```
import tkinter as tk
from tkinter import font

class FontChangerApp:
    def __init__(self, root):
        self.root = root
        self.root.title("Font Changer App")

        # Initial font settings
        self.current_font = font.nametofont("TkDefaultFont")
        self.label_text = "Hello, Tkinter!"
        self.create_widgets()

    def create_widgets(self):
        # Create a label with initial font settings
        self.label = tk.Label(self.root, text=self.label_text, font=self.current_font)
        self.label.pack(pady=20)

        # Font Name Entry
        font_name_label = tk.Label(self.root, text="Font Name:")
        font_name_label.pack()
        self.font_name_entry = tk.Entry(self.root)
        self.font_name_entry.pack()

        # Bold Checkbutton
        bold_var = tk.BooleanVar()
        bold_checkbutton = tk.Checkbutton(self.root, text="Bold", variable=bold_var,
command=self.update_font)
        bold_checkbutton.pack()

        # Font Size Entry
```

```
        font_size_label = tk.Label(self.root, text="Font Size:")
        font_size_label.pack()
        self.font_size_entry = tk.Entry(self.root)
        self.font_size_entry.pack()

        # Update Font Button
        update_button = tk.Button(self.root, text="Update Font", command=self.update_font)
        update_button.pack(pady=10)

    def update_font(self):
        # Get values from entry widgets
        font_name = self.font_name_entry.get() if self.font_name_entry.get() else
self.current_font.cget("family")
        font_size = int(self.font_size_entry.get()) if self.font_size_entry.get() else
self.current_font.cget("size")
        font_weight = "bold" if self.bold_checkbutton.get() else "normal"

        # Update label font
        new_font = font.Font(family=font_name, size=font_size, weight=font_weight)
        self.label.config(font=new_font)
        self.current_font = new_font

if __name__ == "__main__":
    root = tk.Tk()
    app = FontChangerApp(root)
    root.mainloop()
```

```
def count_repeated_characters(input_string):
    char_count = {}

    for char in input_string:
        if char.isalpha():  # Consider only alphabetical characters
            char = char.lower()  # Convert to lowercase to treat 'A' and 'a' as the same character
            char_count[char] = char_count.get(char, 0) + 1

    repeated_chars = {char: count for char, count in char_count.items() if count > 1}
    return repeated_chars

# Sample string
sample_string = 'the quick brown fox jumps over the lazy dog'

# Count repeated characters
result = count_repeated_characters(sample_string)

# Display the result
output = ', '.join([f'{char}-{count}' for char, count in result.items()])
print(f"Expected output: {output}")
```