Zeus D. Elderfield
CS3202N - G2
Programming Assignment #3

Python
```python
import tkinter as tk

# ----------------------------
# Assignment Algorithm
# ----------------------------

def checkABCCount(text: str) -> str:
    letter_count = {
        'a': 0,
        'b': 0,
        'c': 0
    }

    prev_letter = chr(ord('a') - 1) # set prev_letter to one character before
'a' to keep order

    for c in text:
        character = c.lower() # to include uppercase

        if character not in letter_count:
            return False # return False if character is not a, b, or c
        if character != prev_letter and ord(character) != ord(prev_letter) + 1:
            return False # return False if there is a character that is skipped
in order

        letter_count[character] += 1
        prev_letter = character

    return letter_count['a'] == letter_count['b'] and letter_count['b'] ==
letter_count['c']

# ----------------------------
# Theme Variables (Minimalist)
# ----------------------------
THEME = {
    "bg": "#fefefe",            # light background
    "fg": "#222222",            # dark text
    "accent": "#4a90e2",        # subtle accent
    "font_main": ("Arial", 12),
    "font_label": ("Arial", 10, "bold"),
    "padding": 12,
    "entry_bg": "#ffffff",     # white input/output boxes
    "entry_fg": "#222222",
    "button_bg": "#4a90e2",
    "button_fg": "#ffffff",
    "button_alt_bg": "#e0e0e0",
    "button_alt_fg": "#222222"
}
# ----------------------------

def process_input():
    text = input_entry.get()
    result = checkABCCount(text)

    output_entry.config(state="normal")
    output_entry.delete(0, tk.END)
    output_entry.insert(0, "YES" if result else "NO")
    output_entry.config(state="readonly")

def clear_fields():
    input_entry.delete(0, tk.END)
    output_entry.config(state="normal")
    output_entry.delete(0, tk.END)
    output_entry.config(state="readonly")

# Main window
root = tk.Tk()
root.title("Programming Assignment #3")
root.configure(bg=THEME["bg"])
root.geometry("400x220")
root.resizable(False, False)

# Input label and entry
input_frame = tk.Frame(root, bg=THEME["bg"])
input_frame.pack(fill="x", padx=THEME["padding"], pady=(THEME["padding"], 0))

tk.Label(
    input_frame,
    text="Enter text:",
    bg=THEME["bg"],
    fg=THEME["fg"],
    font=THEME["font_label"],
    anchor="w"
```

```
).pack(fill="x")

input_entry = tk.Entry(
    input_frame,
    font=THEME["font_main"],
    bg=THEME["entry_bg"],
    fg=THEME["entry_fg"],
    relief="solid",
    bd=1
)
input_entry.pack(fill="x", pady=(2, THEME["padding"]))

# Output label and entry
output_frame = tk.Frame(root, bg=THEME["bg"])
output_frame.pack(fill="x", padx=THEME["padding"])

tk.Label(
    output_frame,
    text="Output:",
    bg=THEME["bg"],
    fg=THEME["fg"],
    font=THEME["font_label"],
    anchor="w"
).pack(fill="x")

output_entry = tk.Entry(
    output_frame,
    font=THEME["font_main"],
    bg=THEME["entry_bg"],
    fg=THEME["entry_fg"],
    relief="solid",
    bd=1,
    state="readonly"
)
output_entry.pack(fill="x", pady=(2, THEME["padding"]))

# Buttons at the bottom
button_frame = tk.Frame(root, bg=THEME["bg"])
button_frame.pack(side="bottom", pady=THEME["padding"])

reverse_btn = tk.Button(
    button_frame,
    text="Check",
    command=process_input,
    bg=THEME["button_bg"],
    fg=THEME["button_fg"],
    font=THEME["font_main"],
    relief="flat",
    padx=20,
    pady=6
)
reverse_btn.pack(side="left", padx=6)

clear_btn = tk.Button(
    button_frame,
    text="Clear",
    command=clear_fields,
    bg=THEME["button_alt_bg"],
    fg=THEME["button_alt_fg"],
    font=THEME["font_main"],
    relief="flat",
    padx=20,
    pady=6
)
clear_btn.pack(side="left", padx=6)

root.mainloop()
```