Institut für Informatik
Prof. Dr. Stefan Conrad

Sommersemester 2021
Übung 5

Hardwarenahe Programmierung
Gruppe 17 (Florian)

In dieser Übung arbeiten Sie mit komplexen Datentypen in C-Programmen, die über structs realisiert werden.

- *Denken Sie wie immer daran, Ihre Lösungen im ILIAS hochzuladen und den Test im ILIAS zu absolvieren!*
- *Sie brauchen keine ungültigen Eingaben oder semantisch falschen Daten abzufangen, wie zum Beispiel das Datum `2020-13-40`.*
- *Die Inhalte aller Übungen werden relevant für das Abschlussprojekt sein. Versuchen Sie trotzdem, alle Aufgaben zu lösen, auch wenn Sie die minimal nötige Anzahl an bestandenen Aufgaben bereits erreicht haben!*

Aufgabe 1 *Haustier (**Pflichtaufgabe**)*

Ergänzen Sie die Dateien `haustier.h` und `haustier.c`, sodass die gegebenen Unittests erfüllt werden.

Ein Haustier wird repräsentiert durch ein `struct` und hat einen Namen, der höchstens 6 Zeichen zuzüglich Nullterminator lang ist und ausschließlich aus Buchstaben a – z (bzw. A – Z für den ersten Buchstaben) besteht. Implementieren Sie weiterhin

- eine Funktion `name_setzen`, die den Namen des Haustiers setzt, dabei alle ungültigen Zeichen verwirft und die Groß- und Kleinschreibung berichtigt, sowie
- eine Funktion `name_ausgeben`, die den Namen in ein übergebenes Array schreibt.

Verwenden Sie den Befehl `make run`, um Ihre Implementierung zu testen.

Aufgabe 2 *Datenstrukturen (**Pflichtaufgabe**)*

(a) Erstellen Sie eine Headerdatei `datum.h`, in welcher Sie ein `struct datum` definieren, das ein Datum mit Tag, Monat und Jahr speichern kann.

(b) Fügen Sie Ihrer Headerdatei einen Prototypen für eine Funktion hinzu, die ein Datum über einen Zeiger mit Jahr, Monat und Tag initialisiert:

```
void datum_setzen(struct datum *d, int jahr, int monat, int tag);
```

Implementieren Sie diese Funktion in der Datei `datum.c`.

(c) Ergänzen Sie Ihre Dateien um eine weitere Funktion, die ein Datum in einen Datenstrom schreibt.

```
void datum_ausgeben(FILE *ausgabe, struct datum *d);
```

Verwenden Sie hierbei das internationale Datumsformat `JJJJ-MM-TT` (Beispiel: `2020-01-30`). In diesem Format werden zuerst das Jahr, dann der Monat und schließlich der Tag ausgegeben. Achten Sie auf die Ausgabe führender Nullen. Hierbei hilft es, die Dokumentation bekannter Ausgabefunktionen zu lesen.

Die Datei wird von der aufrufenden Funktion geöffnet und geschlossen. Dateien dürfen nicht doppelt geschlossen werden, denn dies würde zu einem Fehler führen. Sie dürfen die Datei also nicht selber öffnen oder schließen.

Schreiben Sie nach dem Datum keinen Zeilenumbruch in die Datei.

(d) Ergänzen Sie Ihre Dateien um eine weitere Funktion, die zwei Datum-Objekte vergleicht:

```
int datum_vergleichen(struct datum *a, struct datum *b);
```

Diese Funktion soll folgende Werte zurückgeben:

- 0, wenn beide eingegebenen Daten gleich sind;
- einen beliebigen negativen Wert, wenn `a` zeitlich vor `b` liegt;
- einen beliebigen positiven Wert, wenn `a` zeitlich nach `b` liegt.

(e) Testen Sie Ihre Funktionen mit dem Befehl `make run`.

Aufgabe 3 *Bibliothek (**Pflichtaufgabe**)*

Eine Bibliothek möchte herausfinden, bei welchen Büchern das Ausleihdatum abgelaufen ist. Die Bücher sind in einer Datei gespeichert. Jede Zeile enthält:

- ein Rückgabedatum (`Tag.Monat.Jahr`),
- ein Genre (entweder `Fantasy`, `Krimi`, `Sachbuch` oder `Sci-Fi`) und
- einen Titel bis zum Ende der Zeile (weniger als 50 Zeichen).

Ihre Aufgabe ist es, die Daten aus der Datei zu lesen, nach einem Datum zu unterteilen und die Ausgabe in zwei weitere Dateien zu schreiben.

Die Ausgabe soll die Form `JJJJ-MM-TT, Titel, Genre` haben und könnte zum Beispiel so aussehen:

```
2022-01-09, C von Kopf bis Fuss, Sachbuch
2021-09-16, C von A bis Z, Sachbuch
```

(a) Implementieren Sie in `bibliothek.h` ein `enum genre`, dass das Genre speichert.

(b) Implementieren Sie in `bibliothek.h` ein `struct buch`, dass das Rückgabedatum, das Genre und den Titel eines Buches speichert. Für das Rückgabedatum **muss** das `struct` aus `datum.h` der vorherigen Aufgabe wiederverwendet werden. Zur Speicherung des Genres **muss** das vorher erstellte `enum` verwendet werden. Wenn Sie andere als diese drei Komponenten in ihrem `struct` deklarieren, dann wird diese Aufgabe als nicht bestanden bewertet.

(c) Implementieren Sie in `bibliothek.c` die Funktion `buecher_einlesen`, die die Daten aus drei gegebenen Dateien einliest, in ein Array von Büchern schreibt und die Anzahl zurückgibt.

(d) Implementieren Sie in `bibliothek.c` die Funktion `buecher_ausgeben`, die Bücher aus einem Array in die Datei `faellig` schreibt, wenn ein gegebenes Datum `testdatum` das Rückgabedatum überschritten hat, oder in die Datei `noch_nicht_faellig`, wenn das Datum noch nicht überschritten wurde.

(e) Die `main`-Funktion darf nicht verändert werden!

(f) Testen Sie Ihr Programm mit dem Befehl `make run`.

Hinweis: In C gibt es keine Funktion, die Structs automatisch einliest oder zwischen Strings und Structs konvertiert. Eine Lösung dafür müssen Sie selber implementieren.