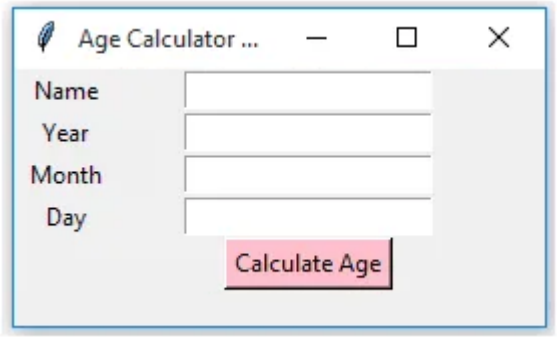
TASK 11 **Use Tkinter module for UI design.**

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()
```

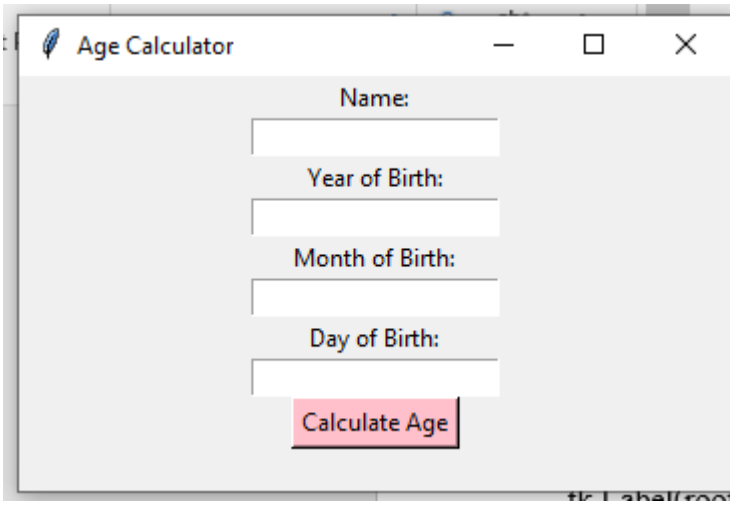
```
day_entry = tk.Entry(root)
day_entry.pack()

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 :

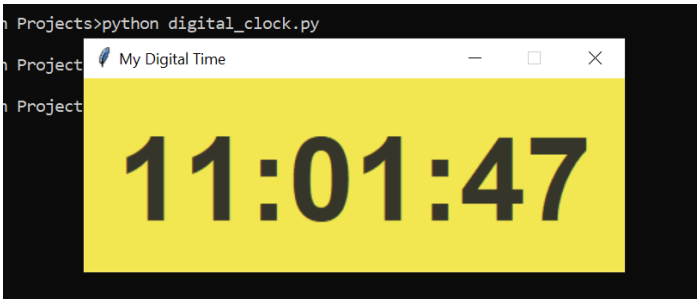
Digital Clock

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: