

Nr. 1

- a) Warum gilt `strncpy` als *sicher* und sollte bevorzugt verwendet werden? Erklären Sie anhand dieses Beispiels typische Problemstellungen beim Umgang mit Zeichenketten und die Bedeutung des Begriffs *sicher* in diesem Zusammenhang.

Warum gilt `strncpy` als sicher und sollte bevorzugt werden?

Erklären Sie anhand dieses Beispiels **typische Problemstellungen** beim Umgang mit Zeichenketten und die **Bedeutung des Begriffs sicher** in diesem Zusammenhang.

`strncpy` gilt als sicher, da es einem die Möglichkeit gibt, festzulegen, wie viele Zeichen kopiert werden sollen an ein Ziel.

Es sollte bevorzugt werden, da dadurch keine anderen Speicherbereiche ungewollt überschrieben werden.

Dies würde zu einem undefinierten Laufzeitverhalten führen. Das eigene Programm oder andere Programme können z.B. dadurch abstürzen oder sogar das Antiviren-Programm könnte einen unbefugten Speicherzugriff als Virus ansehen (mir schon passiert).

`strncpy` bietet für diese Probleme also eine Lösung an.

Mit "sicher" ist also gemeint, dass insbesondere keine Speicherbereiche überschrieben werden, die einem selber als Programm nicht "gehören" oder die man nicht explizit reserviert hat für sich.

- b) Erstellen Sie ein Programm, das die "Unsicherheit" von `strcpy` demonstriert:



nr1b

Nr. 2

- a) Modellieren Sie einen strukturierten Datentyp für eine Person, der mindestens die Felder Vor- und Nachname besitzt.



nr2a



nr2a

- b) Implementieren Sie eine einfach oder doppelt verkettete Liste, die als Datentyp eine Person aufnehmen kann.

Stellen Sie die Datenstruktur in einer separaten c-Datei zur Verfügung, die über einen geeigneten Header eingebunden werden kann.

Hinweis:

Orientieren Sie sich an den Beispielen aus der Vorlesung (G3-01 Liste & G3-02 Doppelliste).



nr2b

- c) Erweitern Sie Ihre Studienplan-App um die Anwendungsfälle:

7. Hinzufügen beliebig vieler Personen zu einem Termin.

Verwenden Sie Ihre Implementation der Liste aus 2b.

8. Ausgabe aller Termine einer vom Anwender ausgewählten Person.

9. (Informatiker) Entfernen einer Person aus einem Termin.

Anforderungen:

- Die Namen der Personen sollen unique (=einzigartig) innerhalb eines Termins sein, d.h. es darf keine zwei Personen mit demselben Vor- und Nachnamen in einem Termin geben.
- Bei der Ausgabe aller Termine (Anwendungsfall 3, Praktikum 4) sollen die Namen der Personen angezeigt werden.
- (Informatiker)
Die Personen eines Termins sollen bei der Ausgabe (vorherige Anforderung) alphabetisch nach dem Nachnamen sortiert sein.

Hier ist der Source Code:



Nr_2c

- d) Source Code soll robust sein gegen Abstürze:

s. Code von Nr. 2 c), da bereits so implementiert.