkinter import \*

def xor\_op(a = 0, b = 0) -> int:

    if(not (a.isnumeric()) or not (b.isnumeric())):

        return str("Invalid Input")

    return int(int(a) ^ int(b))

def xor\_show(a = 0, b = 0) -> int:

    if(not (a.isnumeric()) or not (b.isnumeric())):

        return str("Invalid Input")

    return str(bin(int(a) ^ int(b)))

def xor\_info(a = 2, b = 3) -> None:

    c = convbinary(int(a) ^ int(b))

    a = convbinary(int(a))

    b = convbinary(int(b))

    tr = Tk()

    #print(a)

    #print(b)

    tr.geometry("800x500+100+100")

    tr.resizable(False, False)

    #if(not (a.isnumeric()) or not (b.isnumeric())):

    #    tr.destroy()

    x = StringVar(tr)

    y = StringVar(tr)

    z = StringVar(tr)

    x.set(a)

    y.set(b)

    z.set(c)

    #print(x)

    #print(y)

    gridx = Frame(tr)

    Label(gridx, text = 'More Information', pady = '10', padx = '10', relief = 'raised', fg = 'Red',bg = '#ADD8E9', font = ('helvetica', '35')).grid()

    gridx.pack(pady = '20')

    gridy = Frame(tr)

    Label(gridy, text = 'Number A : ', pady = '10', padx = '10', relief = 'ridge', fg = '#ADD8E9', font = ('helvetica', '25'), width = '15', height = '1').grid(column = 0, row = 0)

    Label(gridy, textvariable = x, pady = '10', padx = '10', relief = 'ridge', fg = '#ADD8E9', font = ('helvetica', '25'), width = '15', height = '1').grid(column = 1, row = 0)

    Label(gridy, text = 'Number B : ', pady = '10', padx = '10', relief = 'ridge', fg = '#ADD8E9', font = ('helvetica', '25'), width = '15', height = '1').grid(column = 0, row = 1)

    Label(gridy, textvariable = y, pady = '10', padx = '10', relief = 'ridge', fg = '#ADD8E9', font = ('helvetica', '25'), width = '15', height = '1').grid(column = 1, row = 1)

    Label(gridy, text = '   XOR :      ', pady = '10', padx = '10', relief = 'ridge', fg = '#ADD8E9', font = ('helvetica', '25'), width = '15', height = '1').grid(column = 0, row = 2)

    Label(gridy, textvariable = z, pady = '10', padx = '10', relief = 'ridge', fg = '#ADD8E9', font = ('helvetica', '25'), width = '15', height = '1').grid(column = 1, row = 2)

    gridy.pack(padx = '50', pady = '90')

    #tr.after(2, tr.update)

    tr.mainloop()

def convbinary(a = 0) -> str: #Aniruddha

    temp = a

    chr = str("")

    if(a == 0):

        return "0000 "

    while(temp):

        chr = chr + str(temp % 2)

        #print(temp % 2)

        temp = temp // 2

    if(len(chr) <= 3):

        chr = chr + "0"\*(4 - len(chr))

    #print(chr)

    return chr[::-1]

#print(convbinary(1))

#xor\_info(1, 1)

#AND operation - Uday

from tkinter import \*

from XOR import \*

def and\_op(a = 0, b = 0) -> int:

    if(not (a.isnumeric()) or not (b.isnumeric())):

        return str("Invalid Input")

    return int(int(a) & int(b))

def and\_show(a = 0, b = 0) -> int:

    if(not (a.isnumeric()) or not (b.isnumeric())):

        return str("Invalid Input")

    return str(bin(int(a) & int(b)))

def and\_info(a = 0, b = 0):

    tr = Tk()

    a = str(a)

    b = str(b)

    tr.geometry("800x500+100+100")

    tr.resizable(False, False)

    if(not (a.isnumeric()) or not (b.isnumeric())):

        tr.destroy()

    x = StringVar(tr)

    y = StringVar(tr)

    z = StringVar(tr)

    x.set(convbinary(int(a)))

    y.set(convbinary(int(b)))

    z.set(convbinary(int(a) & int(b)))

    #print(x)

    gridx = Frame(tr)

    Label(gridx, text = 'More Information', pady = '10', padx = '10', relief = 'raised', fg = 'Red',bg = '#ADD8E9', font = ('helvetica', '35')).grid()

    gridx.pack(pady = '20')

    gridy = Frame(tr)

    Label(gridy, text = 'Number A : ', pady = '10', padx = '10', relief = 'ridge', fg = '#ADD8E9', font = ('helvetica', '25'), width = '15', height = '1').grid(column = 0, row = 0)

