golang

day #2 assignment

Assignment #1

Number to words

Write a function called "numberToWords" that takes a number between 1 and 99,99,99,999 and returns a `string` representing the input number in words.

```
func numberToWords(num int) string {
   return ""
}
```

For example,

```
numberToWords(12345) // should return "twelve thousand three hundred forty five"
numberToWords(10203040) // should return "one crore two lakh three thousand for
numberToWords(101) // should return "one hundred one"
```

In the `main()` function, acccept a number from the user and print the text equivalent of the same.

Assignment #2

Text File Analyzer

Objective: Write a Go program that reads a text file, analyzes its contents, and performs various operations on slices, implements error handling, and defines functions to accomplish the tasks.

Requirements:

1. **Slice Operations:**
    - Read the contents of the text file into a slice of strings. Each line of the file should be an element in the slice.
    - Implement a function to find the longest and shortest lines in the file.
    - Implement a function to count the number of words in each line and store them in a slice of integers.
    - Implement a function to sort the slice of words count in descending order.
2. **Error Handling:**
    - Handle errors that may occur during file reading and other operations.
    - Display appropriate error messages if any operation fails.
3. **Text File Handling:**

    - Open and read the contents of a text file specified by the user.
    - Close the file after reading.
4. **Functions:**
    - Define functions for each of the tasks mentioned above (e.g., `readFile`, `findLongestLine`, `findShortestLine`, `countWords`, `sortWordCount`).

Additional Instructions:

- The program should take the filename as input from the user.
- Display the contents of the longest and shortest lines along with their line numbers.
- Display the word count for each line.
- Display the sorted word count.
- Ensure the program is well-documented with comments explaining the purpose of each function and major blocks of code.
- Test your program with different text files of varying lengths and contents to ensure it works correctly under various scenarios.

Sample Output:

```
Enter the filename: example.txt

Contents of the file:
This is a sample text file.
It contains multiple lines.
Each line will be analyzed.

Longest line:
Line 2: It contains multiple lines.

Shortest line:
Line 1: This is a sample text file.

Word count for each line:
Line 1: 5 words
Line 2: 4 words
Line 3: 3 words

Sorted word count:
5 words
4 words
3 words
```

Function signature reference:

```
// Function to find the longest line in the slice of strings
func findLongestLine(lines []string) (string, int) {
    // Implementation to find the longest line
}
```

```
// Function to find the shortest line in the slice of strings
func findShortestLine(lines []string) (string, int) {
    // Implementation to find the shortest line
}

// Function to count the number of words in each line and store them in a slice
func countWords(lines []string) []int {
    // Implementation to count words in each line
}

// Function to sort the slice of word counts in descending order
func sortWordCount(wordCounts []int) {
    // Implementation to sort word counts
}
```

Assignment #3

CRUD operations on CSV file

Create CLI based menu-driven Go application that provides the following menu option:

```
1. Add new customer
2. View all customers
3. Search customers by city
4. Delete a customer (by id)
5. Search customer by id and edit/update details
6. Exit
```

The data for the application is maintained in a CSV file.

Sample CSV data with headers:

```
id,name,city,email,phone
1,Dell Goldfinch,Shijiazhai,dgoldfinch0@netlog.com,8609657940
2,Robb Groven,Tabu,rgroven1@bandcamp.com,5208396145
3,Frannie Mardoll,Pacajus,fmardoll2@tuttocitta.it,3377870047
4,Donaugh Bierling,Gvardeysk,dbierling3@cmu.edu,1214078739
5,Clarie Swiggs,Bílovice,cswiggs4@rediff.com,6662222043
6,Cahra Michiel,Cennan,cmichiel5@usnews.com,3641707554
7,Hinda Moye,Sanjiang,hmoye6@yellowbook.com,1037138359
8,Malia Larmouth,Ābdānān,mlarmouth7@histats.com,8858893642
9,Florentia Blundon,Zaoxi,fblundon8@quantcast.com,7373066451
10,Erwin Kohneke,Zhongxiao,ekohneke9@alibaba.com,9725490770
```

Make the application modular by splitting the code into reusable functions.
