

Index	Binary	Char	Index	Binary	Char	Index	Binary	Char	Index	Binary	Char
0	000000	A	16	010000	Q	32	100000	g	48	110000	w
1	000001	B	17	010001	R	33	100001	h	49	110001	x
2	000010	C	18	010010	S	34	100010	i	50	110010	y
3	000011	D	19	010011	T	35	100011	j	51	110011	z
4	000100	E	20	010100	U	36	100100	k	52	110100	ø
5	000101	F	21	010101	V	37	100101	l	53	110101	1
6	000110	G	22	010110	W	38	100110	m	54	110110	2
7	000111	H	23	010111	X	39	100111	n	55	110111	3
8	001000	I	24	011000	Y	40	101000	o	56	111000	4
9	001001	J	25	011001	Z	41	101001	p	57	111001	5
10	001010	K	26	011010	a	42	101010	q	58	111010	6
11	001011	L	27	011011	b	43	101011	r	59	111011	7
12	001100	M	28	011100	c	44	101100	s	60	111100	8
13	001101	N	29	011101	d	45	101101	t	61	111101	9
14	001110	O	30	011110	e	46	101110	u	62	111110	+
15	001111	P	31	011111	f	47	101111	v	63	111111	/

```

import random

while True:
    print("1. Rock, 2. Scissors, 3. Paper")
    user_input = int(input("Enter the number"))

    while user_input > 3 or user_input < 1:
        user_input = int(input("Please enter number between 1-3"))

    if user_input == 1:
        user_input_choice = 'Rock'
    elif user_input == 2:
        user_input_choice = 'Scissors'
    else:
        user_input_choice = 'Paper'

    print("User has selected",user_input_choice)

    device_choice = random.randint(1,3)

    while user_input == device_choice:
        device_choice = random.randint(1, 3)

    if device_choice == 1:
        device_input_choice = 'Rock'
    elif device_choice == 2:
        device_input_choice = 'Scissors'
    else:
        device_input_choice = 'Paper'

    print("Device has selected",device_input_choice)

    if (user_input == 1 and device_choice == 2) or (user_input == 2
and device_choice == 1):
        print("Rock Wins in this situation")
        result = 'Rock'
    elif (user_input == 1 and device_choice == 3) or (user_input == 3
and device_choice == 1):
        print("Paper Wins in this situation")
        result = 'Paper'
    else:
        print("Scissors wins in this situation")
        result = 'Scissors'

    if user_input_choice == result:
        print("User Wins the game")
    else:
        print("Device Wins the game")

    print("Would you like to play again ?")
    print("Y for Yes else N for No")
    rematch = input()
    if rematch == 'n' or rematch == 'N':
        break

print("Thank for playing.")

```

```
import base64

# Encoding

sample_string = 'Raunak Joshi'
sample_string_bytes = sample_string.encode('ascii')

encoded_bytes = base64.b64encode(sample_string_bytes)
encoded_bytes_string = encoded_bytes.decode('ascii')

# Decoding

encoded_bytes_string_revert = encoded_bytes_string.encode('ascii')
decoded_bytes = base64.b64decode(encoded_bytes_string_revert)

secret_message = decoded_bytes.decode('ascii')

print(secret_message)
```

```

from tkinter import *

root = Tk()

class Calculator:
    def click_button(self, numbers):
        global operator
        global variable
        self.operator = self.operator + str(numbers)
        self.variable.set(self.operator)

    def clear(self):
        self.entry.delete(0, END)
        self.operator = ""

    def evaluate(self):
        self.answer = eval(self.entry.get())
        self.variable.set(self.answer)
        self.operator = str(self.answer)

    def __init__(self, master):
        self.operator = ""
        self.variable = StringVar()
        frame_s = Frame(master, height=400, width=45)
        frame_s.pack(side=TOP, fill=BOTH, expand=True)
        self.entry = Entry(
            frame_s,
            textvariable=self.variable,
            bg='white',
            width=45,
            bd=20,
            insertwidth=4,
            justify='right',
            font=('arial', 10, 'bold')
        )
        self.entry.pack()

        self.t = Text(self.entry, height=40)

        label_key = Label(root, height=15, width=30, bd=10,
bg='gray50')
        label_key.pack(side=LEFT, fill=BOTH, expand=True)

        label_fkey = Label(root, height=15, width=15, bg='gray25')
        label_fkey.pack(fill=BOTH, expand=True)

        label_7 = Label(label_key)
        label_7.grid(row=0, column=0)
        button_7 = Button(
            label_7,
            text='7',
            font=('Helvetica', '16'),
            command= lambda : self.click_button(7),
            bg='black',
            fg='white'
        )