    Label(gridy, textvariable = x, pady = '10', padx = '10', relief = 'ridge', fg = '#ADD8E9', font = ('helvetica', '25'), width = '15', height = '1').grid(column = 1, row = 0)

    Label(gridy, text = 'Number B : ', pady = '10', padx = '10', relief = 'ridge', fg = '#ADD8E9', font = ('helvetica', '25'), width = '15', height = '1').grid(column = 0, row = 1)

    Label(gridy, textvariable = y, pady = '10', padx = '10', relief = 'ridge', fg = '#ADD8E9', font = ('helvetica', '25'), width = '15', height = '1').grid(column = 1, row = 1)

    Label(gridy, text = '   AND :       ', pady = '10', padx = '10', relief = 'ridge', fg = '#ADD8E9', font = ('helvetica', '25'), width = '15', height = '1').grid(column = 0, row = 2)

    Label(gridy, textvariable = z, pady = '10', padx = '10', relief = 'ridge', fg = '#ADD8E9', font = ('helvetica', '25'), width = '15', height = '1').grid(column = 1, row = 2)

    gridy.pack(padx = '50', pady = '90')

    tr.mainloop()

#and\_info(20, 30)

#OR operation - Karthik

from tkinter import \*

from XOR import \*

def or\_op(a = 0, b = 0) -> int:

    if(not (a.isnumeric()) or not (b.isnumeric())):

        return str("Invalid Input")

    return int(int(a) | int(b))

def or\_show(a = 0, b = 0) -> int:

    if(not (a.isnumeric()) or not (b.isnumeric())):

        return str("Invalid Input")

    return str(bin(int(a) | int(b)))

def or\_info(a = 0, b = 0):

    tr = Tk()

    a = str(a)

    b = str(b)

    tr.geometry("800x500+100+100")

    tr.resizable(False, False)

    if(not (a.isnumeric()) or not (b.isnumeric())):

        tr.destroy()

    x = StringVar(tr)

    y = StringVar(tr)

    z = StringVar(tr)

    x.set(convbinary(int(a)))

    y.set(convbinary(int(b)))

    z.set(convbinary(int(a) | int(b)))

    #print(x)

    gridx = Frame(tr)

    Label(gridx, text = 'More Information', pady = '10', padx = '10', relief = 'raised', fg = 'Red',bg = '#ADD8E9', font = ('helvetica', '35')).grid()

    gridx.pack(pady = '20')

    gridy = Frame(tr)

    Label(gridy, text = 'Number A : ', pady = '10', padx = '10', relief = 'ridge', fg = '#ADD8E9', font = ('helvetica', '25'), width = '15', height = '1').grid(column = 0, row = 0)

    Label(gridy, textvariable = x, pady = '10', padx = '10', relief = 'ridge', fg = '#ADD8E9', font = ('helvetica', '25'), width = '15', height = '1').grid(column = 1, row = 0)

    Label(gridy, text = 'Number B : ', pady = '10', padx = '10', relief = 'ridge', fg = '#ADD8E9', font = ('helvetica', '25'), width = '15', height = '1').grid(column = 0, row = 1)

    Label(gridy, textvariable = y, pady = '10', padx = '10', relief = 'ridge', fg = '#ADD8E9', font = ('helvetica', '25'), width = '15', height = '1').grid(column = 1, row = 1)

    Label(gridy, text = '   OR :        ', pady = '10', padx = '10', relief = 'ridge', fg = '#ADD8E9', font = ('helvetica', '25'), width = '15', height = '1').grid(column = 0, row = 2)

    Label(gridy, textvariable = z, pady = '10', padx = '10', relief = 'ridge', fg = '#ADD8E9', font = ('helvetica', '25'), width = '15', height = '1').grid(column = 1, row = 2)

    gridy.pack(padx = '50', pady = '90')

    tr.mainloop()

#or\_info(20, 30)

#Not operation - Karthik

from tkinter import \*

from XOR import \*

def not\_op(a = 0, b = 0) -> str:

    if(not (a.isnumeric()) or not (b.isnumeric())):

        return str("Invalid Input")

    a = int(a)

    b = int(b)

    return str((-a, -b))

def not\_show(a = 0, b = 0) -> str:

    if(not (a.isnumeric()) or not (b.isnumeric())):

        return str("Invalid Input")

    a = int(a)

    b = int(b)

    return str((bin(-a), bin(-b)))

def not\_info(a = 0, b = 0):

    tr = Tk()

    a = str(a)

    b = str(b)

    tr.geometry("800x500+100+100")

    tr.resizable(False, False)

    if(not (a.isnumeric()) or not (b.isnumeric())):

        tr.destroy()

    x = StringVar(tr)

    y = StringVar(tr)

    z = StringVar(tr)

    x.set(convbinary(int(a)))

    y.set(convbinary(int(b)))

    z.set(('-' + convbinary(int(a)), '-' + convbinary(int(b))))

