slip27

```
import tkinter as tk

class SentenceAltererApp:
    def __init__(self, master):
        self.master = master
        self.master.title("Sentence Alterer")

        self.label = tk.Label(master, text="Enter a sentence:")
        self.label.pack(pady=10)

        self.entry = tk.Entry(master, width=50)
        self.entry.pack(pady=5)

        self.alter_button = tk.Button(master, text="Alter Sentence",
command=self.alter_sentence)
        self.alter_button.pack(pady=20)

        self.result_label = tk.Label(master, text="", wraplength=400)
        self.result_label.pack(pady=10)

    def alter_sentence(self):
        original_sentence = self.entry.get()
        altered_sentence = []
        for char in original_sentence:
```

```
            if char.isdigit():
                altered_sentence.append('?')
            elif char.isspace():
                altered_sentence.append('*')
            elif char.isalpha():
                altered_sentence.append(char.swapcase())
            else:
                altered_sentence.append(char)
        result = ''.join(altered_sentence)

        self.result_label.config(text=result)
root = tk.Tk()
app = SentenceAltererApp(root)
root.mainloop()
```

```
import java.awt.*;
import java.awt.event.*;
import java.io.File;

public class s27q2 extends Frame implements ActionListener {

   private TextField directoryField;
   private List fileList;
   private Button listButton;
```

```
public s27q2() {

    setTitle("Directory Lister");
    setSize(500, 400);
    setLayout(new FlowLayout());
    setResizable(false);


    directoryField = new TextField(30);
    listButton = new Button("List Files");
    fileList = new List(15, false);


    add(new Label("Enter Directory Path:"));
    add(directoryField);
    add(listButton);
    add(fileList);


    listButton.addActionListener(this);


    addWindowListener(new WindowAdapter() {
        public void windowClosing(WindowEvent e) {
            System.exit(0);
        }
```

```
    });

    setVisible(true);
}


public void actionPerformed(ActionEvent e) {

    fileList.removeAll();


    String dirPath = directoryField.getText();


    File directory = new File(dirPath);


    if (directory.exists() && directory.isDirectory()) {

        String[] items = directory.list();

        if (items != null) {


            for (String item : items) {

                fileList.add(item);

            }

        } else {
```

```
            fileList.add("No files or subdirectories found.");
        }
    } else {

        fileList.add("Invalid directory. Please enter a valid directory path.");
    }
  }

  public static void main(String[] args) {

    new s27q2();
  }
}
```

```
tuples_list = [(1, 'apple'), (2, 'banana'), (3, 'cherry')]

numbers, fruits = zip(*tuples_list)

numbers_list = list(numbers)
fruits_list = list(fruits)

print("Numbers:", numbers_list)
print("Fruits:", fruits_list)
```

```
import tkinter as tk
from tkinter import messagebox

class DecimalConverterApp:
    def __init__(self, master):
        self.master = master
        self.master.title("Decimal to Other Bases Converter")

        # Label for instructions
        self.label = tk.Label(master, text="Enter a decimal number:")
        self.label.pack(pady=10)

        # Entry for the decimal number
        self.entry = tk.Entry(master)
        self.entry.pack(pady=5)

        # Button to convert the number
        self.convert_button = tk.Button(master, text="Convert",
command=self.convert_number)
        self.convert_button.pack(pady=20)

        # Labels to display the results
        self.binary_label = tk.Label(master, text="")
        self.binary_label.pack(pady=5)
```

```
        self.octal_label = tk.Label(master, text="")
        self.octal_label.pack(pady=5)

        self.hex_label = tk.Label(master, text="")
        self.hex_label.pack(pady=5)

    def convert_number(self):
        """Convert the decimal number to binary, octal, and hexadecimal."""
        decimal_str = self.entry.get()

        # Validate input
        try:
            decimal_number = int(decimal_str)
            if decimal_number < 0:
                raise ValueError("Please enter a non-negative integer.")
        except ValueError as e:
            messagebox.showerror("Invalid input", str(e))
            return

        # Perform conversions
        binary_number = bin(decimal_number)[2:]  # Remove the '0b' prefix
        octal_number = oct(decimal_number)[2:]   # Remove the '0o' prefix
        hex_number = hex(decimal_number)[2:].upper()  # Remove the '0x' prefix and convert to uppercase

        # Display the results
        self.binary_label.config(text=f"Binary: {binary_number}")
```

```
        self.octal_label.config(text=f"Octal: {octal_number}")
        self.hex_label.config(text=f"Hexadecimal: {hex_number}")

# Create the main window
if __name__ == "__main__":
    root = tk.Tk()
    app = DecimalConverterApp(root)
    root.mainloop()
```