Problem

Create a Countdown Timer window with following features: Pause, Resume, Reset, Stop.

Tools/Modules Used

- **Tkinter:** In this project I have used tkinter module in python to create window, buttons with their respective actions, and message box
- **Time:** I have used time module to implement the timer logic using sleep() function

Procedure/Implementation Logic:

- Step-1: As a first step, I worked on a simple timer program in python
    - Get an input from user from where to start the countdown timer
      Timer = input(“Enter time in seconds”)
    - Run a loop from input by user, till it becomes zero and perform the following:
        - Calculate time in seconds, minutes, hours by doing modulus(%) and division (/) on the input
        - Print the time in required format here I have used hh:mm:ss format
        - Sleep for 1 second
        - Decrement the timer counter by 1
        - Run the loop until timer becomes 0

    ```
    for x in range(timer, 0, -1):
        second = x % 60
        minute = int(x / 60) % 60
        hour = int(x / 3600)
        print(f"{hour:02}:{minute:02}:{second:02}")
        time.sleep(1)
    ```
- Step-2: Next I moved on to create an outline of the window that I want for timer, with required dimensions and title
- Step-3: Next I placed 3 text boxes and their labels in the window. These text boxes will display time in hours, minutes and seconds.
- Step-4; Then I created 5 buttons with names: Start, Pause, Resume, Reset, Stop
- Step-5: After that, I assigned a function in command field for each button to be invoked when the button is clicked
- Step-6: For start button, I used the logic explained in Step-1 and the loop would start running once start is clicked
    - Note: The Input can be provided into all or any of the 3 text boxes, which will be read and converted to total seconds after start button is clicked and this value would serve as input for the timer. Logic is below:

      ```
      timer = int(seconds_tb.get(1.0, 'end')) + int(minutes_tb.get(1.0, 'end')) * 60 +
      int(hours_tb.get(1.0, 'end')) * 3600
      ```
- Step-7: For pause button, since the start loop is already running, a way should be figured out to pause the loop until resume is clicked so I created IntVar() from tkinter the value for this pause_var can be set as desired
- Step-8: I checked the value of pause_var in the loop and used frame.wait_variable(pause_var) function, to wait until the value of variable changes

- Step-9: Then I changed the value of pause_var in resume function so that the loop will continue from where it paused. This resume function will be invoked on clicking Resume button
- Step-10: Finally I implemented Reset and Stop using different IntVar() variables, checked the value of variables in loop and break out of the loop
- Step-11: For Reset button action, the loop will break, also the values will be deleted and 00 will be inserted in all 3 boxes, however for Stop button, only the loop will break and values will remain the same as they were when stop button was clicked
- Step-12: Finally, I used messagebox of tkinter to pop a warning window displaying "Time's Up, in case timer completed till 0

