Programmieren in C | Programmentwurf

Mit dieser Hausaufgabe können 10 Punkte erreicht werden. Stellen Sie ihren Source Code zur dieswöchigen Deadline als Github Repository zur Verfügung.

Zur Abgabe Legen Sie die Datein linkedListLib.c, linkedListLib.h und main.c, welche ihren Lösungscode beinhalten in einem Ordner HA03 in ihrem persönliches Repository ab. Schreiben Sie ihr Programm so, dass der Compiler weder Errors noch Warnings wirft. `(gcc linkedListLib.c main.c -Wall -o main.exe)`

Aufgabe: Simple Linked List **10 Punkte**

Lesen Sie den Quellcode des *linked List*-Programms bestehend aus:

- main.c
- linkedListLib.c
- linkedListLib.h

Das Programm soll in der finalen Version folgende Funktionalitäten bieten:

1. Liste ausgeben
2. Neues Element an die Liste anhängen
3. Spezifisches Listenelement löschen **(2 Punkte)**
4. Gesamte Liste löschen **(1 Punkt)**
5. Liste speichern **(3 Punkte)**
6. Liste laden **(3 Punkte)**
7. Liste sortieren
8. Programm beenden **(1 Punkt)**

1. und 2. und wurde bereits in der Vorlesung implementiert. **In dieser Hausaufgabe sollen die Funktionen 3. - 6. und 8. vervollständigt werden.** Dazu finden Sie in der *HA03_linkedListLib.c** Datei als Kommentar folgende Anweisung:

```
/* YOUR CODE HERE */
/* ---------------- */
```

Implementieren Sie hier ihren Code. Halten Sie bei ihrer Umsetzung die in der Vorlesung besprochenen clean code Richtlinien ein. Des Weiteren führen Bugs, welche beim Testen der Abgabe auftreten zu Punktabzügen. Denken Sie deshalb daran, wo immer nötig ein sinnvolles Errorhandling zu implementieren, so dass ihr Programm auch dann nicht abstürzt, wenn der user eine nicht vorgesehene Eingabe tätigt.

Spezifisches Element löschen Mit dieser Funktion soll ein spezielles Listenelement gelöscht werden. Das Programm gibt dazu zunächst die gesamte Liste in die Konsole aus. Anschließend wird der user nach dem Index des zu löschenden Elements gefragt. Achten Sie bei der Umsetzung auf ein entsprechendes Errorhandling.

Gesamte Liste löschen Mit dieser Funktion wird die gesamte geladene Liste gelöscht. Nicht jedoch etwaige Savedateien.

Liste speichern Mit dieser Funktion wir die Liste in einer Textdatei abgespeichert. Verwenden Sie hierfür die `fprintf()` Funktion so, dass die gesamten Informationen über die in der Liste abgespeicherten Dateien in die Savedatei geschrieben werden. Die Datei soll per default im gleichen Verzeichnis abgelegt werden, in dem sich auch das Programm befindet. Der user soll die Möglichkeit haben den Namen der Savedatei (name.txt) festzulegen.

Liste laden Mit dieser Funktion wird dem user zunächst alle im Verzeichnis verfügbaren Savedatein (Textdatein) aufgelistet, welche potentiell geladen werden können. Verwenden Sie hierzu den Befehl `system("dir *.txt")` bzw. `system("ls *.txt")` Nach Eingabe des users wird die entsprechende Savedatei geladen. Verwenden Sie dazu die `fscanf()`-Funktion.

Programm beenden Mit dieser Funktion wird der user abgefragt, ob er vor dem Beenden des Programms die aktuell im Speicher befindliche Liste speichern möchte. In kommerziellen Programmen wird der user beim Beenden des Programmes lediglich zum Speichern aufgefordert, wenn noch nicht abgespeicherte Änderungen vorliegen. Diese Differenzierung muss hier nicht umgesetzt werden. Der user wird also unabhängig vom aktuellen Arbeitsstand gefragt, ob er die Liste speichern möchte.

*Die Funktion 8. wird in den nächsten Veranstaltung umgesetzt