

**Hardwarenahe Programmierung**  
Gruppe 11 (Thomas)

*In dieser Übung liegt der Fokus auf dem Umgang mit Strings in C-Programmen.*

**Wichtig:**

- Denken Sie daran, Strings zu terminieren und Variablen zu initialisieren!
- Denken Sie daran, ihre Abgabe hochzuladen und die Tests im ILIAS zu absolvieren!
- Achten Sie darauf, dass Sie Ihre Lösung im ILIAS auch bei der richtigen Übung abgeben, also zum Beispiel nicht die Lösungen dieses Übungsblattes bei Übung 1 hochladen.
- Achten Sie darauf, dass Sie nicht die Abgabe für ein vorheriges Übungsblatt hochladen!
- Laden Sie ihre Abgabe aus dem ILIAS runter, um sich zu vergewissern, dass Sie beim zippen keine Dateien vergessen haben.

**Aufgabe 1** *Strings verbinden (Pflichtaufgabe)*

- (a) Schreiben Sie eine Funktion `strings_verbinden`, welche zwei Strings entgegen nimmt, diese jeweils umdreht und in einem Ausgabestring speichert. Zum Beispiel würden die Strings „nE“ und „et“ zu „Ente“ verbunden werden.

Implementieren Sie die Funktion in einer Datei mit Namen `strings_verbinden.c`. Wählen Sie eine der beiden folgenden Funktionssignaturen:

1. `void strings_verbinden(char zusammen[], char string0[], char string1[])`
2. `void strings_verbinden(char *zusammen, char *string0, char *string1)`

Sie dürfen hierbei annehmen, dass das `char`-Array mit Namen `zusammen`, in dem Sie den String speichern, genau groß genug ist.

- (b) Verwenden Sie den Kommandozeilenbefehl `make run`, der die Tests baut, kompiliert und ausführt. Wenn die Tests durchlaufen heißt dies allerdings noch nicht, dass Ihr Code auch fehlerfrei ist. Achten Sie deshalb wie immer darauf, dass Sie nicht aus uninitialisierten Speicherbereichen lesen und auch sonst keine Speicherzugriffsfehler begehen.

## Aufgabe 2 Groß- und Kleinbuchstaben (*Pflichtaufgabe*)

Schreiben Sie ein Programm, das einen String von `snake_case` zu `camelCase` übersetzt. Hierbei sollen beliebig viele Unterstriche entfernt und eventuell der nächste darauf folgende Kleinbuchstabe durch einen Großbuchstaben ersetzt werden. Zeichen außer Unterstrichen und a-z sollen nicht modifiziert werden.

- Implementieren Sie eine Funktion `void zu_camel_case(char *string)`, die einen String als Parameter übergeben kriegt und diesen entsprechend modifiziert. Zum Beispiel würde der String „eine\_variable“ zu „eineVariable“ übersetzt werden.
- Wir haben Ihnen Unit-Tests und eine Makefile-Datei zur Verfügung gestellt. Sie können die Tests mit dem Befehl `make run` bauen und ausführen. Testen Sie Ihre beiden Funktion mit den bereitgestellten Unit-Tests und stellen Sie sicher, dass alle erfüllt sind. Schauen Sie sich außerdem den Aufbau des Makefiles an, damit Sie in Zukunft selbst Makefiles erstellen können, um Ihren Kompilier- und Testprozess zu vereinfachen.

Mit `make camel_case` wird das Programm `camel_case` erzeugt, womit Sie ihre Funktion auf der Kommandozeile ausprobieren können, wenn Sie möchten.

## Aufgabe 3 Strings ersetzen (*Pflichtaufgabe*)

Gegeben ist ein Array von bis zu 100 Strings und ein Text. Tauschen Sie alle Vorkommen von aufeinanderfolgenden Multiplikationssymbolen (“\*\*\*\*”) im Text nacheinander durch den jeweils nächsten String aus dem String-Array aus. Wenn zum Beispiel das String-Array {“Ohrwurm”, “belegtes Brot”, “Schinken”} zusammen mit dem Text

Was hilft gegen einen \*? - Ein \*\*\* mit \*\*\*\*\*!

gegeben sind, dann lautet der resultierende Text

Was hilft gegen einen Ohrwurm? - Ein belegtes Brot mit Schinken!

und im String-Array stehen danach die Strings {“\*”, “\*\*\*”, “\*\*\*\*\*”}

- Implementieren Sie die Funktion `strings_ersetzen`, die die Platzhalter im String `text` mit Strings aus dem Array `strings` ersetzt und den resultierenden String in das Array `ausgabe` schreibt. Der Eingabetext soll dabei nicht verändert werden. Weiterhin soll die Funktion die Anzahl der ersetzten Strings zurückgeben.

```
int strings_ersetzen(char ausgabe[], char *text, char strings[100][100]);
```

- Es darf angenommen werden, dass die Arrays `ausgabe` und `strings` ausreichend groß sind und das höchstens 100 Strings ersetzt werden.
- Führen Sie die Unit-Tests mit dem Befehl `make run` aus.