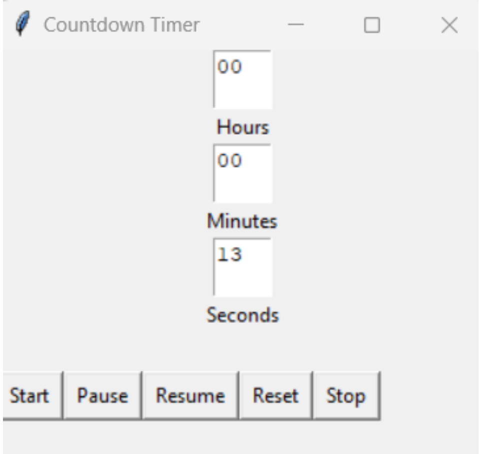
Coding

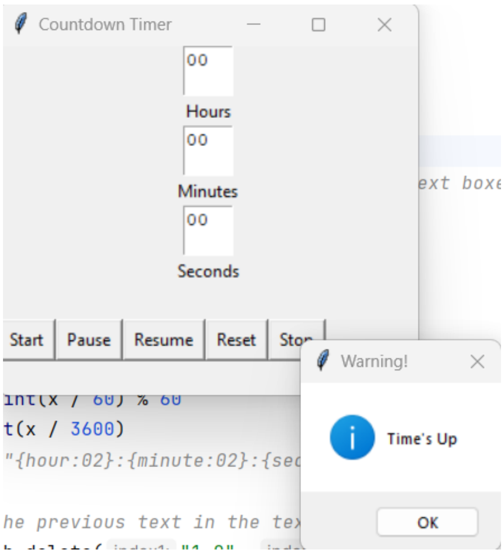
```
import time
from tkinter import *
import tkinter
from tkinter.messagebox import showinfo

def pause():
    pause_var.set(1)

def resume():
    pause_var.set(2)

def reset():
    reset_var.set(3)

    seconds_tb.delete("1.0", "end")
    minutes_tb.delete("1.0", "end")
    hours_tb.delete("1.0", "end")

    # Set the values to 00 after clearing the boxes
    seconds_tb.insert(END, "00")
    minutes_tb.insert(END, "00")
    hours_tb.insert(END, "00")

def stop():
    stop_var.set(4)

def start():
    #Get input from user through data entered in text boxes and convert them to seconds
    timer = int(seconds_tb.get(1.0, 'end')) + int(minutes_tb.get(1.0, 'end')) * 60 + int(hours_tb.get(1.0,
'end')) * 3600  # int(input("Enter the time in seconds: "))

    #After getting input, clear the text boxes
    seconds_tb.delete("1.0", "end")
    minutes_tb.delete("1.0", "end")
    hours_tb.delete("1.0", "end")

    # After clearing input, initialise the values in text boxes (set to 00)
    seconds_tb.insert(END, "00")
    minutes_tb.insert(END, "00")
    hours_tb.insert(END, "00")

    for x in range(timer, 0, -1):
        second = x % 60
        minute = int(x / 60) % 60
        hour = int(x / 3600)
        # print(f"{hour:02}:{minute:02}:{second:02}")

        # Clear the previous text in the text boxes
        seconds_tb.delete("1.0", "end")
        minutes_tb.delete("1.0", "end")
        hours_tb.delete("1.0", "end")
        frame.update()

        #Insert the newly calculated values in the text boxes
        seconds_tb.insert(END, f"{second:02}")
        minutes_tb.insert(END, f"{minute:02}")
        hours_tb.insert(END, f"{hour:02}")
        frame.update()

        #for pause functionality. If pause is clicked, this variable will be set to 1(any value can be used here), hence waiting the program if the value is 1
        if pause_var.get() == 1:
            frame.wait_variable(pause_var)

        #for reset or stop functionality.
        # If reset/stop is clicked, the corresponding variables will be set to 3/4(any value can be used here)
        # and will break out of loop

        if reset_var.get() == 3 or stop_var.get() == 4:
            break

        time.sleep(1)

    if stop_var.get() != 4 :
        # After getting input, clear the text boxes
        seconds_tb.delete("1.0", "end")
        minutes_tb.delete("1.0", "end")
        hours_tb.delete("1.0", "end")

        # After clearing input, initialise the values in text boxes (set to 00)
        seconds_tb.insert(END, "00")
        minutes_tb.insert(END, "00")
        hours_tb.insert(END, "00")

    if stop_var.get() != 4 and reset_var.get() != 3:
        showinfo("Warning!","Time's Up")
```

```
# creating Tk window
frame = Tk() #It helps to display the root window and manages all the other components of the tkinter application

# setting geometry(size) of tk window
frame.geometry("300x250")

#giving the title for frame
frame.title("Countdown Timer")

pause_var = tkinter.IntVar()
reset_var = tkinter.IntVar()
stop_var = tkinter.IntVar()

#Syntax for creating text box, T = Text(root, bg, fg, bd, height, width, font, ..)

#Creating text box for hours
hours_tb = Text(frame, height = 2, width = 4)
hours_tb.pack(side=TOP)

#Creating label for giving the name for hours text box
hours_lb = Label(frame, text = "Hours")
hours_lb.pack(side=TOP)

#Creating text box for Minutes
minutes_tb = Text(frame, height = 2, width = 4)
minutes_tb.pack(side=TOP)

#Creating label for giving the name for Minutes text box
minutes_lb = Label(frame, text = "Minutes")
minutes_lb.pack(side=TOP)

#Creating text box for Seconds
seconds_tb = Text(frame, height = 2, width = 4)
seconds_tb.pack(side=TOP)

#Creating label for giving the name for Seconds text box
seconds_lb = Label(frame, text = "Seconds")
seconds_lb.pack(side=TOP)

#Initalise the text boxes with 0

seconds_tb.insert(END, "00")
minutes_tb.insert(END, "00")
hours_tb.insert(END, "00")

# Create a Start Button
start_btn = Button(frame, text = 'Start',command = start)
start_btn.pack(ipadx=3,ipady=2,side=LEFT)

# Create a Pause Button
```

```
pause_btn = Button(frame, text = 'Pause',command = pause)
pause_btn.pack(ipadx=3,ipady=2,side=LEFT)

# Create a Resume Button
resume_btn = Button(frame, text = 'Resume',command = resume)
resume_btn.pack(ipadx=3,ipady=2,side=LEFT)

# Create a Reset Button
reset_btn = Button(frame, text = 'Reset',command = reset)
reset_btn.pack(ipadx=3,ipady=2,side=LEFT)

# Create a Stop Button
stop_btn = Button(frame, text = 'Stop',command = stop)
stop_btn.pack(ipadx=3,ipady=2,side=LEFT)

#to run the frame using mainloop()
#mainloop() is an infinite loop used to run the application, wait for an event to occur and process the event as long as the window is not closed.
frame.mainloop() #to initiate the program
```

Output Screenshot





Time's Up

```
ext boxe
int(x / 60) % 60
t(x / 3600)
"{hour:02}:{minute:02}:{sec

he previous text in the tex
```