    #print(x)

    gridx = Frame(tr)

    Label(gridx, text = 'More Information', pady = '10', padx = '10', relief = 'raised', fg = 'Red',bg = '#ADD8E9', font = ('helvetica', '35')).grid()

    gridx.pack(pady = '20')

    gridy = Frame(tr)

    Label(gridy, text = 'Number A : ', pady = '10', padx = '10', relief = 'ridge', fg = '#ADD8E9', font = ('helvetica', '25'), width = '15', height = '1').grid(column = 0, row = 0)

    Label(gridy, textvariable = x, pady = '10', padx = '10', relief = 'ridge', fg = '#ADD8E9', font = ('helvetica', '25'), width = '15', height = '1').grid(column = 1, row = 0)

    Label(gridy, text = 'Number B : ', pady = '10', padx = '10', relief = 'ridge', fg = '#ADD8E9', font = ('helvetica', '25'), width = '15', height = '1').grid(column = 0, row = 1)

    Label(gridy, textvariable = y, pady = '10', padx = '10', relief = 'ridge', fg = '#ADD8E9', font = ('helvetica', '25'), width = '15', height = '1').grid(column = 1, row = 1)

    Label(gridy, text = '   NOT :      ', pady = '10', padx = '10', relief = 'ridge', fg = '#ADD8E9', font = ('helvetica', '25'), width = '15', height = '1').grid(column = 0, row = 2)

    Label(gridy, textvariable = z, pady = '10', padx = '10', relief = 'ridge', fg = '#ADD8E9', font = ('helvetica', '25'), width = '15', height = '1').grid(column = 1, row = 2)

    gridy.pack(padx = '50', pady = '90')

    tr.mainloop()

#not\_info(8, 10)

#Tkinter/main driver code - Aniruddha

from XOR import \*

from AND import \*

from OR import \*

from NOT import \*

def XOR\_2():

    x1 = str(xor\_show(q1.get(), q2.get()))

    z.set(x1)

    x.set(bin(int(q1.get())))

    y1.set(bin(int(q2.get())))

    info.set("XOR operation is done to find difference between values, here each bit is operated on, which has 4 outcomes\n1 ^ 0 = 0\n0 ^ 1 = 0\n1 ^ 1 = 1\n0 ^ 0 = 1")

def AND\_2():

    x1 = str(and\_show(q1.get(), q2.get()))

    z.set(x1)

    x.set(bin(int(q1.get())))

    y1.set(bin(int(q2.get())))

    info.set("AND operation is done to find whether if both values are 1, here each bit is operated on, this has 4 outcomes\n1 & 0 = 0\n0 & 1 = 1\n1 & 1 = 1\n0 & 0 = 0")

def NOT\_2():

    x1 = str(not\_show(q1.get(), q2.get()))

    z.set(x1)

    x.set(bin(int(q1.get())))

    y1.set(bin(int(q2.get())))

    info.set("NOT operation is done to negate a value, here each bit is operated on and essentially forms the 1's complement of the same, this has only 2 outcomes\n~1 = 0\n~0 = 1")

def OR\_2():

    x1 = str(or\_show(q1.get(), q2.get()))

    z.set(x1)

    x.set(bin(int(q1.get())))

    y1.set(bin(int(q2.get())))

    info.set("OR operation is done to find if atleast 1 bit is 1, here each bit is operated on, which has only 4 outcomes\n1 | 0 = 1\n1 | 0 = 1\n0 | 0 = 0\n1 | 1 = 1")

choicer = int()

def chosen():

    global choicer

    global q1

    global q2

    #print(q1.get())

    #print(choicer)

    if(choicer == 1):

        xor\_info(q1.get(), q2.get())

        #print("1")

    elif(choicer == 2):

        and\_info(int(q1.get()), int(q2.get()))

    elif(choicer == 3):

        not\_info(int(q1.get()), int(q2.get()))

    elif(choicer == 4):

        or\_info(int(q1.get()), int(q2.get()))

def XOR\_1():

    x1 = str(xor\_op(q1.get(),q2.get()))

    m.set(x1)

    XOR\_2()

    global choicer

    choicer = 1

def AND\_1():

    x1 = str(and\_op(q1.get(), q2.get()))

    m.set(x1)

    AND\_2()

    global choicer

    choicer = 2

def NOT\_1():

    x1 = (not\_op(q1.get(), q2.get()))

    m.set(x1)

