Go Programming Fundamentals

Assignment #4 Due Date:

Employee Database

Programming Requirements

- Submit your assignment into a **single Go language file**.
- You must write **your name** at the top of your assignment source list.
- Write your **compiler version and operating system name** at the top of your assignment source list.
- Assignment that is turned in late will lose **one point per day** starting after the due date.
- Make sure that you do appropriate **error checking** in your program. (User-friendliness)
- Do not turn in **incomplete or crashing program**, you will receive **zero points**.
- Do not **Import third part packages** into your golang source file, your assignment will not be graded.
- Do not use golang **container** package for **Link List**, your assignment will not be graded.
- Make sure to read **grading policy** carefully that will tell you how your assignment is graded.

Assignment #4 Grading Policy

Category	Points Possible	Points Received
Correctness and Efficiency .	10	
Meaningful variable names	10	
Usage of **comma** in a salary field.	10	
Use of `iota` type	10	
Command line argument for input file	10	
Deleting an Employee	10	
Adding an Employee in a sorted order while reading a file.	10	
Saving Link List data to a file.	10	
Complete Documentation	10	
User-friendliness and Code Readability	10	
Total	100	

Notes:
Use comma in the salary field if salary is greater than 999.99 (1,000.00)
Use data file that is provided for this assignment.
Write Link List data to an input file using **semicolon delimiter**. See Example 1. **Do not write data to a separate file**.

Example 1:
```
Kurt Lamm;27;90000
```

Assignment Description

Write a program using link list to manage an Employee database. Each entry in the data file includes three fields: Employee Name, Employee Age, and Employee Salary. Data is semicolon delimited in the file. Your program should read data from the file into a link list in a sorted order by employee name. Your program must be able to open a data file when its name is passed as a **command line argument**. When user adds an employee record, make sure that data goes into a link list in a sorted order. When "**List All Employees**" is selected by the user it should display the employee data in a sorted order by employee name and formatted according the table in figure 1. When user selects "**Exit Employee Database**", write all the data, if need, from the link list to a file and terminate the program gracefully. Display the following menu one the screen after reading the data file into a link list:

```
Menu Options:

1. Add Employee
2. Delete Employee
3. Search Employee
4. List All Employees
5. Save Employee Database
6. Exit Employee Database
Enter Your Choice:
```

Input File Data for Project #4

Use any text editor to enter following data into a file.

```
Patrick Stroud;48;140565
Ursula Spencer;27;36450
Clifton Stillman;65;99900
William Reynolds;37;77550
Dean Niles;53;120000
John Kaufman;53;69597
Paul Kane;51;169650
Paul Goldsmith;60;200000
Larry Godwin;45;59500
Kurt Lamm;39;90000
Susan Carltom;42;85000
Cameron Wu;29;50589
```

Output from the program

Figure 1

```
# Employee Name       Age          Salary
=============================================
1. Cameron Wu          29           50,589
2. Clifton Stillman    65           99,900
3. Dean Niles          53          120,000
4. John Kaufman        53           69,597
5. Kurt Lamm           39           90,000
6. Larry Godwin        45           59,500
7. Patrick Stroud      48          140,565
8. Paul Goldsmith      60          200,000
9. Paul Kane           51          169,650
10. Susan Carlton      42           85,000
11. Ursula Spencer     27           36,450
12. William Reynolds   37           77,550
```

Error-Handling

1. The program ignores invalid user commands.