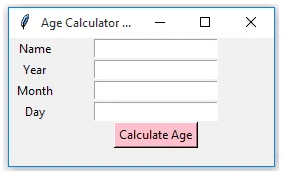
**TASK 11 Use Tkinter module for UI design.**

**Problem 1:**

Create an Age calculator using Tkinter module. In this age calculator app, users can type in their date of birth, and the app will calculate and display their age automatically.



**Program:**

import tkinter as tk

from datetime import datetime

def calculate\_age():

today = datetime.now()

birth\_date = datetime(int(year\_entry.get()), int(month\_entry.get()), int(day\_entry.get()))

age = today.year - birth\_date.year - ((today.month, today.day) < (birth\_date.month, birth\_date.day))

result\_label.config(text=f"Your age is {age}")

root = tk.Tk()

root.title("Age Calculator")

tk.Label(root, text="Name:",width=50).**pack() #** his method adds the label to the parent widget (root)

name\_entry = **tk.Entry(root)** # which is a single-line text input field that allows users to enter information.

name\_entry.**pack()** # to add the name\_entry widget (which is an entry field for user input) to the main application window (root).

tk.Label(root, text="Year of Birth:").pack()

year\_entry = tk.Entry(root)

year\_entry.pack()

tk.Label(root, text="Month of Birth:").pack()

month\_entry = tk.Entry(root)

month\_entry.pack()

tk.Label(root, text="Day of Birth:").pack()

day\_entry = tk.Entry(root)

day\_entry.pack()

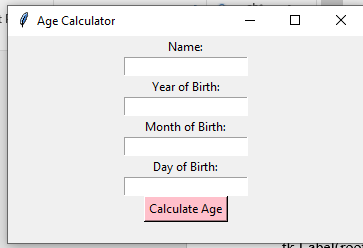
tk.Button(root, text="Calculate Age", background='pink',foreground='black',command=calculate\_age).pack()

result\_label = tk.Label(root, text="")

result\_label.pack()

root.mainloop() # This line starts the Tkinter event loop, keeps the application responsive, The event loop keeps the application running, waiting for user interactions (like button clicks or keyboard input).

**Output:**



**Problem 2:**

Your father wants you to create a digital clock and you decided to show your programming skills. You also offer your father to give specifications to design your clock. Create a Digital clock using Tkinter.

**Input Format :**

Style

size

**Output Format :**

Digital Clock

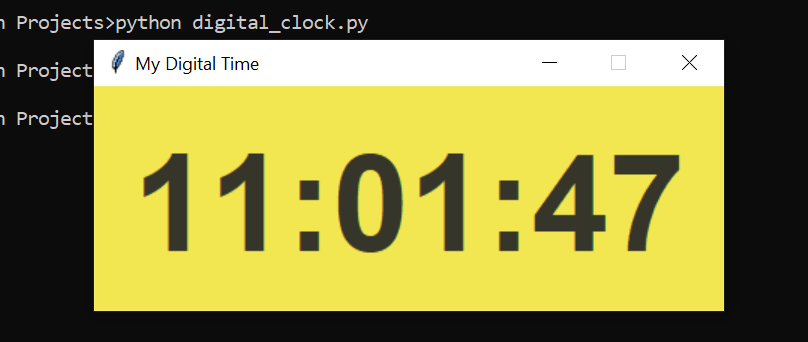
**Test Case 1  
Input (stdin)**

calibri

40

Bold

**Expected Output**



**Program:**

import tkinter as tk

from datetime import datetime

def update\_time():

current\_time = datetime.now().strftime("%H:%M:%S")

time\_label.config(text=current\_time)

time\_label.after(1000, update\_time)

root = tk.Tk()

root.title("Digital Clock")

style = "calibri"

size = 40

font\_style = (style, size, "bold")

time\_label = tk.Label(root, font=font\_style, anchor="center", background='purple',foreground='white') # **anchor="center"**: Centers the text within the label

time\_label.pack(fill=tk.BOTH, expand=True) # The label dynamically adjusts to fill the entire window space, ensuring that the user interface remains neat and functional, regardless of the window size.

update\_time() # responsible for updating the displayed time at regular intervals

root.mainloop()

**Output:**