    NOT\_2()

    global choicer

    choicer = 3

def OR\_1():

    x1 = str(or\_op(q1.get(), q2.get()))

    m.set(x1)

    OR\_2()

    global choicer

    choicer = 4

root = Tk()

m = StringVar()

x = StringVar()

y1 = StringVar()

z = StringVar()

info = StringVar()

root.title("Calc")

root.geometry('1280x800+20+25')

frame = Frame(root)

#frame.grid()

Label(frame, text = "Bitwise Operations Calculator", font = ('Helvetica', 25), pady = '30', fg = 'Red').pack()

frame.pack()

frame1 = Frame(root)

Label(frame1, text = "Enter No. 1", font = ('Times New Roman', 20), pady = '15', padx = '80', fg = 'Blue').pack(side = 'left')

Label(frame1, text = "Enter No. 2", font = ('Times New Roman', 20), pady = '15', padx = '80', fg = 'Blue').pack(side = 'right')

frame1.pack()

frame2 = Frame(root)

q1 = Entry(frame2, bg = "#FFCCCB", relief = "raised", width = '30', font = ('Helvetica', '15'))

q2 = Entry(frame2, bg = "#FFCCCB", relief = "raised", width = '30', font = ('Helvetica', '15'))

q1.pack(side = 'left', pady = '25', padx = '50')

q2.pack(side = 'right', pady = '25', padx = '50')

frame2.pack()

frame3 = Frame(root)

Label(frame3, text = 'Choose your operation', pady = '15', fg = "Red", font = ('Times New Roman', '15')).pack(side = 'top')

frame3.pack()

frame4 = Frame(root)

Button(frame4, text = "XOR",bg = '#ADD8E6', padx = '25', pady = '20', command = XOR\_1, height = ' 1', width = '10').pack(side = 'left', padx = '25')

Button(frame4, text = "AND",bg = '#ADD8E6', padx = '25', pady = '20', command = AND\_1, height = ' 1', width = '10').pack(side = 'left', padx = '25')

Button(frame4, text = "OR",bg = '#ADD8E6', padx = '25', pady = '20', command = OR\_1, height = ' 1', width = '10').pack(side = 'right', padx = '25')

Button(frame4, text = "NOT",bg = '#ADD8E6', padx = '25', pady = '20', command = NOT\_1, height = ' 1', width = '10').pack(side = 'right', padx = '25')

frame4.pack(pady = '25')

resultfrm = Frame(root)

Label(resultfrm,text = 'Result := ', font = ('Helvetica', 25), padx = '20', pady = '20', fg = '#ADD8E9', bg = 'Red', width = '5').pack(side = 'left')

Label(resultfrm,textvariable = m, font = ('Helvetica', 25), padx = '20', pady = '20', fg = '#ADD8E9', bg = 'Red', width = '5').pack(side = 'right')

resultfrm.pack(side = 'left', padx = '25', pady = '15')

explainfrm = Frame(root)

Label(explainfrm, text = 'No. 1 :', padx = '10', pady = '10', font = ('helvetica', 15), bg = '#ADD8E9', width = '12').grid(column = 0, row = 0)

Label(explainfrm, textvariable = x, padx = '10', pady = '10', font = ('helvetica', 15), bg = '#ADD8E9', width = '12').grid(column = 0, row = 1)

Label(explainfrm, text = 'No. 2', padx = '10', pady = '10', font = ('helvetica', 15), bg = '#ADD8E9', width = '12').grid(column = 1, row = 0)

Label(explainfrm, textvariable = y1, padx = '10', pady = '10', font = ('helvetica', 15), bg = '#ADD8E9', width = '12').grid(column = 1, row = 1)

Label(explainfrm, text = 'Result', padx = '10', pady = '10', font = ('helvetica', 15), bg = '#ADD8E9', width = '12').grid(column = 2, row = 0)

Label(explainfrm, textvariable = z, padx = '10', pady = '10', font = ('helvetica', 15), bg = '#ADD8E9', width = '12').grid(column = 2, row = 1)

explainfrm.pack(side = 'right', padx = '15', pady = '15')

def infy():

    inf = Tk()

    inf.title("Explanation")

    inf.geometry("500x500+100+100")

    xc = info

    Label(inf, textvariable = xc, pady = '10', padx = '10',bg = '#ADD8E9', font = ('helvetica', 10)).pack(side = 'top')

    inf.resizable(False, False)

    inf.mainloop()

expl = Frame(root)

Button(expl, text = "Information", pady = '10', padx = '10',bg = '#ADD8E9', font = ('helvetica', 10), command = chosen, width = '15', height = '2').pack()

expl.pack(side = 'top',padx = '15', pady = '15')

root.resizable(True, True)

root.mainloop()