Funktionsargumente

- Aus der Vorlesung kennen wir zwei Möglichkeiten, Daten an eine Funktion zu übergeben
  - Übergabe als Wertparameter
  - Übergabe als Referenzparameter
- Felder werden *immer* als Referenzparameter übergeben → `pass-array-1.cpp`
- Das gilt aber nicht für Felder in neuen benutzerdefinierten Datentypen (z.B. `struct` → `pass-array-2.cpp`)

& und Überladen von Operatoren

- Das Verhalten des &-Operators unterscheidet sich in Deklarationen und Ausdrücken → `address-op.cpp`
- Überladen von Operatoren ist insbesondere für mathematische Anwendungen sehr nützlich!
- Wir können nun unseren Beispielcode zu komplexen Zahlen aus der letzten Präsenzübung entsprechend anpassen → `complex-struct-2.cc`

Nachbesprechung Übungsaufgaben

- Aufgabe 17: → `a17-kub-spline.cc`
- Aufgabe 18: Umwandlung von Kleinbuchstaben in Großbuchstaben: ASCII Tabelle (5. ÜB)
- Die Buchstaben A bis Z haben Zahlenwerte 66 bis 90
- Die Buchstaben a bis z haben Zahlenwerte 96 bis 112
- Zieht man von einem Kleinbuchstaben den Zahlenwert von a ab und addiert den Zahlenwert von A dazu, so erhält man den entsprechenden Großbuchstaben → `a18-roemisch.cc`

Vorbesprechung Übungsaufgaben

- Aufgabe 19: Rechteck um $\pi/2$ rotieren: Höhe und Breite vertauschen
- Aufgabe 20: Türme von Hanoi, rekursive Lösung
- Lösungsskizze für 3 Scheiben

A	B	C
$[123]$	$[]$	$[]$
$[23]$	$[1]$	$[]$
$[3]$	$[1]$	$[2]$
$[3]$	$[]$	$[12]$
$[]$	$[3]$	$[12]$
$[1]$	$[3]$	$[2]$
$[1]$	$[23]$	$[]$
$[]$	$[123]$	$[]$

- Lösungsidee: Angenommen, man hat eine Funktion `schiebe`, die das Problem für $n-1$ Scheiben löst. Wie kann man sie im Falle von n Scheiben einsetzen?
- Ausgangslage

$$A : [1, \ldots, n] \quad B : [] \quad C : []$$

- Aufruf der Funktion `schiebe(n-1, von A, nach C, über B)`. Ergebnis:

$$A : [n] \quad B : [] \quad C : [1, \ldots, n-1]$$

- Nun verschieben wir die n-te Scheibe von A nach B

$$A : [] \quad B : [n] \quad C : [1, \ldots, n-1]$$

- Aufruf der Funktion `schiebe(n-1, von C, nach B, über A)`. Ergebnis:

$$A : [] \quad B : [1, \ldots, n] \quad C : []$$

- Im Wesentlichen zwei Teilaufgaben
  - Zuerst die Quellstange A leer bekommen (2^{n-1} Schritte)
  - Danach den Zielturm bei B richtig aufbauen ($2^{n-1} - 1$ Schritte)
- Man kann eine rekursive Funktion verwenden, z.B. `schiebe(n, von A, nach B, über C)`, die sich entsprechend für $n - 1$ zweimal aufruft
- Der 1. Aufruf von `schiebe(n-1,...)` + eine zusätzliche Verschiebung der n. Scheibe erledigt die 1. Aufgabe
- Der 2. Aufruf von `schiebe(n-1,...)` erledigt die 2. Aufgabe
- Auf die Abbruchbedingung achten: Wenn die Quellstange leer ist, kann man keine Scheiben mehr verschieben!
- Besonders schön: Die Stangen visualisieren

- A21: Quersumme:

$$10 \to 1 + 0 = 1,$$
$$123 \to 1 + 2 + 3 = 6$$

- Vollständig reduzierte Quersumme

$$124 \to 1 + 2 + 4 = 7 \to 7,$$
$$392 \to 3 + 9 + 2 = 14 \to 1 + 4 = 5$$

- Berechnung der Quersumme: PÜ 3 (wie extrahiert man die Koeffizienten einer Zahl im Dezimalsystem? $\to$ Modulo und restfreie Division)
- Der Rest ist eine einfache Rekursion