slip12

```
class s12q1 {
    public static void main(String[] args) {
               String[] s1 = {"Hello", "World", "Java", "Programming"};
        System.out.println("Original and Reversed Strings:");
        for (String str : s1) {
            String reversed = reverseString(str);
            System.out.println("Original: " + str + " | Reversed: " + reversed);
        }
    }

        private static String reverseString(String str) {
        StringBuilder reversed = new StringBuilder(str);
        return reversed.reverse().toString();
    }
}
```

```
import javax.swing.*;
import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;

 class s12q2 extends JFrame implements ActionListener {
    private JTextField numberField;
    private JButton generateButton;
private JList<String> tableList;
private DefaultListModel<String> listModel;

public MultiplicationTableApp() {
    setTitle("Multiplication Table");
    setSize(300, 400);
    setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    setLayout(new FlowLayout());

    JLabel numberLabel = new JLabel("Enter a number:");
    numberField = new JTextField(10);


    generateButton = new JButton("Generate");
    generateButton.addActionListener(this);

    listModel = new DefaultListModel<>();
    tableList = new JList<>(listModel);
    JScrollPane scrollPane = new JScrollPane(tableList);
    scrollPane.setPreferredSize(new Dimension(250, 200));

    add(numberLabel);
    add(numberField);
    add(generateButton);
    add(scrollPane);

    setVisible(true);
}
```

```
    public void actionPerformed(ActionEvent e) {
          String input = numberField.getText();
        try {
            int number = Integer.parseInt(input);
            listModel.clear();

            for (int i = 1; i <= 10; i++) {
                String result = number + " x " + i + " = " + (number * i);
                listModel.addElement(result);
            }
        } catch (NumberFormatException ex) {
            JOptionPane.showMessageDialog(this, "Please enter a valid number.", "Error", JOptionPane.ERROR_MESSAGE);
        }
    }

    public static void main(String[] args) {
        SwingUtilities.invokeLater(MultiplicationTableApp::new);
    }
}
```

```
import tkinter as tk

from tkinter import font
```

```
def update_font():
    selected_font = font_name_var.get()
    selected_size = font_size_var.get()
    is_bold = bold_var.get()

    weight = "bold" if is_bold else "normal"
    new_font = (selected_font, selected_size, weight)
    label.config(font=new_font)

root = tk.Tk()
root.title("Font Style Changer")

label = tk.Label(root, text="Change my font style!", font=("Arial", 16))
label.pack(pady=20)

font_name_var = tk.StringVar(value="Arial")
font_size_var = tk.IntVar(value=16)
bold_var = tk.BooleanVar()

font_name_label = tk.Label(root, text="Font Name:")
font_name_label.pack()
font_name_menu = tk.OptionMenu(root, font_name_var, "Arial", "Courier", "Helvetica", "Times", "Verdana")
font_name_menu.pack()

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

```
font_size_spinbox = tk.Spinbox(root, from_=8, to=72, textvariable=font_size_var)
font_size_spinbox.pack()

bold_checkbox = tk.Checkbutton(root, text="Bold", variable=bold_var)
bold_checkbox.pack()

apply_button = tk.Button(root, text="Apply", command=update_font)
apply_button.pack(pady=10)
root.mainloop()
```

```
from collections import Counter

def count_repeated_chars(input_string):

    char_count = Counter(input_string)

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

    sorted_repeated_chars = sorted(repeated_chars.items(), key=lambda x: input_string.index(x[0]))

    output = ', '.join(f"{char}-{count}" for char, count in sorted_repeated_chars)
    return output
```

```
sample_string = 'thequickbrownfoxjumpsoverthelazydog'

result = count_repeated_chars(sample_string)

print("Repeated characters:", result)
```