

# Übungsblatt Woche 12

## Präsenzaufgaben (18. - 20. Januar)

### Aufgabe 01 (\*\*)

Lassen Sie den Nutzer einen Integer eingeben. Allokieren Sie dynamisch ein Array mit der Größe dieses Integers. Befüllen Sie das Array mit Zufallswerten (`rand()`). Finden Sie anschließend das kleinste und größte Element in diesem Array und berechnen Sie den Durchschnitt aller Zahlen im Array.

### Aufgabe 02 (\*\*\*)

In dieser Aufgabe soll eine Funktion geschrieben werden, welche zwei übergebene Zeichenketten konkateniert. Gehen Sie dabei in folgenden Schritten vor:

- Schreiben Sie zunächst eine Funktion `int string_length(const char* string)`, welche die Länge des übergebenen Strings zurückgibt. Benutzen Sie nicht `strlen` aus der Standardbibliothek. Denken daran, dass Strings in C mit dem Character `'\0'` terminieren (Sie können die Funktion aus der letzten Übung verwenden).
- Schreiben Sie eine Funktion `char* concat(const char *string1, const char *string2)`, der die übergebenen Zeichenketten mit einem Leerzeichen getrennt (Character: `' '`) konkateniert. Benutzen Sie dabei `malloc` (überlegen Sie genau, wieviel Speicherplatz Sie brauchen), gehen Sie zeichenweise vor und benutzen Sie keine String-Funktionen von C. Beispiel: Bei `"Wort1"` und `"Wort2"` soll als Ergebnis `"Wort1 Wort2"` geliefert werden.
- Erklären Sie, warum der Speicherplatz, den Sie in der Funktion `concat` anlegen, nicht am Ende der Funktion freigegeben wird.

### Aufgabe 03 (\*\*\*)

Schreiben Sie eine Funktion, die eine Matrix mit einer festen Kantenlänge anlegt. Nun transponieren Sie die Matrix und speichern das Ergebnis in ein neues Array, das dynamisch angelegt wurde.

## Zusatzaufgaben

### Aufgabe 01 (\*\* - \*\*\*)

Schreiben Sie ein Programm, dass anhand von Kommandozeilenparametern (Integers) Arrays dynamisch so anlegt und dabei jeweils so viel Byte Speicher genutzt werden, wie die Eingaben es vorschreiben.

Beispiel: `linus@tux:~ $ ./a.out 5 10 250`

## Hausaufgaben (18.-20. Januar)

### Aufgabe 01 (\*\*)

Erstellen Sie eine Struktur „Name“, die in den Vor- und den Nachnamen speichert (Jeweils maximal 20 Zeichen). Geben Sie in einer Funktion den vollständigen Namen mit einer Begrüßung aus. Nutzen Sie dazu:

- a) Call-by-Value
- b) Call-by-Reference

Welche Übergabevariante ist sinnvoller?