```

```
button_7.pack()

label_8 = Label(label_key)
label_8.grid(row=0, column=1, padx=20)
button_8 = Button(
    label_8,
    text='8',
    font=('Helvetica', '16'),
    command= lambda: self.click_button(8),
    bg='black',
    fg='white'
)
button_8.pack()

label_9 = Label(label_key)
label_9.grid(row=0, column=2, padx=10)
button_9 = Button(
    label_9,
    text='9',
    font=('Helvetica', '16'),
    command= lambda: self.click_button(9),
    bg='black',
    fg='white'
)
button_9.pack()

label_4 = Label(label_key)
label_4.grid(row=1, column=0, padx=10, pady=10)
button_4 = Button(
    label_4,
    text='4',
    font=('Helvetica', '16'),
    command= lambda: self.click_button(4),
    bg='black',
    fg='white'
)
button_4.pack()

label_5 = Label(label_key)
label_5.grid(row=1, column=1, padx=10, pady=10)
button_5 = Button(
    label_5,
    text='5',
    font=('Helvetica', '16'),
    command= lambda: self.click_button(5),
    bg='black',
    fg='white'
)
button_5.pack()

label_6 = Label(label_key)
label_6.grid(row=1, column=2, padx=10, pady=10)
button_6 = Button(
    label_6,
    text='6',
    font=('Helvetica', '16'),
    command= lambda: self.click_button(6),
```

```

        bg='black',
        fg='white'
    )
    button_6.pack()

    label_1 = Label(label_key)
    label_1.grid(row=2, column=0, padx=10)
    button_1 = Button(label_1, text='1', font=('Helvetica',
'16'),command= lambda: self.click_button(1),bg='black',fg='white')
    button_1.pack()

    label_2 = Label(label_key)
    label_2.grid(row=2, column=1, padx=10)
    button_2 = Button(label_2, text='2', font=('Helvetica',
'16'),command= lambda: self.click_button(2),bg='black',fg='white')
    button_2.pack()

    label_3 = Label(label_key)
    label_3.grid(row=2, column=2, padx=10)
    button_3 = Button(label_3, text='3', font=('Helvetica',
'16'),command= lambda: self.click_button(3),bg='black',fg='white')
    button_3.pack()

    label_0 = Label(label_key)
    label_0.grid(row=3, column=0, padx=10, pady=10)
    button_0 = Button(label_0, text='0', font=('Helvetica',
'16'),command= lambda: self.click_button(0),bg='black',fg='white')
    button_0.pack()

    label_deci = Label(label_key)
    label_deci.grid(row=3, column=1, padx=10, pady=10)
    button_deci = Button(label_deci, text='.', font=('Helvetica',
'16'),command= lambda: self.click_button('.'),bg='black',fg='white')
    button_deci.pack()

    label_equal = Label(label_key)
    label_equal.grid(row=3, column=2, padx=10, pady=10)
    button_equal = Button(label_equal, text='=', font=
('Helvetica', '16'),command= self.evaluate,bg='black',fg='white')
    button_equal.pack()

    label_C = Label(label_fkey)
    label_C.grid(row=0, column=0, columnspan=2)
    button_C = Button(label_C, text='C', font=('Helvetica', '16'),
height=1, width=10,command= self.clear,bg='black',fg='white')
    button_C.pack(side=LEFT)

    label_sub = Label(label_fkey)
    label_sub.grid(row=1, column=0, sticky=W, pady=10)
    button_sub = Button(label_sub, text='-', font=('Helvetica',
'16'), height=1, width=3,command= lambda: self.click_button('-
'),bg='black',fg='white')
    button_sub.pack(side=LEFT)

    label_mul = Label(label_fkey)
    label_mul.grid(row=1, column=1, sticky=E)
    button_mul = Button(label_mul, text='x', font=('Helvetica',

```

```

'16'), height=1, width=3,command= lambda:
self.click_button('*'),bg='black',fg='white')
    button_mul.pack()

    label_div = Label(label_fkey)
    label_div.grid(row=2, column=0, sticky=W)
    button_div = Button(label_div, text='/', font=('Helvetica',
'16'), height=1, width=3,command= lambda:
self.click_button('/'),bg='black',fg='white')
    button_div.pack()

    label_add = Label(label_fkey)
    label_add.grid(row=2, column=1, sticky=E)
    button_add = Button(label_add, text='+', font=('Helvetica',
'16'), height=1, width=3,command= lambda:
self.click_button('+'),bg='black',fg='white')
    button_add.pack()


c = Calculator(root)
root.mainloop()

```